

GymApp



GymApp

CURSO PROFISSIONAL DE
GESTÃO E PROGRAMAÇÃO
DE SISTEMAS INFORMÁTICOS

José Carlos da Cunha Rebelo

Trabalho realizado sob a orientação de:

Prof. António Araújo

Escola Profissional de Braga

Braga, 13 de maio de 2024

Nota Introdutória

A Prova de Aptidão Profissional consiste na apresentação e defesa, perante um júri, de um projeto, consubstanciado num produto, material ou intelectual, numa intervenção ou numa atuação, bem como do respetivo relatório final de realização e apreciação crítica, demonstrativo de saberes e competências profissionais, adquiridos ao longo da formação e estruturante do futuro profissional.

A sua realização constitui-se como um dos imperativos legais para a conclusão do curso profissional de (designação do curso), que confere uma dupla certificação: qualificação profissional de nível IV e o 12º ano como certificação escolar de nível secundário.

A presente prova foi realizada na Escola Profissional de Braga no presente ano letivo e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.

Dedicatória

Dedico a minha prova de aptidão profissional a todos os meus professores, aos meus amigos e companheiros que me apoiaram no minha jornada como aluno.

José Carlos da Cunha Rebelo

3

Agradecimentos

Agradeço a todos os meus professores que acompanharam o meu percurso escolar. Agradeço aos meus colegas de turma, que sempre me apoiavam. Agradeço ao professor António Araújo, pelas dicas de melhorias que podia fazer no projeto. Agradeço à professora Ana Marta Vilaverde, a minha diretora de turma, pelo seu apoio e disposição para ajudar todos os alunos.

Índice

1.	Introdução.....	7
2.	Capítulo	10
3.	Novo capítulo	9
3.1	Novo subtítulo.....	9
4.	Novo capítulo	10
4.1	Novo capítulo	11
5.	Conclusão.....	156
6.	Bibliografia	157
7.	Anexos.....	158

Índice de Figuras

Figura 1 - Logótipo do Visual Studio Code.....	10
Figura 2 - Logótipo do Figma.....	11
Figura 3 – Logótipo do Photopea	11
Figura 4 - Logótipo do Android Studio.....	11
Figura 5 - Logótipo da Firebase.....	11
Figura 6 - Logótipo do Expo Go.....	13
Figura 7 - Logótipo do Netlify.....	13
Figura 8 - Logótipo do JavaScript.....	14
Figura 9 - Logótipo do React Native.....	14
Figura 10 - Logótipo do HTML.....	15
Figura 11 - Logótipo do CSS.....	15
Figura 12 - Logótipo da GymApp.....	13

Introdução

Fundamentação do projeto

Como prova de aptidão profissional foi desenvolvida uma app para dispositivos móveis *Android* em *React Native (JavaScript)*, para ginásios, com o objetivo de ajudar os utilizadores e o próprio ginásio. Além da aplicação, fiz uma página web de promoção à aplicação. A aplicação contém um sistema de horário de aulas que permite aos utilizadores visualizarem as aulas disponíveis e marcarem as suas aulas, os utilizadores podem criar os seus próprios planos de treino escolhendo os exercícios. A aplicação permite criar próprios planos de alimentação adequados à dieta pessoal e contém um contador de passos para mostrar o progresso na saúde diariamente, um botão de notificações, uma página para *scan* de código qr (seja para entrada no ginásio como para efetuar outras ações) e uma página de definições para fornecer informações detalhadas sobre o funcionamento da aplicação.

Como surgiu a ideia: A ideia para a realização da minha PAP surgiu quando eu observava o *design* complexo e pouco prático de outras apps de ginásio dos meus colegas. Além disso, achei que seria uma experiência enriquecedora para mim trabalhar com uma linguagem de programação que nunca tinha utilizado.

Objetivos

1. Reforçar a capacidade de superar situações inesperadas ou imprevistas nas diferentes fases do projeto.
2. Desenvolver o espírito crítico e sentido de autonomia.
3. Desenvolver competências de decisão, escolha e procura de informação.
4. Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas.
5. Aumentar a minha capacidade de domínio sobre este tema.
6. Melhorar a disciplina mental, capacidade de organização, metodologia de trabalho.
7. Desenvolver competências nos desenvolvimentos de aplicações moveis.
8. Aplicar conhecimentos e competências adquiridos na área de programação em dispositivos móveis.
9. Aumentar a capacidade e domínio das linguagens *React Native* e *JavaScript*.
10. Desenvolver competências no desenvolvimento de aplicações para o sistema operativo móvel *Android* e adquirir a sensibilidade para a publicação e gestão das mesmas na loja *online Google Play*.

José Carlos da Cunha Rebelo

7

Cronograma do projeto

Cronograma		
Fase	Calendarização	Elenco de atividades nas diversas fases
Conceção	Segunda 25 setembro	Brainstorming
	Terça 26 setembro	Decisão fixa das ideias
	Quarta 27 setembro	Decisão do nome
	Quinta 28 setembro	Separação dos Objetivos
	Sexta 29 setembro	Organização
	Sábado 30 setembro	Preenchimento
Desenvolvimento	Outubro	Design da aplicação
	Até 10 novembro	Desenvolvimento das páginas iniciais

	Até 10 dezembro	Funcionamento de todas as páginas em conjunto
	Até 10 janeiro	Desenvolvimento de funcionalidades básicas
	Até 30 janeiro	Base de dados e ligações à aplicação
	Fevereiro	Desenvolvimento de funcionalidades mais complexas
	Março	Testes e verificação de funcionalidade
Autoavaliação e Relatório	Maio 2024	Entrega do relatório

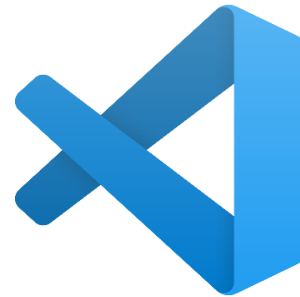
1. Capítulo – Componente teórica

1.1 Serviços e aplicações para o desenvolvimento

Para o desenvolvimento do meu projeto, tive de recorrer ao uso de diversas ferramentas, variando desde linguagens de programação para o desenvolvimento Android, como JavaScript até editores de texto IDE (*Integrated Development Environment*) como Android Studio Code.

1.1.1 Visual Studio Code

O *Visual Studio Code* é um *software* de edição de código-fonte desenvolvido pela *Microsoft* para *Windows*, *Mac OS* e *Linux*. Ele inclui suporte para depuração e *GitHub* integrado, inclui também realce de sintaxe. Além disso este *software* é customizável, contendo diversos *plugins* e temas para usar.



Recorri à utilização do *Visual Studio Code* para escrever todo o código da minha aplicação, pois é o editor de código que eu me sinto mais confortável e que me oferece imensas ferramentas que me facilitariam a realização deste projeto.

Figura 1 – Logótipo do Visual Studio Code

1.1.2 Figma

O *Figma* é um editor gráfico de vetores e protótipos de *designs* de projetos. Este é focado em desenvolvimento de experiência da interface com o utilizador, também permitindo a partilha de projetos e o desenvolvimento colaborativo.

Recorri à utilização do Figma para fazer o *design* das páginas e os ícones da minha aplicação, com a ajuda das suas ferramentas gráficas que me proporcionavam uma melhor performance.



Figura 2 – Logótipo do Figma

1.1.3 Photopea

O *Photopea* é um *Photoshop* online. É utilizado para a edição de imagens, criação de ilustrações e conversão de imagem para diferentes formatos. O *Photopea* armazena todos os ficheiros localmente e não carrega nenhum dado em um servidor.

Recorri à utilização do *Photopea* pois este traz uma vasta coleção de efeitos e ferramentas de design, nas quais eu usei para editar ícones e imagens da aplicação.



Figura 3 – Logótipo do Photopea

1.1.4 Android Studio

O *Android Studio* é um *software* (da *Google*) de desenvolvimento de aplicações *Android*. A tecnologia dele é baseada no *software IntelliJ*. Este contém diversos emuladores de diferentes dispositivos móveis da *Google* e podendo mudar a versão do Sistema Operativo.

Recorri à utilização do *Android Studio* para emular dispositivos móveis para testar a responsividade da aplicação, além de trazer inúmeros recursos que me ajudaram bastante.



Figura 4 – Logótipo do Android Studio

1.1.5 Firebase

A *Firebase* é um conjunto de serviços em nuvem de back-end e plataformas de desenvolvimento de aplicações proporcionado pela *Google*. Ele faz hospedagem de bases de dados, serviços, autenticação e integração para diversas aplicações.

Recorri à utilização da *Firebase* para utilizar como base de dados para a minha aplicação.



Figura 5 – Logótipo da Firebase

1.1.6 Expo Go

O Expo Go é uma aplicação móvel para Android e iOS (fornecida pelo React Native) de código aberto para simular a aplicação no próprio dispositivo móvel do seu utilizador sem ter de recorrer a diversas ferramentas para compilar a aplicação.

Recorri à utilização do Expo Go para simular a minha aplicação no meu telemóvel, facilitando o seu desenvolvimento



Figura 6 – Logótipo do Expo Go

1.1.7 Netlify

A Netlify é uma empresa de computação em nuvem remota, que oferece uma plataforma de desenvolvimento que inclui construção e implantação de serviços back-end sem servidor para aplicações web e sites dinâmicos.

Eu recorri à utilização do Netlify para ligar o meu projeto html à internet criando a minha página web.



Figura 7 – Logótipo do Netlify

~

1.2 Linguagem de programação

1.2.1 JavaScript

JavaScript é uma linguagem de programação estruturada em scripts. Esta torna as páginas Web interativas, logo é uma parte essencial no desenvolvimento de aplicações web. O JavaScript suporta estilos dirigidos a eventos (ações) e possui API's para trabalhar com textos, matrizes e datas.

Recorri à utilização do JavaScript como linguagem de programação

única da aplicação, tanto para o *frontend* como para o *backend*



Figura 8 – Logótipo do JavaScript

1.2.2 React Native

React Native é uma *framework* de *JavaScript* criada pelo Facebook é utilizado para a criação de aplicações Android, iOS e Web de forma nativa. Todo o código desenvolvido é convertido para a linguagem nativa do sistema operativo.

Recorri à utilização do React Native, para o desenvolvimento da maioria do projeto, pois parecia-me uma maneira interessante para

começar o desenvolvimento de aplicações móveis.

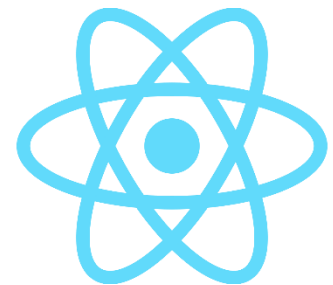


Figura 9 – Logótipo do React Native

1.2.3 HTML

HTML é uma linguagem de marcação utilizada na construção de páginas por documentos que podem ser interpretados por navegadores. Este é fundamental em todos os sites, uma vez que ele é a base.

Recorri à utilização do HTML para construir a estrutura minha página web juntamente com outra tecnologia que a complementa, o CSS.



Figura 10 – Logótipo do HTML

1.2.4 CSS

CSS é uma linguagem de estilos que complementa o HTML, este serve para dar estilos à página HTML. O CSS é muito importante para a criação de páginas web, pois este melhora a estética de qualquer site.

Eu recorri ao CSS devido à sua vasta coleção de estilos que eu me dispunha a usar, para criar a minha página web.



Figura 11 – Logótipo do CSS

2. Desenvolvimento

2.1 Pesquisa e Aprendizagem

Para realizar este projeto necessitei de fazer algumas pesquisas, para o tema, a linguagem a utilizar, a base de dados, após isso tomei a iniciativa de aprender a linguagem para este projeto, pois a mesma não foi lecionada nas aulas de programação. Pesquisei ferramentas que me pudessem auxiliar com o design/protótipo da aplicação, pesquisei outros designs online de aplicações de ginásio onde me inspirei para fazer o design. Visitei diversos sites que me ajudaram a realizar este projeto tal como o GitHub, neste site procurei outros projetos que me ajudaram a inspirar na realização de certos pormenores, YouTube, neste site visualizei vídeos que me auxiliaram no aprendizado/estudo da linguagem de programação utilizada.

2.1.1 Desafios

Durante a produção do meu projeto encontrei diversos desafios, tanto no começo como no final. O principal desafio foi a aprendizagem da *framework*, uma vez que era algo desconhecido eu tive que aprender sozinho, tendo ainda que gerir o tempo, pois não podia perder muito tempo só na aprendizagem, mas eu também tinha que avançar para a realização.

No começo do projeto tive algumas dificuldades nas importações de imagens e de algumas bibliotecas.

A meio do tempo de realização da PAP, o meu *backend* de Node.js deixou de funcionar subitamente, o que consequentemente me atrasou um pouco durante o tempo de realização. Por outro lado, mudei o meu *backend* para *Firebase*, o que me disponibilizou mais funções, como terminar sessão, impressão do email do utilizador, ligação ao *Firestore Database*, onde eu podia guardar diversas informações dividindo por coleções, ou seja, os dados ficam mais organizados.

Um desafio proposto por mim mesmo para a realização da aplicação foi manter uma estabilidade entre as funcionalidades e a responsividade da aplicação, ou seja, eu tentava manter o projeto com um aspeto agradável ao mesmo tempo que o deixava funcional, um exemplo disto foi na integração de caixas de texto, quando aparecia o teclado no ecrã o design desformava impossibilitando qualquer utilizador de manipular.

2.2 O que é GymApp?

GymApp é uma aplicação que tem como principal objetivo auxiliar um ginásio e o(s) seu(s) cliente(s), contendo diversas dicas fitness, desde receitas para cada tipo de dieta, como vídeos explicativos de exercícios de treino para cada parte do corpo. Esta aplicação também contém uma secção para marcação de aulas, onde o utilizador é notificado das aulas marcadas. Como extra a aplicação contém um *scan* de código qr que pode ser utilizado para entrada no ginásio ou outras operações.

2.3 Logótipo



Figura 9 – Logótipo do projeto (GymApp)

O logótipo do projeto traz um ambiente minimalista onde estão contidos o nome da app (logótipo em texto) e a sua simbolização (um peso). O nome do projeto provém do inglês *Gym* (Ginásio) + *App*

José Carlos da Cunha Rebelo 17

(Aplicação). O logo em imagem simboliza um peso, em uma forma *cartoon* e minimalista, sendo o peso um grande símbolo representante, e o mais popular, para a simbolização de um ginásio. A cor verde representa o bem-estar e saúde, que é uma das preocupações da aplicação ou de qualquer ginásio.

2.4 Design

O protótipo do design deste projeto foi iniciado em *Figma*, inspirado em diversos designs de outras pessoas disponibilizados na internet e em alguns casos inspirados em outras aplicações já existentes. O design é bastante simples, facilmente perceptível (o que era a ideia principal da aplicação) e muito organizado, levando em consideração usar as cores do logótipo na estrutura da aplicação.

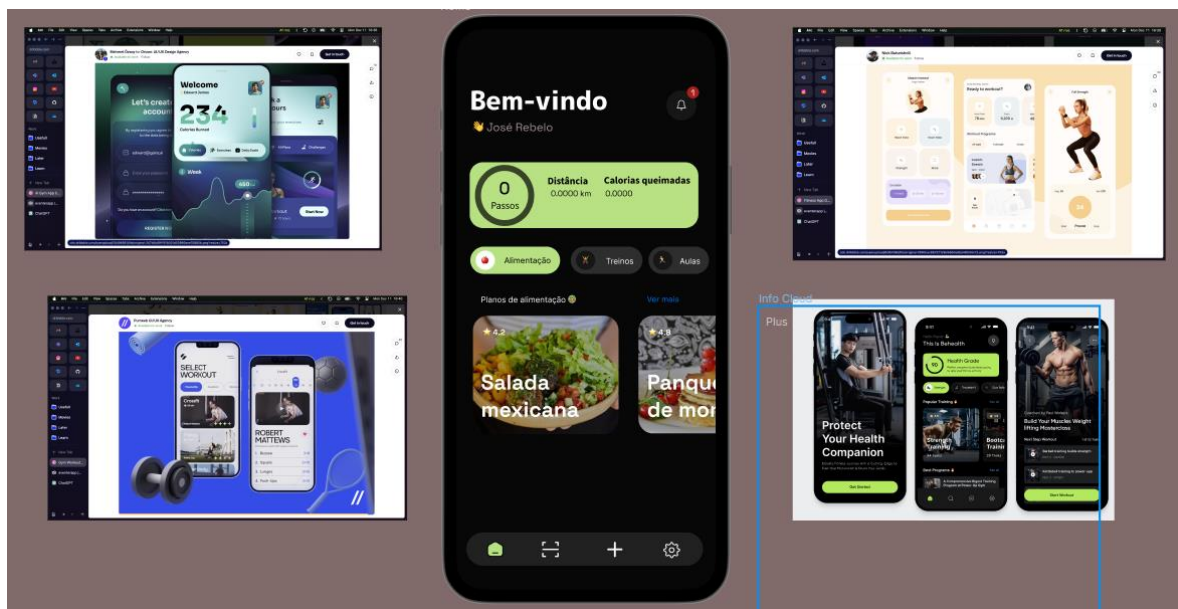
2.5 Código

O código em *React Native* (.js) é bastante simples para construir uma aplicação, o código é composto por *tags* (como em HTML), e contém inúmeras bibliotecas que facilitam a construção do projeto, podendo incluir código padrão do *JavaScript*, que ajuda a implementar a lógica que vai fazer a aplicação funcionar.

3. Versões da Aplicação

3.1 Esboços digitais

A melhor maneira de eu começar a aplicação era arranjar um esboço, onde eu ia replicar o seu design, então recorri ao Figma e comecei um esboço de como seria a aplicação, deixando ao lado os designs de inspiração.



3.2 Desenvolvimento das páginas

Criei o projeto em outubro, no começo tinha algumas dificuldades, uma vez que a linguagem era nova para mim. Tive de estudar um pouco sobre a framework e a linguagem, para depois começar a devida aplicação.

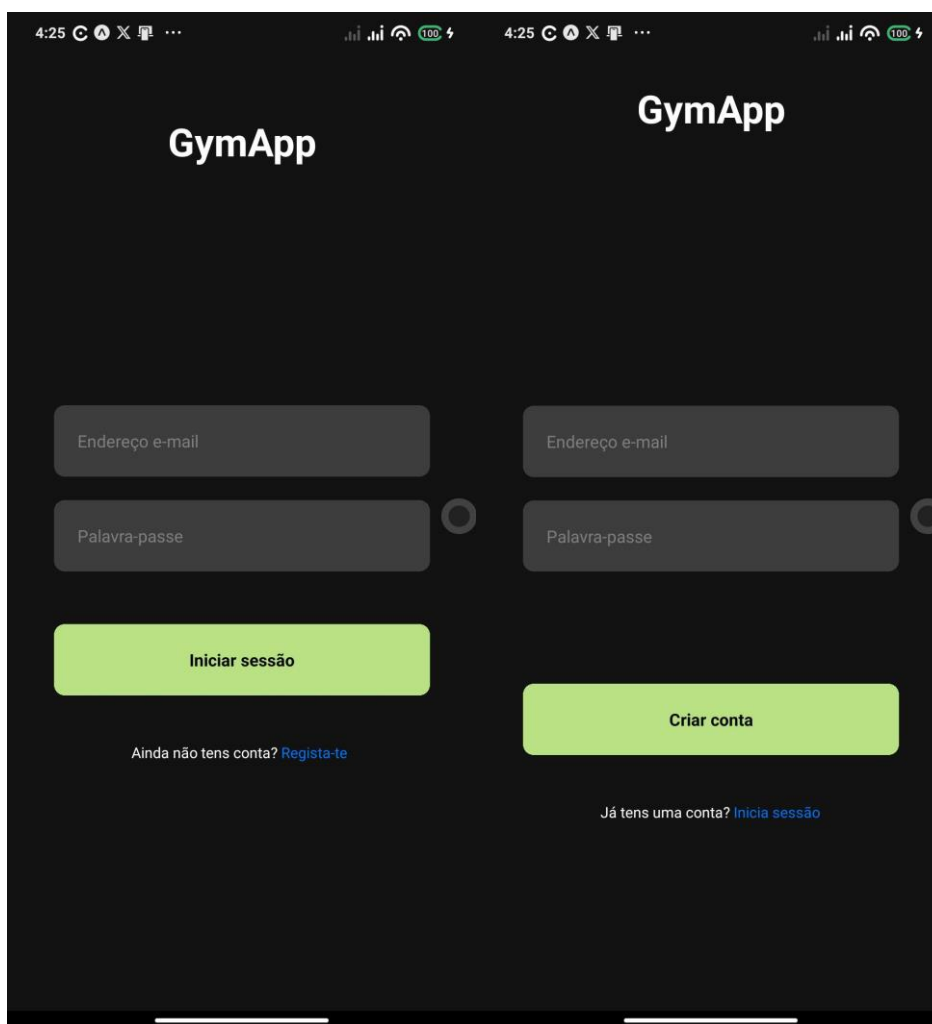
3.2.1 Página principal e página de receção

A primeira página que decidi fazer foi a principal (apesar de ter recebido diversas melhorias ao longo do tempo) e consequentemente desenvolvi uma página de receção ao utilizador.



3.2.2 Página de login e registo

Depois do desenvolvimento da página de receção e da página principal, decidi desenvolver as páginas que as ligavam, a página de login e a página de registo, no qual eu utilizei o firebase auth, onde implementei ao meu projeto um ficheiro de ligação à firebase. Com a implementação do firebase auth, tive que fazer uma lógica de renderização de páginas, em caso de o utilizador não estiver registado ou logado, não vai conseguir aceder às páginas principais da aplicação.



3.2.3 Página de scan

Sabendo que todas as aplicações de ginásio possuem um código qr, implementei uma biblioteca de código qr da própria framework, deixando-o numa página à parte, sem qualquer tipo de informação em volta que possa atrapalhar na hora de scan.



3.2.4 Página de definições

A página de definições contém as informações do utilizador (nome e email), tem opção de mudar foto do perfil, contém botão de terminar sessão, e contém outras páginas, como acerca do utilizador, que guarda os dados (idade, peso, altura e sexo), a página de contacto onde se pode encontrar as redes sociais do programador e por fim um botão de reportar bug, que está ligado ao meu email, onde as pessoas podem relatar problemas com a app.

4:39 4:50

Sobre ti

zecarebelo05
zecarebelo05@gmail.com

Acerca de ti >

Contacto >

Reportar bug >

Terminar sessão

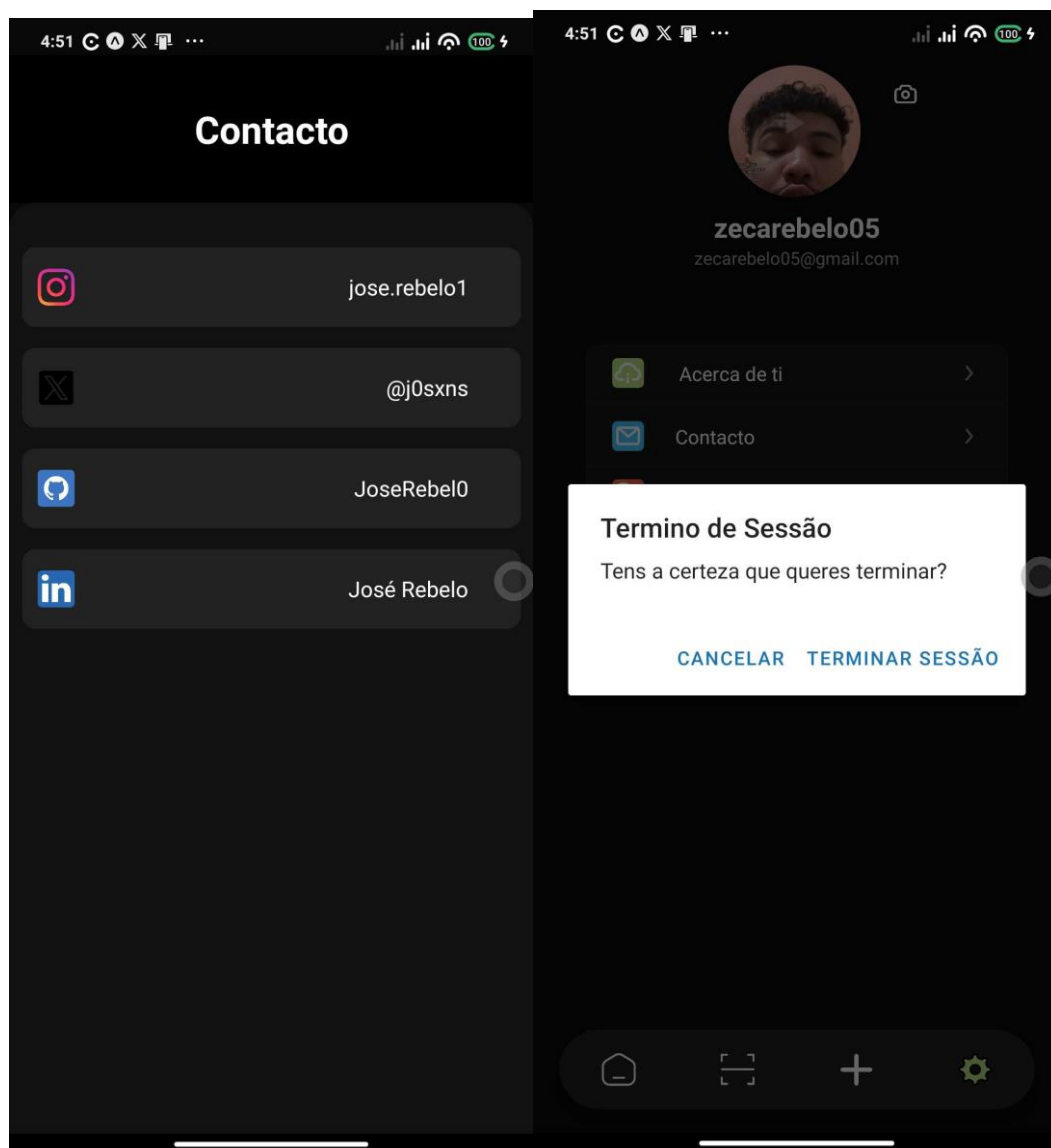
Idade (y): 19

Peso (kg): 60

Altura (m): 1,60

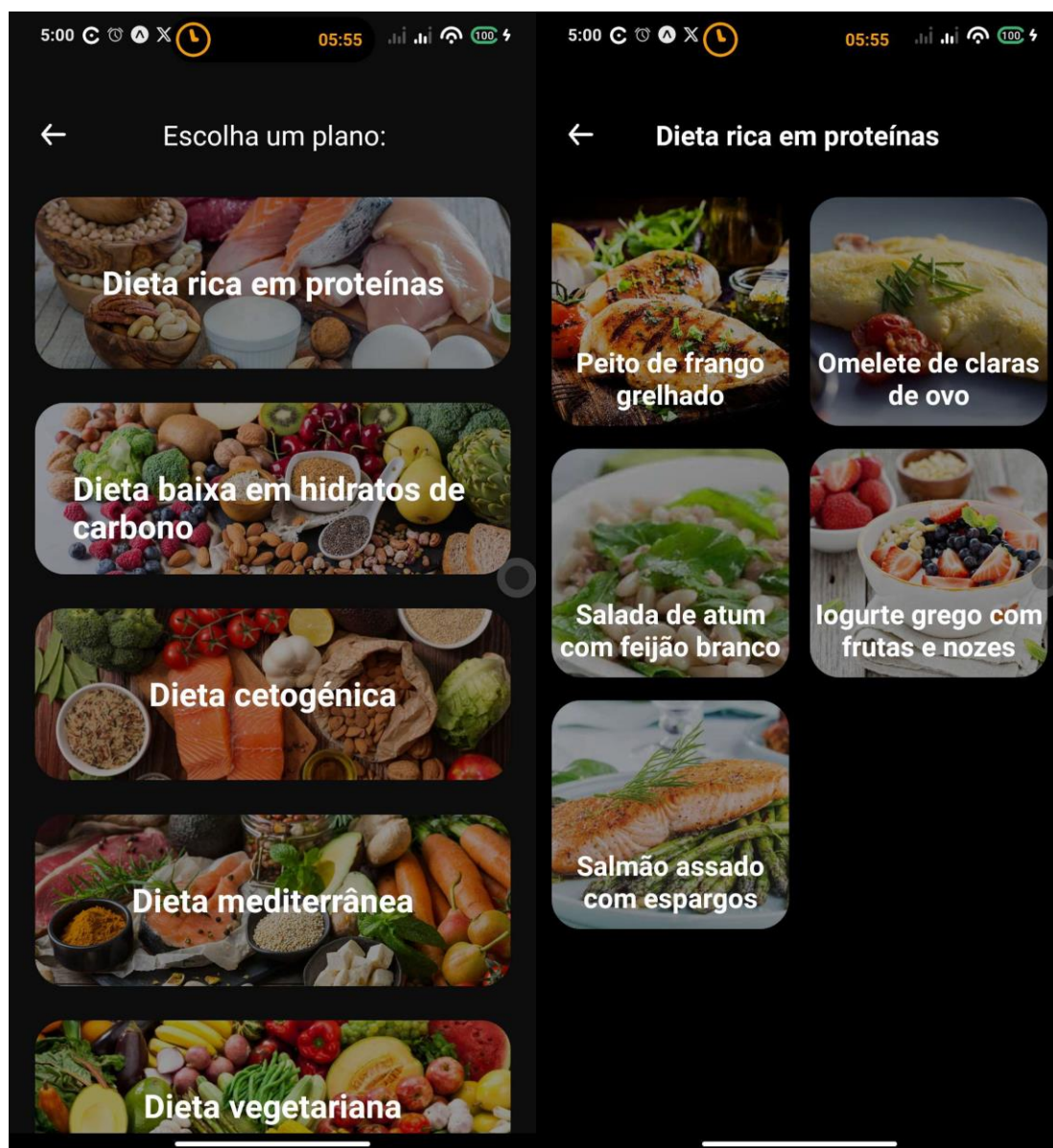
Sexo: ☒ Masculino ☐ Feminino

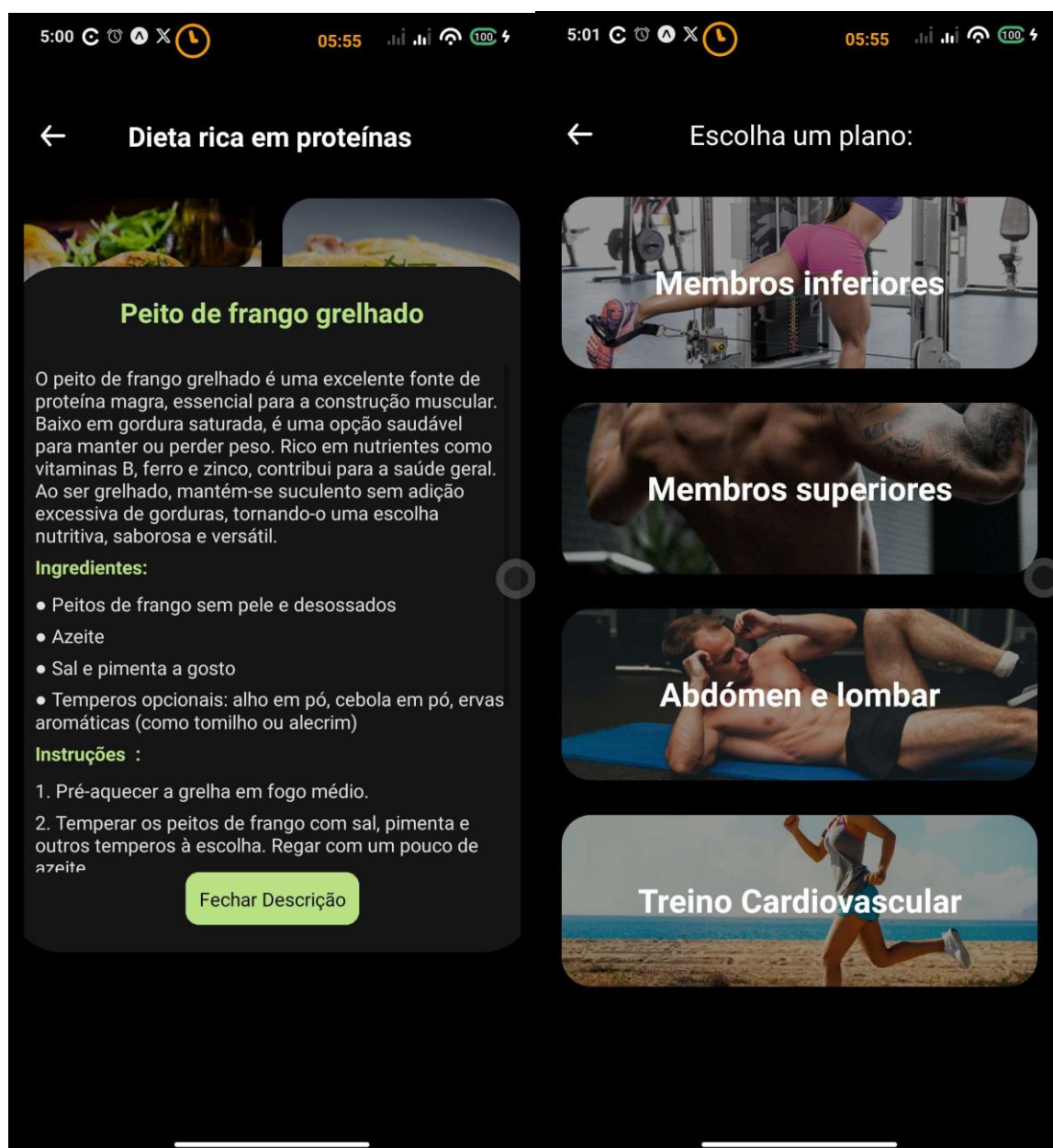
Guardar



3.2.5 Página Planos de alimentação/Planos de treino

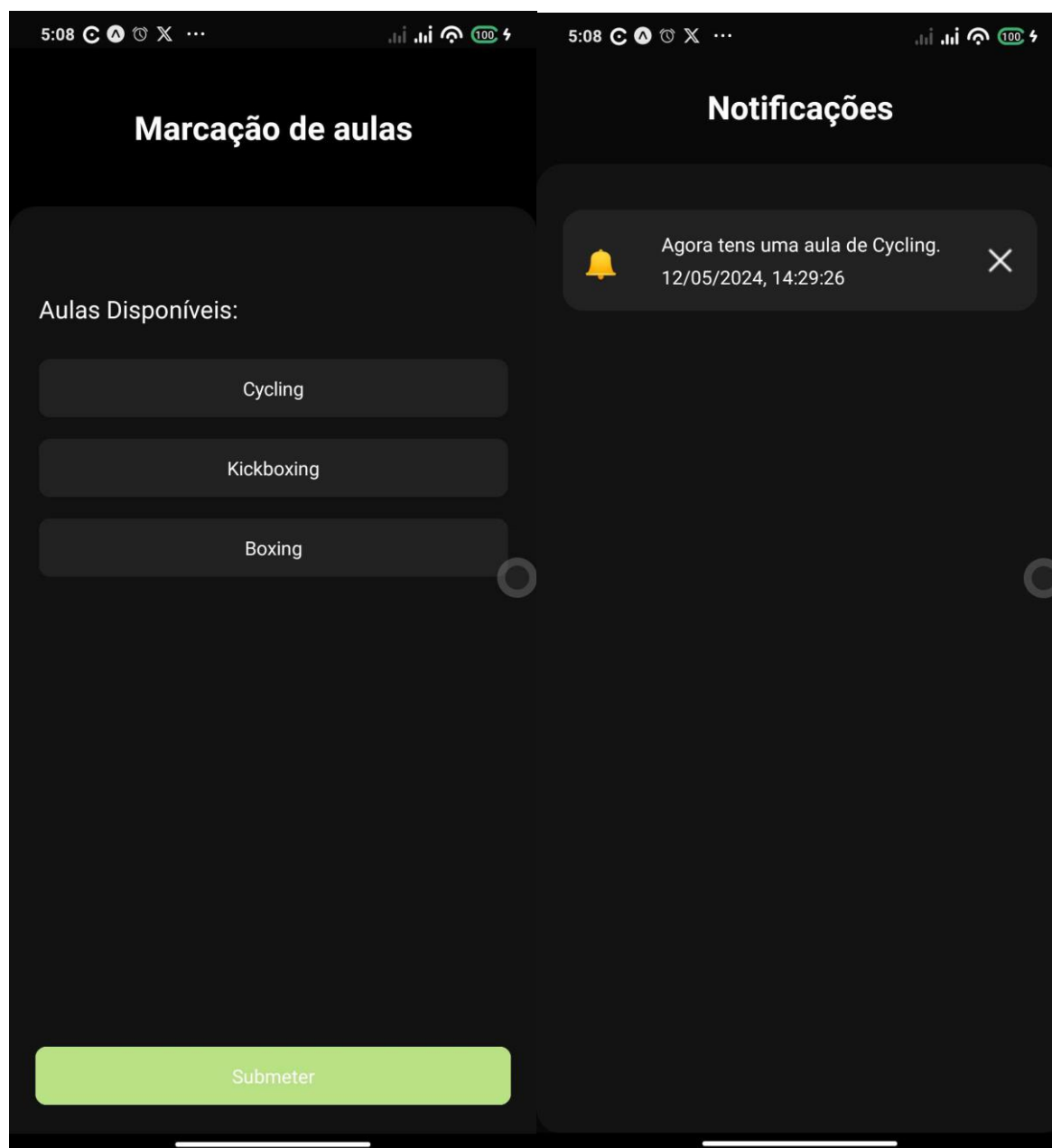
Para saber as receitas indicadas para cada tipo de dieta tive de fazer pesquisas na internet, após isso fui fazendo as páginas dedicadas a cada tipo de dieta com as receitas e o modo de preparo. Usei o mesmo modelo para a página dos planos de treino, mas por falta de tempo, não consegui concluir.





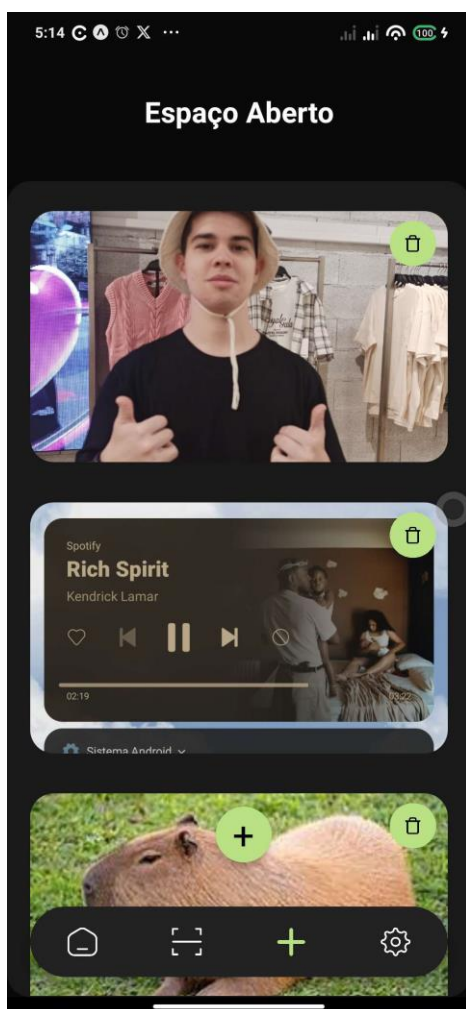
3.2.6 Página de marcação de aulas e página de notificações

Para fazer a marcação das aulas funcionar tive que recorrer ao firestore, onde armazenei as aulas que podiam ser agendadas e também guardei as aulas marcadas/notificações, que no final iriam aparecer na página de notificações.



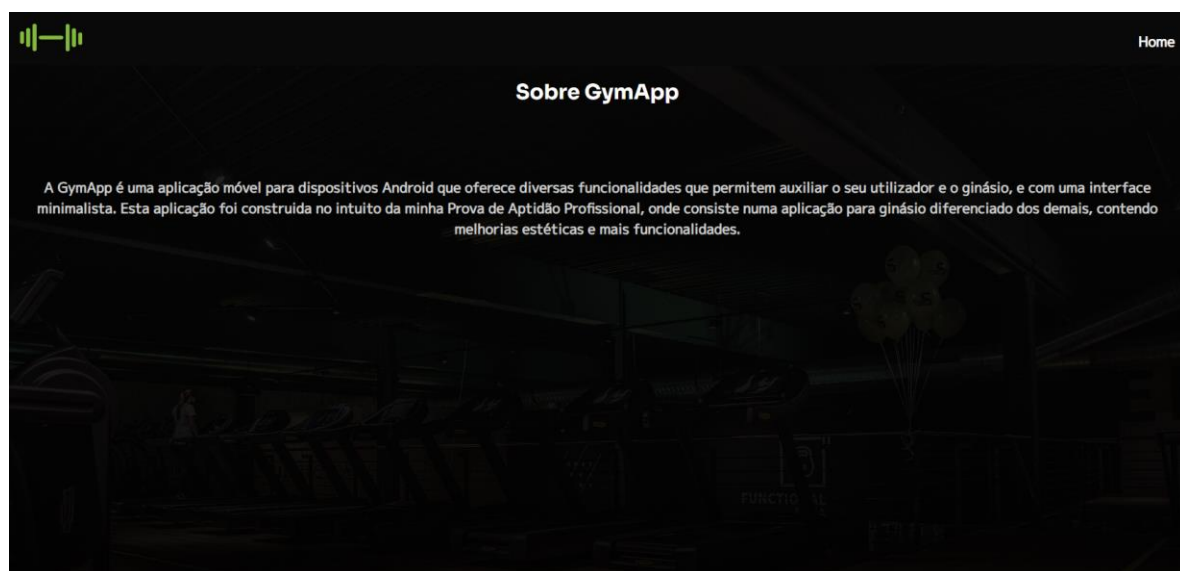
3.2.1 Página de espaço aberto

Esta página tinha como objetivo mostrar a grande evolução do seu utilizador, podendo tirar fotos e guardá-las, em caso de não gostar da foto ou ter colocado a foto por engano, o utilizador tem um botão individual para eliminar cada uma das fotos.



3.3 Página Web

A página web é bastante simples, é só uma página de entrada promocional à aplicação com um cabeçalho que contem o logo da aplicação e uma ligação para a página de sobre.



4. Código

4.1 Aplicação

4.1.1 Hall/index.js (Página de receção)

```
import React,{useEffect} from 'react';

import { StatusBar } from 'expo-status-bar';

import { Image, ImageBackground, StyleSheet, TouchableWithoutFeedback, Text, TouchableOpacity, View,} from 'react-native';

import backg from "../../assets/images/anastase-maragos-7kEpUPB8vNk-unsplash.png";

import { LinearGradient } from 'expo-linear-gradient';

import { useNavigation } from '@react-navigation/native';

import { SafeAreaView, useSafeAreaInsets, SafeAreaView } from 'react-native-safe-area-context';

import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

{/* import {useFonts, Inter_400Regular, Inter_700Bold, Inter_800ExtraBold} from '@expo-google-fonts/inter';

import {Poppins_400Regular, Poppins_700Bold, Poppins_800ExtraBold} from '@expo-google-fonts/poppins';

import {SpaceGrotesk_400Regular, SpaceGrotesk_700Bold} from '@expo-google-fonts/space-grotesk';

import {MPLUS1p_400Regular, MPLUS1p_700Bold, MPLUS1p_800ExtraBold} from '@expo-google-fonts/m-plus-1p';

import {NanumMyeongjo_400Regular, NanumMyeongjo_700Bold, NanumMyeongjo_800ExtraBold} from '@expo-google-fonts/nanum-myeongjo';*/}

import { useFonts } from 'expo-font';

export const Hall = () => {

  const navigation = useNavigation();

  const insets = useSafeAreaInsets();

  const [loaded] = useFonts({

    Inter: require('../../assets/fonts/Inter-Regular.ttf'),

    InterBold: require('../../assets/fonts/Inter-Bold.ttf'),

    SpaceGrotesk: require('../../assets/fonts/SpaceGrotesk-Regular.ttf'),
```

José Carlos da Cunha Rebelo

31

```

    })

    if (!loaded){
        return null;
    }

    return (
        <View style={[styles.container,{paddingTop: insets.top, paddingBottom: insets.bottom}]}>
            <ImageBackground source={backg} style = {styles.imageBackground}>
                <View style = {styles.title}>
                    <Text numberOfLines = {2} adjustsFontSizeToFit style = {{color: "white", fontSize: responsiveScreenFontSize(5),fontWeight:
                    'bold'}}>Vamos melhorar a tua saúde.</Text>

                    <Text style = {{color: "white", opacity: 0.5, marginTop: "4%}}>Eleva a tua jornada fitness com a motivação certa.</Text>
                </View>

                <TouchableOpacity hitSlop={{left: 50, right: 50 }} onPress={() => navigation.navigate('Login')} style={{flex:1, marginBottom:
                responsiveScreenWidth(5), marginTop: responsiveScreenHeight(15)}}>

                    <View style = {styles.startbutt}>
                        <Text style = {{color: "black", alignSelf: 'center', fontWeight: 'bold'}}>Vamos começar</Text>
                    </View>
                </TouchableOpacity>

                </ImageBackground>

                <StatusBar style="light" />
            </View>
        );
    }

```

```

const styles = StyleSheet.create({
    container: {
        flex: 1,

```

```
},
```

```
imageBackground: {
```

```
  flex: 1,
```

```
  width: '100%',
```

```
  resizeMode: 'cover'
```

```
},
```

```
title: {
```

```
  flex: 9,
```

```
  width: "85%",
```

```
  height: responsiveScreenHeight(5),
```

```
  alignItems: "center",
```

```
  justifyContent: 'flex-end',
```

```
  alignSelf: 'center',
```

```
},
```

```
button:{
```

```
  flex:1,
```

```
  margin: responsiveScreenWidth(5),
```

```
},
```

```
startbutt: {
```

```
  flex: 1,
```

```
  width: responsiveScreenWidth(80),
```

```
  height: responsiveScreenHeight(7),
```

```
  borderRadius: 50,
```

```
  backgroundColor: "#B8E183",
```

```
  alignItems: "center",
```

```
  alignSelf: 'center',
```

```
  justifyContent: 'center'
```

```
},  
});
```

4.1.2 Login/index.js (Página de login)

```
import React, { useState } from 'react';  
  
import { StatusBar } from 'expo-status-bar';  
  
import { KeyboardAvoidingView, TextInput, StyleSheet, TouchableOpacity, Text, View, TouchableWithoutFeedback, Alert } from 'react-native';  
  
import { useSafeAreaInsets } from 'react-native-safe-area-context';  
  
import { useNavigation } from '@react-navigation/native';  
  
import { responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from 'react-native-responsive-dimensions';  
  
import { signInWithEmailAndPassword } from 'firebase/auth';  
  
import auth from '../../config/firebase';  
  
import { getAuth } from 'firebase/auth';  
  
  
export const Login = () => {  
  
  const navigation = useNavigation();  
  
  const insets = useSafeAreaInsets();  
  
  const [email, setEmail] = useState("");  
  
  const [password, setPassword] = useState("");  
  
  
  const handleSubmit = async () => {  
  
    if (email && password){  
  
      try{  
  
        const authInstance = getAuth(auth);  
  
        await signInWithEmailAndPassword(authInstance, email, password);  
  
      }  
  
      catch(err){  
  
        console.log('got error: ',err.message);  
  
      }  
  
    }  
  
  }  
  
}
```

```
return (

<KeyboardAvoidingView behavior="height" style={{styles.container, { paddingTop: insets.top, paddingBottom: insets.bottom }}}>

  <View style={styles.title}>

    <Text style={{ color: 'white', fontSize: responsiveScreenFontSize(4), fontWeight: 'bold' }}>GymApp</Text>

  </View>

  <View style={styles.login}>

    <TextInput

      style={styles.username}

      placeholder="Endereço e-mail"

      placeholderTextColor={grey}

      value={email}

      onChangeText={value => setEmail(value)}

    />

    <TextInput

      secureTextEntry

      style={styles.password}

      placeholder="Palavra-passe"

      placeholderTextColor={grey}

      value={password}

      onChangeText={value => setPassword(value)}

    />

  </View>

  <TouchableOpacity onPress={handleSubmit}>

    <View style={styles.loginbutt}>

      <Text style={{ color: 'black', fontWeight:'bold' }}>Iniciar sessão</Text>

    </View>

  </TouchableOpacity>

)
```

```

<View style={{ alignItems: 'center', marginTop: responsiveScreenHeight(0) }}>

  <View style={styles.signupbutt}>

    <Text style={{ color: 'white', fontSize: responsiveScreenFontSize(1.5) }}>Ainda não tens conta? </Text>

    <TouchableOpacity onPress={() => navigation.navigate('SignUp')}>

      <Text style={{ color: '#0075FF', fontSize: responsiveScreenFontSize(1.5) }}>Regista-te</Text>

    </TouchableOpacity>

  </View>

</View>

<StatusBar style="light" />

</KeyboardAvoidingView>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#121212',

  },

  title: {

    width: '100%',

    height: '30%',

    alignItems: 'center',

    marginTop: '15%',

  },

  username: {

    width: '80%',

    height: responsiveScreenHeight(7),

```

```
borderRadius: 9,  
  
backgroundColor: '#3d3d3d',  
  
paddingLeft: '5%',  
  
color: 'white',  
  
},  
  
password: {  
  
  width: '80%',  
  
  height: responsiveScreenHeight(7),  
  
  borderRadius: 9,  
  
  backgroundColor: '#3d3d3d',  
  
  paddingLeft: '5%',  
  
  color: 'white',  
  
  marginTop: '5%',  
  
},  
  
login: {  
  
  width: '100%',  
  
  height: responsiveScreenHeight(16.5),  
  
  alignItems: 'center',  
  
},  
  
loginbutt: {  
  
  width: '80%',  
  
  height: responsiveScreenHeight(7),  
  
  borderRadius: 9,  
  
  backgroundColor: '#B8E183',  
  
  alignSelf: 'center',  
  
  alignItems: 'center',  
  
  justifyContent: 'center',  
  
  marginTop: responsiveScreenHeight(5),
```

```

},

signupbutt: {

  width: '47%',

  height: responsiveScreenHeight(2),

  alignSelf: 'center',

  alignItems: 'center',

  marginTop: '10%',

  flexDirection: 'row',

},

});

```

4.1.3 SignUp/index.js (Página de registo)

```

import React, { useState } from 'react';
import { StatusBar } from 'expo-status-bar';
import { KeyboardAvoidingView, StyleSheet, Text, TextInput, TouchableOpacity, View, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native';
import { responsiveScreenFontSize, responsiveScreenHeight } from 'react-native-responsive-dimensions';
import { createUserWithEmailAndPassword } from 'firebase/auth';
import auth from '.././././config/firebase';
import { getAuth } from 'firebase/auth';

export const SignUp = () => {
  const navigation = useNavigation();
  //const [username, setUsername] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = async () => {
    if (email && password) {
      try {
        const authInstance = getAuth(auth); // Use a função getAuth para obter a instância correta de auth
        await createUserWithEmailAndPassword(authInstance, email, password);
      } catch (err) {
        console.log('got error: ', err.message);
      }
    }
  };

  return (
    <KeyboardAvoidingView behavior="height" style={styles.container}>

```

```
<View style={styles.title}>
  <Text style={styles.titleText}>GymApp</Text>
</View>
<View style={styles.signup}>

  <TextInput
    style={{styles.input, { marginTop: '5%' }}}
    placeholder='Endereço e-mail'
    placeholderTextColor={'grey'}
    value={email}
    onChangeText={value => setEmail(value)}
  />
  <TextInput
    secureTextEntry
    style={{styles.input, { marginTop: '5%' }}}
    placeholder='Palavra-passe'
    placeholderTextColor={'grey'}
    value={password}
    onChangeText={value => setPassword(value)}
  />
</View>
<TouchableOpacity onPress={handleSubmit}>
  <View style={styles.loginButton}>
    <Text style={styles.loginButtonText}>Criar conta</Text>
  </View>
</TouchableOpacity>
<View style={styles.signupButton}>
  <Text style={styles.signupButtonText}>Já tens uma conta? </Text>
  <TouchableOpacity onPress={() => navigation.navigate('Login')}>
    <Text style={styles.signupButtonLink}>Inicia sessão</Text>
  </TouchableOpacity>
</View>
<StatusBar style='light' />
</KeyboardAvoidingView>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#121212',
  },
  title: {
    width: '100%',
    height: '30%',
    alignItems: 'center',
    marginTop: '15%',
  },
  titleText: {
    color: 'white',
    fontSize: responsiveScreenFontSize(4),
    fontWeight: 'bold'
  }
});
```

```
},
input: {
  width: '80%',
  height: responsiveScreenHeight(7),
  borderRadius: 9,
  backgroundColor: '#3d3d3d',
  paddingLeft: '5%',
  color: 'white',
},
signup: {
  width: '100%',
  height: responsiveScreenHeight(25),
  alignItems: 'center',
  marginTop: 0,
},
loginButton: {
  width: '80%',
  height: responsiveScreenHeight(7),
  borderRadius: 9,
  backgroundColor: '#B8E183',
  alignSelf: 'center',
  alignItems: 'center',
  justifyContent: 'center',
  marginTop: '10%',
},
loginButtonText: {
  color: 'black',
  fontWeight: 'bold'
},
signupButton: {
  width: '47%',
  height: responsiveScreenHeight(2),
  alignSelf: 'center',
  alignItems: 'center',
  marginTop: '10%',
  flexDirection: 'row',
},
signupButtonText: {
  color: 'white',
  fontSize: responsiveScreenFontSize(1.5),
},
signupButtonLink: {
  color: '#0075FF',
  fontSize: responsiveScreenFontSize(1.5),
},
});

export default SignUp;
```

4.1.4 Home/index.js (Página principal)

```
import React from 'react';
import { StatusBar } from 'expo-status-bar';
import { Image, StyleSheet, View, Text, TouchableOpacity, ScrollView, FlatList, TouchableWithoutFeedback, } from 'react-native';
import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';
import Bell from "../assets/images/iconbell.png";
import salada from "../assets/images/salada.png";
import smoothie from "../assets/images/smoothie.png";
import flexoes from "../assets/images/flexoes.jpg";
import supino from "../assets/images/supino.png";
import { SafeAreaProvider, useSafeAreaInsets, SafeAreaView } from 'react-native-safe-area-context';
import { useNavigation } from '@react-navigation/native';
import { useState } from 'react';
import { useEffect } from 'react';
import { Pedometer } from 'expo-sensors';
import { auth } from '../config/firebase';
import { PermissionsAndroid } from 'react-native';
import CircularProgress from 'react-native-circular-progress-indicator';
//import { Platform } from 'react-native/types';

export const Home = ({ route }) => {

  //const { username } = route.params || {}; value
  const navigation = useNavigation();
  const insets = useSafeAreaInsets();

  const [showAlimentacaoContent, setShowAlimentacaoContent] = useState(false);
  const [selectedButton, setSelectedButton] = useState(null);

  const [PedometerAvailability, setPedometerAvailability] = useState("");
  const [stepCount, updateStepCount] = useState(0);

  var Dist = stepCount/1300;
  var DistanceCovered = Dist.toFixed(4);

  var cal = DistanceCovered * 60;
  var caloriesBurnt = cal.toFixed(4);

  useEffect(() => {
    subscribe();
  }, [])

  subscribe = () => {
    const subscription = Pedometer.watchStepCount((result) => {
```

```

    updateStepCount(result.steps);
  })

Pedometer.isAvailableAsync().then(
  (result) => {
    setPedometerAvailability(String(result));
  },
  (error) => {
    setPedometerAvailability('Could not get isAvailableAsync: ' + error);
  }
)
}

const [userEmail, setUserEmail] = useState(""); // Estado do email do utilizador

useEffect(() => {
  // Vai buscar o email do utilizador ao firebase
  const unsubscribe = auth.onAuthStateChanged((user) => {
    if (user) {
      setUserEmail(user.email);
    } else {
      setUserEmail(""); // Limpar email, se não estiver registado
    }
  });
});

return () => unsubscribe();
}, []);

const handleButtonClick = (buttonName) => {
  setSelectedButton(buttonName);
};

const isButtonSelected = (buttonName) => {
  return selectedButton === buttonName;
};

const getButtonBackgroundColor = (buttonName) => {
  return isButtonSelected(buttonName) ? '#B8E183' : '#2E2E2E';
};

const getIconBackgroundColor = (buttonName) => {
  return isButtonSelected(buttonName) ? 'white' : '#171717';
};

const getTextStyles = (buttonName) => {
  return isButtonSelected(buttonName) ? 'black' : 'white';
};

useEffect(() => {
  // Simular pressionamento do botão após 1 milissegundo (ajuste conforme necessário)
  const timer = setTimeout(() => {

```

```

    handleClick('alimentacao');
  }, 1);

  // Limpar o temporizador para evitar efeitos colaterais
  return () => clearTimeout(timer);
}, []);

const renderContent = () => {
  switch (selectedButton) {
    case 'alimentacao':
      return (
        <View>
          <View style={{paddingRight: responsiveScreenWidth(6),paddingLeft: responsiveScreenWidth(6), flexDirection:
'row',justifyContent: 'space-between'}}>
            <Text style={{ color: 'white'}}>Planos de alimentação 🍽️</Text>
            <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>
              <Text style={{ color: 'white', fontWeight: 'bold',fontSize: responsiveScreenFontSize(2) }}>Ver mais</Text>
            </TouchableWithoutFeedback>
          </View>
          <ScrollView horizontal showsHorizontalScrollIndicator={false} style={{marginTop: responsiveScreenHeight(2.5)}}>
            <View style={{flexDirection: 'row', gap: responsiveScreenWidth(3)}}>

              <View style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20), alignContent:'center',
paddingTop: responsiveScreenHeight(1), paddingLeft: responsiveScreenWidth(3), marginRight:
responsiveScreenWidth(3)}}>
                <Image source={salada} style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20),
borderRadius: 22, position: 'absolute'}}/>
                <View style={styles.overlay} />
                <Text style={{color:'white', alignSelf:'flex-start'}}>★ 4.2</Text>
                <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3),
marginTop: responsiveScreenHeight(5.5)}}>Salada de quinoa</Text>
              </View>

              <View style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20), alignContent:'center',
paddingTop: responsiveScreenHeight(1), paddingLeft: responsiveScreenWidth(3)}}>
                <Image source={smoothie} style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20),
borderRadius: 22, position: 'absolute',}}/>
                <View style={styles.overlay} />
                <Text style={{color:'white', alignSelf:'flex-start'}}>★ 4.8</Text>
                <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3),
marginTop: responsiveScreenHeight(5.5)}}>Smoothie de proteínas</Text>
              </View>

            </View>
          </ScrollView>
        </View>
      );
    case 'treinos':
      return (
        <View>

```

```

    <View style={{paddingRight: responsiveScreenWidth(6),paddingLeft: responsiveScreenWidth(6), flexDirection:
'row',justifyContent: 'space-between'}}>
      <Text style={{ color: 'white' }}>Planos de treino <img alt="lock icon" data-bbox="488 151 505 165"/></Text>
      <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuTreinos')}>
        <Text style={{ color: 'white', fontWeight: 'bold', fontSize: responsiveScreenFontSize(2)}}>Ver mais</Text>
      </TouchableWithoutFeedback>
    </View>
    <ScrollView horizontal showsHorizontalScrollIndicator={false} style={{marginTop: responsiveScreenHeight(2.5)}}>
      <View style={{flexDirection: 'row', gap: responsiveScreenWidth(3)}}>

        <View style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20), alignContent:'center',
paddingTop: responsiveScreenHeight(1), paddingLeft: responsiveScreenWidth(3), marginRight:
responsiveScreenWidth(3)}}>
          <Image source={flexoes} style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20),
borderRadius: 22, position: 'absolute'}}/>
          <View style={styles.overlay} />
          <Text style={{color:'white', alignSelf:'flex-start'}}>★ 4.9</Text>
          <Text style={{color:'white', alignSelf:'center', fontWeight:'bold', fontSize: responsiveScreenFontSize(3),
marginTop: responsiveScreenHeight(5.5)}}>Flexões</Text>
        </View>

        <View style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20), alignContent:'center',
paddingTop: responsiveScreenHeight(1), paddingLeft: responsiveScreenWidth(3)}}>
          <Image source={supino} style={{width: responsiveScreenWidth(50), height: responsiveScreenHeight(20),
borderRadius: 22, position: 'absolute'}}/>
          <View style={styles.overlay} />
          <Text style={{color:'white', alignSelf:'flex-start'}}>★ 5.0</Text>
          <Text style={{color:'white', alignSelf:'center', fontWeight:'bold', fontSize: responsiveScreenFontSize(3),
marginTop: responsiveScreenHeight(5.5)}}>Supino</Text>
        </View>

      </View>
    </ScrollView>
  </View>
);
case 'aulas':
  return (
    <View>
      <View style={{ paddingRight: responsiveScreenWidth(6), paddingLeft: responsiveScreenWidth(6), flexDirection:
'row', justifyContent: 'space-between' }}>
        <Text style={{ color: 'white' }}>Aqui vem o horário</Text>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAulas')}>
          <Text style={{ color: 'white', fontWeight: 'bold', fontSize: responsiveScreenFontSize(2) }}>Marcar aulas</Text>
        </TouchableWithoutFeedback>
      </View>
    </View>
  );
default:
  return null;
}

```

```
};

return (
  <SafeAreaView style={styles.container}>
    <View style={{flexDirection: 'row', justifyContent: 'space-between', paddingTop: '5%', width: '100%', marginTop:
responsiveScreenHeight(2)}}>
      <View style={styles.title}>
        <Text style={{color: 'white', fontWeight: 'bold', fontSize: responsiveScreenFontSize(5),}}>Bem-vindo(a)</Text>
      <View style={{flexDirection: 'row',}}>
        <Text style={{fontSize: responsiveScreenFontSize(2)}}>👋 </Text>
        <Text style={{fontSize: responsiveScreenFontSize(2), color: "#A4A4A4"}}>{userEmail.split('@')[0].replace(/\./g,
'')}</Text>
      </View>
    </View>

    <TouchableOpacity onPress={() => navigation.navigate('Notifications')}>
      <View style={styles.notbutton}>
        <View style={styles.circle}></View>
        <Image source={Bell} style={{width: responsiveScreenWidth(4), height: responsiveScreenHeight(4), resizeMode:
'contain', alignSelf: "center", position: "absolute", }} />
      </View>
    </TouchableOpacity>
  </View>

  <View style = {styles.healthtable}>
    <View style={{alignItems: 'center', justifyContent: 'center', marginTop: responsiveScreenHeight(0.5)}}>
      <CircularProgress
        value={stepCount}
        maxValue={6500}
        radius={45}
        activeStrokeColor='black'
        inactiveStrokeColor='black'
        inactiveStrokeOpacity={0.5}
        inactiveStrokeWidth={10}
        activeStrokeWidth={10}
        title='Passos'
      />
    </View>
    <View style={{marginTop: responsiveHeight(2)}}>
      <Text style={{fontWeight: 'bold'}}>Distância</Text>
      <Text>{DistanceCovered} km</Text>
    </View>

    <View style={{marginTop: responsiveHeight(2)}}>
      <Text style={{fontWeight: 'bold'}}>Calorias queimadas</Text>
      <Text>{caloriesBurnt}</Text>
    </View>
  </View>
  <View style = {styles.healthcircle}>
    <Text style = {{fontWeight: "500", fontSize: responsiveScreenFontSize(3), alignSelf: "center"}}>90</Text>
  </View>
);
```

```

</View> */}
</View>

<View style={{width: '100%', height: responsiveScreenHeight(20), paddingTop:'20%', marginTop:
responsiveScreenHeight(-5), marginBottom: responsiveHeight(1)}}>
  <ScrollView horizontal showsHorizontalScrollIndicator={false}>
    <View style = {styles.buttons}>

      <TouchableWithoutFeedback onPress={() => handleButtonClick('alimentacao')}>
        <View style={{width: responsiveScreenWidth(40), height: responsiveScreenHeight(5.9), borderRadius: 50,
        backgroundColor: isButtonSelected('alimentacao') ? '#B8E183' : '#2E2E2E', alignItems: 'center', flexDirection:'row',
        justifyContent: 'space-between', paddingRight: responsiveScreenWidth(2), paddingLeft: responsiveScreenWidth(1.5)}}>
          <View style={{width: responsiveScreenWidth(10.3), height: responsiveScreenHeight(4.5), backgroundColor:
          getIconBackgroundColor('alimentacao'), borderRadius: 9999, alignItems: 'center', justifyContent: 'center'}}>
            <Text>🍷</Text>
          </View>
          <Text style={{marginRight: responsiveScreenWidth(1),color: getTextStyles('alimentacao')}}>Alimentação</Text>
        </View>
      </TouchableWithoutFeedback>

      <TouchableWithoutFeedback onPress={() => handleButtonClick('treinos')}>
        <View style={{width: responsiveScreenWidth(32), height: responsiveScreenHeight(5.9), borderRadius: 40,
        backgroundColor: isButtonSelected('treinos') ? '#B8E183' : '#2E2E2E', alignItems: 'center', flexDirection:'row',
        justifyContent: 'space-between', paddingRight: responsiveScreenWidth(3), paddingLeft: responsiveScreenWidth(1.5)}}>
          <View style={{width: responsiveScreenWidth(10.3), height: responsiveScreenHeight(4.5), backgroundColor:
          getIconBackgroundColor('treinos'), borderRadius: 50, alignItems: 'center', justifyContent: 'center'}}>
            <Text>🏋️</Text>
          </View>
          <Text style={{color: getTextStyles('treinos')}}>Treinos</Text>
        </View>
      </TouchableWithoutFeedback>

      <TouchableWithoutFeedback onPress={() => handleButtonClick('aulas')}>
        <View style={{width: responsiveScreenWidth(30), height: responsiveScreenHeight(5.9), borderRadius: 40,
        backgroundColor: isButtonSelected('aulas') ? '#B8E183' : '#2E2E2E', alignItems: 'center', flexDirection:'row',
        justifyContent: 'space-between', paddingRight: responsiveScreenWidth(3), paddingLeft: responsiveScreenWidth(1.5)}}>
          <View style={{width: responsiveScreenWidth(10.3), height: responsiveScreenHeight(4.5), backgroundColor:
          getIconBackgroundColor('aulas'), borderRadius: 50, alignItems: 'center', justifyContent: 'center'}}>
            <Text>👩</Text>
          </View>
          <Text style={{color: getTextStyles('aulas')}}>Aulas</Text>
        </View>
      </TouchableWithoutFeedback>
    </ScrollView>
  </View>

  {renderContent()}

  <StatusBar style="auto" />
</SafeAreaView>

```

```
);  
};  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: '#090909',  
  },  
  
  title: {  
    width: responsiveScreenWidth(75),  
    height: responsiveScreenHeight(9),  
    //backgroundColor: 'tomato',  
    paddingLeft: '5%',  
  },  
  
  circle: {  
    width: responsiveWidth(15),  
    height: responsiveHeight(6),  
    aspectRatio: 1,  
    borderRadius: 30,  
    backgroundColor: "#333333",  
    alignSelf: "center",  
  },  
  
  notbutton: {  
    width: responsiveScreenWidth(7),  
    height: responsiveScreenHeight(7),  
    marginRight: responsiveScreenWidth(6),  
    alignItems: 'center',  
    justifyContent: 'center',  
    //backgroundColor: 'pink',  
  },  
  
  toppage: {  
    flexDirection: 'row',  
    alignItems: 'center',  
    marginTop: "5%",  
    //backgroundColor: 'red',  
    width: '100%',  
    height: responsiveFontSize(15),  
  },  
  
  buttons:{  
    width: '105%',  
    height: '45%',  
    //backgroundColor: 'red',  
    flexDirection: 'row',  
    paddingLeft: responsiveScreenWidth(2),  
    gap: responsiveScreenWidth(3),  
  },  
});
```

```
healthtable: {
  width: responsiveWidth(90),
  height: responsiveHeight(13),
  backgroundColor: "#B8E183",
  borderRadius: 20,
  alignSelf: "center",
  justifyContent: 'space-between',
  marginTop: responsiveScreenHeight(7),
  paddingBottom: responsiveScreenHeight(0.5),
  flexDirection: 'row',
  paddingLeft: responsiveScreenWidth(2),
  paddingRight: responsiveScreenWidth(5),
},

scrollcircle: {
  width: responsiveWidth(9),
  height: responsiveHeight(6),
  aspectRatio: 1,
  borderRadius: 30,
  backgroundColor: "#333333",
  alignSelf: "center",
},

overlay: {
  ...StyleSheet.absoluteFillObject,
  backgroundColor: 'rgba(0, 0, 0, 0.4)', // Ajuste a opacidade conforme necessário
  borderRadius: 22,
},

spacebtw:{
  marginRight: responsiveScreenWidth(2),
},

titulo: {
  fontSize: 24,
  fontWeight: 'bold',
  marginBottom: 10,
},

diaDaSemana: {
  fontSize: 20,
  marginBottom: 20,
},

atividade: {
  fontSize: 16,
  marginBottom: 8,
},

smalltext:{
  width: responsiveScreenWidth(50),
```

```
marginLeft: responsiveScreenWidth(33),
}
});
```

4.1.5 Scan/index.js (Página de Scan)

```
import React from 'react';
import { StatusBar } from 'expo-status-bar';
//import QRCode from '../assets/images/qr-code.png';
import QRCode from 'react-native-qrcode-svg';
import backg from '../assets/images/fundoscan4.png';
import seta from '../assets/images/arrowLeft.png';
import { Image, StyleSheet, ImageBackground, View, Text, Dimensions, TouchableOpacity, ScrollView, FlatList } from 'react-native';
import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight,
responsiveScreenFontSize, useResponsiveScreenHeight } from 'react-native-responsive-dimensions';
import { SafeAreaView } from 'react-native-safe-area-context';

export const Scan = () => {
  return (
    <SafeAreaView style={styles.container}>
      <Text style={{color: 'white', fontSize: responsiveScreenFontSize(3), fontWeight: 'bold', alignSelf: 'center', marginTop:
responsiveScreenHeight(5), marginBottom: responsiveScreenHeight(5)}}>Scan</Text>
      <View style={styles.centeredView}>
        <Text style={{color: 'white', fontSize: responsiveScreenFontSize(2), marginBottom: responsiveScreenHeight(4)}}>Aqui será gerado o
seu código QR:</Text>
        <QRCode value='https://www.instagram.com/jose.rebelo1/' style={{width: responsiveScreenWidth(55), height:
responsiveScreenHeight(25)}}/>
        {/*<Image source={QRCode} style={{width: responsiveScreenWidth(55), height: responsiveScreenHeight(25), borderRadius: 15,
resizeMode: 'contain'}}></Image>*/}
      </View>
      <StatusBar style="light" />
    </SafeAreaView>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#090909',
  },
  centeredView: {
    width: responsiveScreenWidth(100),
    height: responsiveScreenHeight(80),
    backgroundColor: '#171717',
    alignItems: 'center',
    justifyContent: 'center',
    paddingBottom: responsiveScreenHeight(15),
    borderRadius: 20,
  },
});
```

4.1.6 Settings/index.js (Página de Definições)

```
import React from 'react';
import { useState, useEffect } from 'react';
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, View, Text, Image, TouchableOpacity, TouchableWithoutFeedback, Alert, Linking } from 'react-native';
import seta from "../assets/images/arrowLefticon.png";
import ProfileP from "../assets/images/emptyprofile1.png";
import flag from "../assets/images/flag.png";
import arrowR from "../assets/images/arrowRight.png";
import mail from "../assets/images/mail.png";
import logout from "../assets/images/logout.png";
import about from "../assets/images/about.png";
import camera from "../assets/images/camera.png";
import check from "../assets/images/check.png";
import { SafeAreaView, SafeAreaView, useSafeAreaInsets } from 'react-native-safe-area-context';
import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';
import { useNavigation } from '@react-navigation/native';
import { useFonts } from 'expo-font';
import { signOut } from 'firebase/auth';
import { auth } from '../config/firebase';
import { getAuth } from 'firebase/auth';
import * as ImagePicker from 'expo-image-picker';
import { firebase } from '../config/firebase';
import { getStorage, ref, app, uploadBytes } from 'firebase/storage';
import { Profile } from '../screens/Profile';

export const Settings = () => {

  const [items, setItems] = useState([]);
  const [open, setOpen] = useState(false);

  const [image, setImage] = useState(null)
  const [uploading, setUploading] = useState(false)

  const [userEmail, setUserEmail] = useState(""); // State to store user's email

  useEffect(() => {
    // Fetch user's email from Firebase
    const unsubscribe = auth.onAuthStateChanged((user) => {
      if (user) {
        setUserEmail(user.email);
      } else {
        setUserEmail(""); // Clear email if user is not signed in
      }
    });
  });
```

```
// Cleanup function to unsubscribe when the component unmounts
return () => unsubscribe();
}, []);
```

```
const checkUserImages = async (uid) => {
  const storage = getStorage();
  const imagesRef = ref(storage, `profile/${uid}`);
  try {
    const imageList = await listAll(imagesRef);
    setUserHasImage(imageList.items.length > 0);
  } catch (error) {
    console.error('Error checking user images:', error);
  }
};
```

```
const [selectedButton, setSelectedButton] = useState(0);
const navigation = useNavigation();
const insets = useSafeAreaInsets();
```

```
const handleLogout = async () => {
  Alert.alert(
    'Termino de Sessão',
    'Tens a certeza que queres terminar?',
    [
      {
        text: 'Cancelar',
        style: 'cancel',
      },
      {
        text: 'Terminar Sessão',
        onPress: async () => {
          await signOut(auth);
        },
      },
    ],
    { cancelable: false }
  );
};
```

```
<Text style={styles.menuItem} onPress={handleLogout}>Log Out</Text>
```

```
const handleEmail = () => {
  const emailAddress = 'zecarebelo05@gmail.com';
  const emailUri = `mailto:${emailAddress}`;
```

```
Linking.openURL(emailUri)
```

José Carlos da Cunha Rebelo **51**

```
.then((supported) => {
  if (!supported) {
    console.error('Não foi possível abrir o aplicativo de e-mail.');
```

```
}
})
.catch((err) => console.error('Erro ao abrir o aplicativo de e-mail:', err));
};

const pickImage = async () => {
  const { status } = await ImagePicker.requestMediaLibraryPermissionsAsync();
  if (status !== 'granted') {
    alert('Sorry, access to your photo library is needed.');
```

```
return;
}
let result = await ImagePicker.launchImageLibraryAsync({
  mediaTypes: ImagePicker.MediaTypeOptions.All,
  allowsEditing: true,
  aspect: [4, 3],
  quality: 1
});
if (!result.cancelled) {
  setImage(result.uri);
}
};

const handleProfileImageClick = () => {
  // Lógica para abrir a imagem do perfil em tela cheia, permitir edição, etc.
  console.log('Usuário clicou na imagem do perfil');
```

```
// Aqui você pode adicionar a lógica para abrir a imagem do perfil em tela cheia, permitir edição, etc.
};

const uploadImage = async () => {
  setUploading(true);

  const response = await fetch(image);
  const blob = await response.blob();

  const auth = getAuth();
  const user = auth.currentUser;
  const uid = user.uid;

  const storage = getStorage(app);
  const filename = image.substring(image.lastIndexOf('/') + 1);
  const storageRef = ref(storage, `profile/${uid}/${filename}`);

  await uploadBytes(storageRef, blob);

  setUploading(false);
}
```

52 José Carlos da Cunha Rebelo

```
Alert.alert(
  'Photo uploaded!'
);
setImage(null);
};

return (
  <SafeAreaView style={styles.container}>
    <View style={{ width: responsiveScreenWidth(100),height: responsiveScreenHeight(17), alignItems:'center',
    justifyContent:'center', marginTop: responsiveScreenHeight(2)}}>

      {/*Imagem default*/}
      {/*<TouchableWithoutFeedback onPress={pickImage}>
        <Image source={ProfileP} style={{flex: 1,resizeMode:'contain',width: responsiveScreenWidth(24), height:
        responsiveScreenHeight(10), borderRadius: 999, marginBottom: responsiveScreenHeight(1), marginLeft:
        responsiveScreenWidth(28)}} />
        </TouchableWithoutFeedback>*/}
        <View style={{flexDirection: 'row', gap: responsiveScreenWidth(6), paddingLeft: responsiveScreenWidth(10)}}>
          <Profile />

          <View>
            <TouchableWithoutFeedback onPress={pickImage}>
              <Image source={require('../assets/images/camera.png')} style={{ width: responsiveScreenWidth(5), height:
              responsiveScreenHeight(5), resizeMode: 'contain' }} />
            </TouchableWithoutFeedback>

            {image && (
              <TouchableWithoutFeedback onPress={uploadImage}>
                <Image source={require('../assets/images/check.png')} style={{ width: responsiveScreenWidth(5), height:
                responsiveScreenHeight(5), resizeMode: 'contain' }} />
              </TouchableWithoutFeedback>
            )}
          </View>
          <View>
            <Text style={{color: 'white', fontSize: responsiveScreenFontSize(2.5), fontWeight:
            'bold'}}>{userEmail.split('@')[0].replace(/\.| /g, '')}</Text>
            <Text style={{color: 'white', fontSize: responsiveScreenFontSize(1.6), opacity:0.6}}>{userEmail}</Text>
          </View>

          <View style={{width: responsiveScreenWidth(80), height: responsiveScreenHeight(16), backgroundColor:'#181818',
          marginTop: responsiveScreenHeight(7), borderRadius:9,alignItems:'center',}}>
            <TouchableWithoutFeedback onPress={() => navigation.navigate('About')}>
              <View style={{width:responsiveScreenWidth(80), height: responsiveScreenHeight(5), flexDirection: 'row',
              borderRadius: 9, justifyContent:'space-between', alignItems:'center',paddingLeft: responsiveScreenWidth(5),
              paddingRight: responsiveScreenWidth(5)}}>
                <View style={{width: responsiveScreenWidth(6), height: responsiveScreenHeight(2.6), backgroundColor: '#B8E183',
                alignItems:'center', borderRadius:4, justifyContent:'center'}}>
                  <Image source={about} style={{width: responsiveScreenWidth(5),height: responsiveScreenHeight(5),
                  resizeMode:'contain'}}/>
                </View>
              </View>
            </TouchableWithoutFeedback>
          </View>
        </View>
      </SafeAreaView>
    </View>
  )
}
```

```

    <Text style={{color: 'white',fontSize: responsiveScreenFontSize(1.8), marginRight:
responsiveScreenWidth(25)}}>Acerca de ti </Text>
    <Image source={arrowR} style={{width: responsiveScreenWidth(5), height: responsiveScreenHeight(5),
resizeMode:'contain', opacity: 0.5}}/>
  </View>
</TouchableWithoutFeedback>

<View style={styles.Separator} />

<TouchableWithoutFeedback onPress={() => navigation.navigate('Contact')}>
  <View style={{width: responsiveScreenWidth(80), height: responsiveScreenHeight(5), marginTop:
responsiveScreenHeight(0.4), justifyContent:'space-between', alignItems: 'center', flexDirection: 'row',paddingLeft:
responsiveScreenWidth(5), paddingRight: responsiveScreenWidth(5)}}>
    <View style={{width: responsiveScreenWidth(6), height: responsiveScreenHeight(2.6), backgroundColor: '#45C7FF',
alignItems:'center', borderRadius:4, justifyContent:'center'}}>
      <Image source={mail} style={{width: responsiveScreenWidth(4.5),height: responsiveScreenHeight(4.5),
resizeMode:'contain'}}/>
    </View>
    <Text style={{color: 'white',fontSize: responsiveScreenFontSize(1.8), marginRight:
responsiveScreenWidth(32)}}>Contacto</Text>
    <Image source={arrowR} style={{width: responsiveScreenWidth(5), height: responsiveScreenHeight(5),
resizeMode:'contain', opacity: 0.5}}/>
  </View>
</TouchableWithoutFeedback>

<View style={styles.Separator} />

<TouchableWithoutFeedback onPress={handleEmail}>
  <View style={{width:responsiveScreenWidth(80), height: responsiveScreenHeight(5), flexDirection: 'row',
borderRadius: 9, justifyContent:'space-between', alignItems:'center',paddingLeft: responsiveScreenWidth(5),
paddingRight: responsiveScreenWidth(5)}}>
    <View style={{width: responsiveScreenWidth(6), height: responsiveScreenHeight(2.6), backgroundColor: 'tomato',
alignItems:'center', borderRadius:4, justifyContent:'center'}}>
      <Image source={flag} style={{width: responsiveScreenWidth(6),height: responsiveScreenHeight(6),
resizeMode:'contain'}}/>
    </View>
    <Text style={{color: 'white',fontSize: responsiveScreenFontSize(1.8), marginRight:
responsiveScreenWidth(25)}}>Reportar bug</Text>
    <Image source={arrowR} style={{width: responsiveScreenWidth(5), height: responsiveScreenHeight(5),
resizeMode:'contain', opacity: 0.5}}/>
  </View>
</TouchableWithoutFeedback>
</View>
<TouchableWithoutFeedback onPress={handleLogout}>
  <View style={{width: responsiveScreenWidth(80), height: responsiveScreenHeight(5), marginTop:
responsiveScreenHeight(0.4), backgroundColor:'#181818', borderRadius:9, justifyContent:'space-between', alignItems:
'center', flexDirection: 'row',paddingLeft: responsiveScreenWidth(5), paddingRight: responsiveScreenWidth(5),marginTop:
responsiveScreenHeight(5)}}>
    <View style={{width: responsiveScreenWidth(6), height: responsiveScreenHeight(2.6), backgroundColor: 'red',
alignItems:'center', borderRadius:4, justifyContent:'center'}}>
      <Image source={logout} style={{width: responsiveScreenWidth(4),height: responsiveScreenHeight(4),
resizeMode:'contain'}}/>

```

```

    </View>
    <Text style={{color: 'white',fontSize: responsiveScreenFontSize(1.8), marginRight:
responsiveScreenWidth(30)}}>Terminar sessão</Text>
  </View>
  </TouchableWithoutFeedback>
</SafeAreaView>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#090909',
    alignItems: 'center',
  },

  Seperator: {
    height: responsiveScreenHeight(0.1),
    width: responsiveScreenWidth(80),
    backgroundColor: '#212121',
  },
});

```

4.1.7 About/index.js (Página de Sobre)

```

import React, { useEffect, useState } from 'react';
import { View, Text, TextInput, TouchableOpacity, StyleSheet } from 'react-native';
import { FIRESTORE_DB, auth } from "../../config/firebase"; // Assuming you have auth imported from Firebase config
import { doc, getDoc, updateDoc, setDoc } from "firebase/firestore";
import { SafeAreaView } from 'react-native-safe-area-context';
import { responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from 'react-native-responsive-
dimensions';

export const About = () => {
  const [userData, setUserData] = useState({});
  const [newAge, setNewAge] = useState("");
  const [newWeight, setNewWeight] = useState("");
  const [newHeight, setNewHeight] = useState("");
  const [newSex, setNewSex] = useState("");

  useEffect(() => {
    const fetchUserData = async () => {
      try {
        const user = auth.currentUser;
        if (user) {
          const userId = user.uid;
          const userRef = doc(FIRESTORE_DB, 'users', userId);
          const docSnap = await getDoc(userRef);
          if (docSnap.exists()) {
            const data = docSnap.data();
            setUserData(data);
          }
        }
      } catch (error) {
        console.log(error);
      }
    };
    fetchUserData();
  });
}

```

```

    setNewAge(data.age || "");
    setNewWeight(data.weight || "");
    setNewHeight(data.height || "");
    setNewSex(data.sex || "");
  } else {
    await setDoc(userRef, { age: "", weight: "", height: "", sex: "" });
    setUserData({ age: "", weight: "", height: "", sex: "" });
  }
} else {
  console.error('Nenhum usuário autenticado.');
```

```

}
} catch (error) {
  console.error('Erro ao buscar dados do usuário:', error);
}
};
fetchUserData();
}, []);

const updateUserData = async () => {
  try {
    const user = auth.currentUser;
    if (user) {
      const userId = user.uid;
      const userRef = doc(FIRESTORE_DB, 'users', userId);
      const docSnap = await getDoc(userRef);
      if (docSnap.exists()) {
        await updateDoc(userRef, {
          age: newAge || userData.age || "",
          weight: newWeight || userData.weight || "",
          height: newHeight || userData.height || "",
          sex: newSex || userData.sex || "",
        });
        setUserData({
          age: newAge || userData.age || "",
          weight: newWeight || userData.weight || "",
          height: newHeight || userData.height || "",
          sex: newSex || userData.sex || "",
        });
      } else {
        console.error('Erro ao atualizar dados do utilizador: Documento não encontrado.');
```

```

      }
    } else {
      console.error('Nenhum utilizador autenticado.');
```

```

    }
  } catch (error) {
    console.error('Erro ao atualizar dados do utilizador:', error);
  }
};

return (
  <SafeAreaView style={styles.container}>
```

```
<View style={{width: '100%', height: responsiveScreenHeight(13), backgroundColor: 'black', alignItems: 'center',
justifyContent: 'center'}}>
  <Text style={{color: 'white', fontSize: responsiveScreenFontSize(3.5), fontWeight: 'bold'}}>Sobre ti</Text>
</View>
<View style={{width: '100%', height: responsiveScreenHeight(90), backgroundColor: '#111111', paddingTop:
responsiveScreenHeight(3), borderRadius: 20}}>
  <View style={styles.datacontainer}>
    <Text style={styles.emoji}>📅</Text>
    <Text style={styles.datatext}>Idade (y):</Text>
    <TextInput
      placeholder="Idade"
      onChangeText={setNewAge}
      value={newAge}
      color='lightgrey'
    />
  </View>
  <View style={styles.datacontainer}>
    <Text style={styles.emoji}>⚖️</Text>
    <Text style={styles.datatext}>Peso (kg):</Text>
    <TextInput
      placeholder="Peso"
      onChangeText={setNewWeight}
      value={newWeight}
      color='lightgrey'
    />
  </View>
  <View style={styles.datacontainer}>
    <Text style={styles.emoji}>📏</Text>
    <Text style={styles.datatext}>Altura (m):</Text>
    <TextInput
      placeholder="Altura"
      onChangeText={setNewHeight}
      value={newHeight}
      color='lightgrey'
      alignSelf='center'
    />
  </View>
  <View style={styles.datacontainer}>
    <Text style={styles.emoji}>👤</Text>
    <Text style={{color: 'white', fontWeight: 'bold', fontSize: responsiveScreenFontSize(2), marginRight:
responsiveScreenWidth(10)}}>Sexo:</Text>
    <View style={{ flexDirection: 'row' }}>
      <TouchableOpacity
        style={[styles.button, newSex === 'Masculino' && styles.selectedButton]}
        onPress={() => setNewSex('Masculino')}
      />
      <Text style={[styles.buttonText, newSex !== 'Masculino' && styles.whiteText]}>Masculino</Text>
    </TouchableOpacity>
    <TouchableOpacity
      style={[styles.button, newSex === 'Feminino' && styles.selectedButton]}
      onPress={() => setNewSex('Feminino')}
    />
  </View>
</View>
```

```

    >
    <Text style={{styles.buttonText, newSex !== 'Feminino' && styles.whiteText}}>Feminino</Text>
  </TouchableOpacity>
</View>
</View>
<TouchableOpacity style={styles.saveButton} onPress={updateUserData}>
  <Text style={styles.saveButtonText}>Guardar</Text>
</TouchableOpacity>
</View>
</SafeAreaView>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#090909',
  },
  button: {
    padding: 12,
    margin: 5,
    backgroundColor: '#212121',
    borderRadius: 20,
  },
  selectedButton: {
    backgroundColor: '#B8E183',
  },
  buttonText: {
    color: 'black',
  },
  whiteText: {
    color: 'white',
  },
  saveButton: {
    backgroundColor: '#B8E183',
    padding: 10,
    margin: 10,
    borderRadius: 10,
    alignItems: 'center',
    marginTop: responsiveScreenHeight(5)
  },
  saveButtonText: {
    color: 'black',
    fontWeight: 'bold',
  },
  datatext: {
    color: 'white',
    fontSize: responsiveScreenFontSize(2),
    fontWeight: 'bold',
    marginLeft: responsiveScreenWidth(-40)
  },
  datacontainer: {

```

```
width: responsiveScreenWidth(95),
height: responsiveScreenHeight(7),
backgroundColor: '#212121',
borderRadius: 10,
alignSelf: 'center',
flexDirection: 'row',
alignItems: 'center',
justifyContent: 'space-between',
paddingRight: responsiveScreenWidth(5),
margin: responsiveScreenWidth(2),
},
emoji: {
  marginLeft: responsiveScreenWidth(5),
  fontSize: responsiveScreenFontSize(2.5),
  marginRight: responsiveScreenWidth(2),
},
});
```

4.1.8 Contact/index.js (Página de Contactos)

```
import React, { useEffect, useState } from 'react';
import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from 'react-native';
import { FIRESTORE_DB, auth } from "../../config/firebase"; // Assuming you have auth imported from Firebase config
import { doc, getDoc, updateDoc, setDoc } from "firebase/firestore";
import { SafeAreaView } from 'react-native-safe-area-context';
import { responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from 'react-native-responsive-dimensions';
```

```
import instagram from '../assets/images/instalogo.png';
import github from '../assets/images/githublogo.png';
import linkedin from '../assets/images/linkedinlogo.png';
import x from '../assets/images/xlogo.png';
```

```
export const Contact = () => {

  return (
    <SafeAreaView style={styles.container}>
      <View style={{width: '100%', height: responsiveScreenHeight(13), backgroundColor: 'black', alignItems: 'center',
        justifyContent: 'center'}}>
        <Text style={{color: 'white', fontSize: responsiveScreenFontSize(3.5), fontWeight: 'bold'}}>Contacto</Text>
      </View>
      <View style={{width: '100%', height: responsiveScreenHeight(90), backgroundColor: '#111111', paddingTop:
        responsiveScreenHeight(3), borderRadius: 20}}>

        <View style={styles.datacontainer}>
          <Image source={instagram} style={{width: responsiveScreenWidth(7), height: responsiveScreenHeight(5),
            resizeMode: 'contain'}}/>
          <Text style={styles.datatext}>jose.rebelo1</Text>
        </View>

      </View>
    </SafeAreaView>
  );
}
```

José Carlos da Cunha Rebelo **59**

```

<View style={styles.datacontainer}>
  <Image source={x} style={{width: responsiveScreenWidth(7), height: responsiveScreenHeight(5),
resizeMode:'contain'}}/>
  <Text style={styles.datatext}>@j0sxns</Text>
</View>

<View style={styles.datacontainer}>
  <Image source={github} style={{width: responsiveScreenWidth(7), height: responsiveScreenHeight(5),
resizeMode:'contain'}}/>
  <Text style={styles.datatext}>JoseRebel0</Text>
</View>

<View style={styles.datacontainer}>
  <Image source={linkedin} style={{width: responsiveScreenWidth(7), height: responsiveScreenHeight(5),
resizeMode:'contain'}}/>
  <Text style={styles.datatext}>José Rebelo</Text>
</View>

</View>
</SafeAreaView>
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#090909',
  },
  button: {
    padding: 12,
    margin: 5,
    backgroundColor: '#212121',
    borderRadius: 20,
  },
  selectedButton: {
    backgroundColor: '#B8E183',
  },
  buttonText: {
    color: 'black',
  },
  whiteText: {
    color: 'white',
  },
  saveButton: {
    backgroundColor: '#B8E183',
    padding: 10,
    margin: 10,
    borderRadius: 10,
    alignItems: 'center',
    marginTop: responsiveScreenHeight(5)
  },
});

```

```

saveButtonText: {
  color: 'black',
  fontWeight: 'bold',
},
datatext:{
  color: 'white',
  fontSize: responsiveScreenFontSize(2),
  marginLeft: responsiveScreenWidth(-40)
},
datacontainer:{
  width: responsiveScreenWidth(95),
  height: responsiveScreenHeight(7),
  backgroundColor: '#212121',
  borderRadius: 10,
  alignSelf: 'center',
  flexDirection: 'row',
  alignItems:'center',
  justifyContent:'space-between',
  paddingRight: responsiveScreenWidth(10),
  padding: responsiveScreenWidth(3),
  margin: responsiveScreenWidth(2),
},
emoji: {
  marginLeft: responsiveScreenWidth(5),
  fontSize: responsiveScreenFontSize(2.5),
  marginRight: responsiveScreenWidth(2),
},
});

```

4.1.9 Upload/index.js (Página de Espaço aberto)

```

import React, {useState} from 'react'
import {View, StyleSheet, Image, Text, TouchableOpacity, SafeAreaView, Alert, Button} from 'react-native'
import * as ImagePicker from 'expo-image-picker'
import { firebase } from '.././././config/firebase';
import { getStorage, ref, app, uploadBytes } from 'firebase/storage';
import { useNavigation } from '@react-navigation/native';
import { getAuth } from 'firebase/auth';
import { Media } from '../././screens/Media';
import { responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from 'react-native-responsive-
dimensions';

```

```

export const Upload = () => {

  const navigation = useNavigation();

  const [image, setImage] = useState(null)
  const [uploading, setUploading] = useState(false)

```

```
const pickImage = async () => {
  const { status } = await ImagePicker.requestMediaLibraryPermissionsAsync();
  if (status !== 'granted') {
    alert('Sorry, access to your photo library is needed.');
```

return;

```
  }
  let result = await ImagePicker.launchImageLibraryAsync({
    mediaTypes: ImagePicker.MediaTypeOptions.All,
    allowsEditing: true,
    aspect: [4,3],
    quality: 1
  });
  const source = {uri: result.assets[0].uri}
  console.log(source)
  setImage(source)
};

const uploadImage = async () => {
  setUploading(true);

  const response = await fetch(image.uri);
  const blob = await response.blob();

  const auth = getAuth();
  const user = auth.currentUser;
  const uid = user.uid;

  const storage = getStorage(app);
  const filename = image.uri.substring(image.uri.lastIndexOf('/')+1);
  const storageRef = ref(storage, `images/${uid}/${filename}`);

  console.log("const storageRef");

  // try {
    await uploadBytes(storageRef, blob); // Now you should be able to upload
    // ... rest of your upload logic
  // } catch (e) {
    // console.log("Upload Error:", e);
  // }
  setUploading(false)
  Alert.alert(
    'Photo uploaded!'
  );
  setImage(null);
};
```

```

return (
  <SafeAreaView style={styles.container}>
    <Text
      style={{
        color: 'white',
        alignSelf: 'center',
        fontSize: responsiveScreenFontSize(3),
        fontWeight: 'bold',
        marginBottom: responsiveScreenHeight(5),
        marginTop: responsiveScreenHeight(3)
      }}
    />
    Espaço Aberto
  </Text>

  <View style={styles.spacebtns}>
    {image && <Image source={{ uri: image.uri }} style={styles.image} />}
    {image && (
      <TouchableOpacity style={styles.uploadButton} onPress={uploadImage}>
        <Text style={styles.btnText}>Upload Image</Text>
      </TouchableOpacity>
    )}
  </View>

  <Media />

  <View style={{ position: 'absolute', top: '85%', left: '45%' }}>
    <TouchableOpacity style={styles.floatingButton} onPress={pickImage}>
      <Text style={styles.pickbtn}>+</Text>
    </TouchableOpacity>
  </View>
</SafeAreaView>
);
};

export default Upload;

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: responsiveScreenHeight(5),
    backgroundColor: '#090909',
  },
  btnText: {
    width: responsiveScreenWidth(25),
    height: responsiveScreenHeight(4),
    backgroundColor: '#B8E183',
    borderRadius: 20,
    textAlign: 'center',
    padding: responsiveScreenHeight(0.5),
  },
});

```

```
fontWeight: 'bold',
},

spacebtns: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  width: '100%',
},

image: {
  width: '100%',
  aspectRatio: 1,
  resizeMode: 'cover',
},

floatingButton: {
  width: responsiveScreenWidth(12),
  height: responsiveScreenWidth(12),
  borderRadius: 999,
  backgroundColor: '#B8E183',
  justifyContent: 'center',
  alignItems: 'center',
  elevation: 5,
  aspectRatio: 1,
},

pickbtn: {
  fontSize: responsiveScreenFontSize(4),
},

deleteButton: {
  marginLeft: responsiveScreenWidth(2),
},

});
```

4.1.10 MenuAulimentacao/index.js (Página de Planos de alimentação)

```
import React from 'react';

import { StatusBar } from 'expo-status-bar';

import { SafeAreaProvider, useSafeAreaInsets, SafeAreaView } from 'react-native-safe-area-context';

import { StyleSheet, Image, View, Text, ScrollView, TouchableWithoutFeedback, } from 'react-native';

import { useNavigation } from '@react-navigation/native';

import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import seta from "../../assets/images/arrowLefticon.png";
```

```
import proteina from '../assets/images/protein2.png';

import semidratos from '../assets/images/nocarbs.png';

import cetogenica from '../assets/images/cetogenica.png';

import mediterranea from '../assets/images/mediterranea.png';

import vegan from '../assets/images/vegan.png';

export const MenuAlimentacao = () => {

  const navigation = useNavigation();

  return (

    <SafeAreaView style={styles.container}>

      <View style={{width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(28.5),marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent:'space-between', alignItems:'center', paddingLeft: responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3)}}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('Home')}>

          <Image source={seta} style={{resizeMode:'contain',width: responsiveScreenWidth(7), height: responsiveScreenHeight(5)}}/>

        </TouchableWithoutFeedback>

        <Text style={{color:'white', fontSize: responsiveScreenFontSize(2.5)}}>Escolha um plano:</Text>

      </View>

      <ScrollView vertical showsVerticalScrollIndicator style={{paddingLeft: responsiveScreenWidth(5), paddingRight: responsiveScreenWidth(5)}}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('Proteina')}>

          <View style={styles.option}>

            <Image source={proteina} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

            <View style={styles.overlay} />

            <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Dieta rica em proteínas</Text>

          </View>

        </TouchableWithoutFeedback>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('BaixoHidratos')}>

          <View style={styles.option}>

            <Image source={semidratos} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>
```

```

<View style={styles.overlay} />

<Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Dieta
baixa em hidratos de carbono</Text>

</View>

</TouchableWithoutFeedback>

<TouchableWithoutFeedback onPress={() => navigation.navigate('Cetogenica')}>

<View style={styles.option}>

<Image source={cetogenica} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

<View style={styles.overlay} />

<Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Dieta
cetogénica</Text>

</View>

</TouchableWithoutFeedback>

<TouchableWithoutFeedback onPress={() => navigation.navigate('Mediterranea')}>

<View style={styles.option}>

<Image source={mediterranea} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

<View style={styles.overlay} />

<Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Dieta
mediterrânea</Text>

</View>

</TouchableWithoutFeedback>

<TouchableWithoutFeedback onPress={() => navigation.navigate('Vegetariana')}>

<View style={styles.option}>

<Image source={vegan} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

<View style={styles.overlay} />

<Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Dieta
vegetariana</Text>

</View>

</TouchableWithoutFeedback>

</ScrollView>

<StatusBar style="light" />

</SafeAreaView>

```

```
);
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#0C0C0C',
  },

  overlay: {
    ...StyleSheet.absoluteFillObject,
    backgroundColor: 'rgba(0, 0, 0, 0.4)', // Ajuste a opacidade conforme necessário
    borderRadius: 22,
  },

  option: {
    width: responsiveScreenWidth(90),
    height: responsiveScreenHeight(15),
    alignContent: 'center',
    marginBottom: responsiveScreenHeight(3),
  },
});
```

4.1.10.1 *PlanosAlimentacao/baixohidratos.js (Página de Dieta baixa em hidratos)*

```
import React, { useState } from "react";
import { StatusBar } from "expo-status-bar";
import { SafeAreaView } from "react-native-safe-area-context";
import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';
import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';
import { useNavigation } from '@react-navigation/native';

import seta from "../../assets/images/arrowLefticon.png";
import esparbobora from "../../assets/images/esparbobora.png";
import frangobroc from "../../assets/images/frangobroc.png";
import wrap from "../../assets/images/wrap.png";
```

José Carlos da Cunha Rebelo **67**

```
import couveflor from '../assets/images/couveflor.png';

import camarao from '../assets/images/camarao.png';

const imagesData = [

  { source: frangobroc, description: 'Frango com brócolos e alho' },

  { source: esparbobora, description: 'Esparguete de abóbora com pesto' },

  { source: wrap, description: 'Wrap de alface e peru e guacamole' },

  { source: couveflor, description: 'Couve-flor assada com parmesão' },

  { source: camarao, description: 'Salada de camarão com abacate' },

];

export const BaixoHidratos = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };

  const handleCloseButton = () => {

    setSelectedImage(null);

  };

  return (

    <SafeAreaView style={styles.container}>

      <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(23.5),

        marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft:

        responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3)}}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>

          <View>

            <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height: responsiveScreenHeight(5) }}

          />


```

```

</View>

</TouchableWithoutFeedback>

<Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.3), fontWeight: 'bold' }}>Dieta baixa em hidratos de
carbono</Text>

</View>

<ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3), paddingRight:
responsiveScreenWidth(3) }}>

  <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

    {imagesData.map((image, index) => (

      <TouchableWithoutFeedback key={index} onPress={() => handleImageClick(image.description)}>

        <View style={styles.imageContainer}>

          <Image source={image.source} style={styles.image} />

          <View style={styles.overlay} />

          <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage === image.description ? 'black'
: 'white' }}>{image.description}</Text>

        </View>

      </TouchableWithoutFeedback>

    ))}

  </View>

  {selectedImage && (

    <View style={styles.overlayContainer}>

      <View style={styles.overlayDescription}>

        {renderContent(selectedImage)}

        <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor: '#B8E183', borderRadius: 10}}>

          <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>

        </TouchableOpacity>

      </View>

    </View>

  )}

</ScrollView>

</SafeAreaView>

```

```

    );
  };

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Frango com brócolos e alho':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Frango com brócolos e alho</Text>
          <ScrollView vertical showsVerticalScrollIndicator>

            <Text style={styles.textinho}>

              Frango com brócolos e alho é uma refeição saudável,

              destacando a proteína magra do frango,

              os benefícios nutritivos dos brócolos e as propriedades antioxidantes do alho.

              Uma combinação equilibrada que promove o crescimento muscular,

              fortalece o sistema imunológico e contribui para a saúde cardiovascular.

            </Text>

            <Text style={styles.titulo}>Ingredientes:</Text>

            <Text style={styles.textinho}>• Peitos de frango sem pele e desossados</Text>
            <Text style={styles.textinho}>• Brócolos frescos</Text>
            <Text style={styles.textinho}>• Dentes de alho picados</Text>
            <Text style={styles.textinho}>• Azeite</Text>
            <Text style={styles.textinho}>• Sal e pimenta a gosto</Text>
            <Text style={styles.textinho}>• Temperos opcionais: paprica, alecrim, sumo de limão</Text>

            <Text style={styles.titulo}>Instruções :</Text>

            <Text style={styles.textinho}>1. Temperar os peitos de frango com sal, pimenta e outros temperos desejados.</Text>
            <Text style={styles.textinho}>2. Numa frigideira ou assadeira, aquecer o azeite em fogo médio-alto.</Text>
          </ScrollView>
        </View>
      );
    }
  };

```

<Text style={styles.textinho}>3. Adicionar os peitos de frango à frigideira e cozinhar até dourarem e cozinham por completo. Reservar.</Text>

<Text style={styles.textinho}>4. Na mesma frigideira, adicionar mais azeite, se necessário, e refogar os brócolos até ficarem macios, mas ainda crocantes.</Text>

<Text style={styles.textinho}>5. Adicionar o alho picado aos brócolos e refogar por mais alguns minutos até que o alho esteja aromático.</Text>

<Text style={styles.textinho}>6. Devolver os peitos de frango à frigideira para aquecer.</Text>

<Text style={styles.textinho}>7. Se desejado, regar com sumo de limão antes de servir.</Text>

<Text style={styles.textinho}>8. Servir o frango sobre uma cama de brócolos e alho. Decorar com ervas aromáticas, se preferir.</Text>

</ScrollView>

</View>

);

case 'Esparguete de abóbora com pesto':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Esparguete de abóbora com pesto</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Esparguete de abóbora com pesto é uma opção leve e saudável,
rica em fibras e antioxidantes. O pesto, feito com manjerição,
azeite, pinhões e queijo parmesão, adiciona sabor.

Embora seja mais baixo em proteínas,
pode ser complementado com fontes adicionais,
como frango grelhado, para uma refeição equilibrada.

Uma escolha deliciosa e nutritiva.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Abóbora esparguete</Text>

José Carlos da Cunha Rebelo 71

<Text style={styles.textinho}>• Para o pesto: manjeriço fresco, azeite, pinhões, queijo parmesão, alho, sal e pimenta a gosto.</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Cortar a abóbora esparguete ao meio e retirar as sementes. Cozinhar no forno ou no micro-ondas até que a carne da abóbora esteja macia.</Text>

<Text style={styles.textinho}>2. Em um processador de alimentos, misturar manjeriço fresco, pinhões, alho, queijo parmesão e azeite até obter uma consistência desejada. Temperar com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>3. Usar um garfo para soltar e desfilar a carne da abóbora parecendo esparguete.</Text>

<Text style={styles.textinho}>4. Misturar o esparguete de abóbora com o pesto preparado.</Text>

<Text style={styles.textinho}>5. Servir imediatamente. Se desejado, adicionar mais queijo parmesão e pinhões por cima para decorar.</Text>

</ScrollView>

</View>

);

case 'Wrap de alface e peru e guacamole':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Wrap de alface e peru e guacamole</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Wrap de alface e peru com guacamole é uma refeição leve e saudável,

destacando a alface como substituto de massa,

o peru como fonte magra de proteína e o guacamole como fornecedor de ácidos graxos saudáveis.

Opção rica em proteínas, baixa em hidratos de carbono, favorecendo a saúde cardiovascular e proporcionando saciedade com nutrientes essenciais.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Folhas de alface grandes</Text>

```

<Text style={styles.textinho}>● Peito de peru em fatias</Text>

<Text style={styles.textinho}>● Guacamole (abacate, tomate, cebola, coentros, limão)</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Temperos opcionais: queijo desfiado, pimenta, alho em pó</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Lavar e secar cuidadosamente as folhas de alface, usando-as como "wraps" para envolver os
ingredientes.</Text>

<Text style={styles.textinho}>2. Espalhar uma porção de guacamole no centro de cada folha de alface.</Text>

<Text style={styles.textinho}>3. Adicionar fatias de peito de peru sobre o guacamole.</Text>

<Text style={styles.textinho}>4. Temperar com sal, pimenta e outros condimentos desejados.</Text>

<Text style={styles.textinho}>5. Adicionar (opcional) uma pitada de queijo desfiado para um toque adicional de sabor.</Text>

<Text style={styles.textinho}>6. Dobrar as laterais da folha de alface sobre o recheio e enrole, formando um wrap.</Text>

<Text style={styles.textinho}>7. Servir imediatamente.</Text>

</ScrollView>

</View>

);

case 'Couve-flor assada com parmesão':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Couve-flor assada com parmesão</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Couve-flor assada com parmesão é uma opção saudável e saborosa,

combinando a couve-flor nutritiva com o queijo parmesão.

Rica em vitaminas, fibras e proteínas,

é uma alternativa leve e benéfica para a saúde digestiva e óssea.

</Text>

```

```

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● 1 cabeça de couve-flor</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Queijo parmesão ralado</Text>

<Text style={styles.textinho}>● Temperos opcionais: alho em pó, ervas secas (alecrim, tomilho)</Text>


<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer o forno a 200° C.</Text>

<Text style={styles.textinho}>2. Cortar a couve-flor em pequenos floretes.</Text>

<Text style={styles.textinho}>3. Numa tigela, misturar os floretes de couve-flor com azeite, sal, pimenta e outros temperos
desejados.</Text>

<Text style={styles.textinho}>4. Espalhar os floretes numa assadeira em uma única camada.</Text>

<Text style={styles.textinho}>5. Assar no forno por 20-25 minutos, virando ocasionalmente, até que a couve-flor fique
dourada e macia.</Text>

<Text style={styles.textinho}>6. Retirar do forno e polvilhar generosamente com queijo parmesão ralado.</Text>

<Text style={styles.textinho}>7. Voltar ao forno por mais 5-7 minutos, ou até que o queijo esteja derretido e dourado.</Text>

<Text style={styles.textinho}>8. Servir imediatamente como acompanhamento ou petisco.</Text>


</ScrollView>

</View>

);

case 'Salada de camarão com abacate':

return (

  <View style={{alignContent:'center', alignItems:'center'}}>

    <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Salada de camarão com abacate</Text>

    <ScrollView vertical showsVerticalScrollIndicator>

      <Text style={styles.textinho}>

        Salada de camarão com abacate é uma opção refrescante e nutritiva,

        destacando a proteína magra do camarão e as gorduras saudáveis do abacate.

```

Rica em ácidos graxos omega-3, proteínas, fibras e nutrientes,
essa salada promove a saúde cardiovascular, suporta a construção muscular e oferece benefícios para a pele e o sistema imunológico.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Camarões cozidos e descascados</Text>

<Text style={styles.textinho}>● Abacate cortado aos cubos</Text>

<Text style={styles.textinho}>● Folhas de alface</Text>

<Text style={styles.textinho}>● Tomates cereja, cortados ao meio</Text>

<Text style={styles.textinho}>● Pepino em rodela</Text>

<Text style={styles.textinho}>● Coentro ou salsa</Text>

<Text style={styles.textinho}>● Sumo de limão</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Numa tigela grande, misturar os camarões cozidos, abacate, folhas de alface ou rúcula, tomates cereja e pepino.</Text>

<Text style={styles.textinho}>2. Adicionar o coentro ou salsa frescos picados para um toque de sabor.</Text>

<Text style={styles.textinho}>3. Temperar a salada com suco de limão, azeite, sal e pimenta a gosto. Misturar suavemente para garantir que todos os ingredientes estejam bem revestidos.</Text>

<Text style={styles.textinho}>4. Deixe a salada no frigorífico por alguns minutos para intensificar os sabores.</Text>

<Text style={styles.textinho}>5. Servir a salada de camarão com abacate como uma refeição leve e deliciosa.</Text>

</ScrollView>

</View>

);

default:

return null;

}

};

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    backgroundColor: 'black',  
  },  
  imageContainer: {  
    width: responsiveScreenWidth(45),  
    height: responsiveScreenHeight(20),  
    marginBottom: responsiveScreenHeight(2),  
    position: 'relative',  
  },  
  image: {  
    width: '100%',  
    height: responsiveScreenHeight(20),  
    borderRadius: 22,  
  },  
  overlay: {  
    ...StyleSheet.absoluteFillObject,  
    backgroundColor: 'rgba(0, 0, 0, 0.4)',  
    borderRadius: 22,  
  },  
  imageText: {  
    position: 'absolute',  
    bottom: 10,  
    left: 0,  
    right: 0,  
    textAlign: 'center',  
    fontWeight: 'bold',  
    fontSize: responsiveScreenFontSize(2.5),  
  },  
},
```

```

overlayContainer: {
  ...StyleSheet.absoluteFillObject,
  justifyContent: 'flex-end',
  alignItems: 'center',
},

overlayDescription: {
  backgroundColor: '#151515',
  padding: 20,
  borderRadius: 40,
  alignItems: 'center',
  width: responsiveScreenWidth(100),
  height: responsiveScreenHeight(60),
  paddingBottom: responsiveScreenHeight(7),
},

textinho:{
  color:'#E1E1E1',
  marginBottom: responsiveScreenHeight(0.5)
},

titulo:{
  color:'#B8E183',
  fontWeight:'bold',
  marginBottom: responsiveScreenHeight(1),
}
});

export default BaixoHidratos;
```

4.1.10.2 PlanosAlimentacao/cetogenica.js (Página de Dieta cetogénica)

```
import React, { useState } from "react";

import { StatusBar } from "expo-status-bar";

import { SafeAreaView } from "react-native-safe-area-context";

import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';

import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import { useNavigation } from '@react-navigation/native';

import seta from "../../assets/images/arrowLefticon.png";

import ovos from "../../assets/images/ovosmex.png";

import salmaom from "../../assets/images/salmaa-manteiga.png";

import pizzafran from "../../assets/images/pizza-frango.png";

import espinafre from "../../assets/images/espinafre.png";

import caprese from "../../assets/images/caprese.png";

const imagesData = [

  { source: salmaom, description: 'Salmão com Molho de Manteiga de Limão' },

  { source: ovos, description: 'Ovos Mexidos com Abacate e Bacon' },

  { source: pizzafran, description: 'Pizza de Frango com Crosta de Couve-Flor' },

  { source: espinafre, description: 'Espinafre Cremoso com Frango' },

  { source: caprese, description: 'Salada Caprese com Azeite' },

];

export const Cetogenica = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };

}
```

```
const handleCloseButton = () => {

  setSelectedImage(null);

};

return (

  <SafeAreaView style={styles.container}>

    <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(23.5),
marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft:
responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3) }}>

      <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>

        <View>

          <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height: responsiveScreenHeight(5) }}
/>

        </View>

      </TouchableWithoutFeedback>

      <Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.3), fontWeight: 'bold', marginRight: responsiveScreenWidth(7)
}}>Dieta cetogénica</Text>

    </View>

    <ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3), paddingRight:
responsiveScreenWidth(3) }}>

      <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

        {imagesData.map((image, index) => (

          <TouchableWithoutFeedback key={index} onPress={() => handleImageClick(image.description)}>

            <View style={styles.imageContainer}>

              <Image source={image.source} style={styles.image} />

              <View style={styles.overlay} />

              <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage === image.description ? 'black'
: 'white' }}>{image.description}</Text>

            </View>

          </TouchableWithoutFeedback>

        ))}

      </View>

    </ScrollView>

  )
);
```

```
{selectedImage && (
  <View style={styles.overlayContainer}>
    <View style={styles.overlayDescription}>
      {renderContent(selectedImage)}
      <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor: '#B8E183', borderRadius: 10}}>
        <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>
      </TouchableOpacity>
    </View>
  </View>
)}
</ScrollView>
</SafeAreaView>
);
};

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Salmão com Molho de Manteiga de Limão':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Salmão com molho de manteiga de limão</Text>
          <ScrollView vertical showsVerticalScrollIndicator>
            <Text style={styles.textinho}>
              Salmão com molho de manteiga de limão é uma refeição saborosa e saudável,
              destacando a proteína e os ácidos graxos ômega-3 do salmão.
              O molho cremoso de manteiga e limão adiciona um toque refrescante.
              Uma combinação nutritiva que promove a saúde cardiovascular e cerebral.
            </Text>
          <Text style={styles.titulo}>Ingredientes:</Text>
        </View>
      )
    }
  }
}
```

```

<Text style={styles.textinho}>● Postas de salmão (quantidade desejada)</Text>

<Text style={styles.textinho}>● Sumo de limão</Text>

<Text style={styles.textinho}>● Alho picado</Text>

<Text style={styles.textinho}>● Manteiga</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Ervas aromáticas (como a salsa)</Text>


<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer o forno a 200° C.</Text>

<Text style={styles.textinho}>2. Tempere as postas de salmão com sal, pimenta e alho picado a gosto.</Text>

<Text style={styles.textinho}>3. Numa frigideira, derreter a manteiga em fogo médio-alto.</Text>

<Text style={styles.textinho}>4. Colocar as postas de salmão na frigideira e dourar por alguns minutos de cada lado.</Text>

<Text style={styles.textinho}>5. Transferir as para uma assadeira e regar com sumo de limão.</Text>

<Text style={styles.textinho}>6. Assar no forno por cerca de 10-15 minutos, ou até que o salmão esteja cozido.</Text>

<Text style={styles.textinho}>7. Enquanto o salmão assa, adicionar mais manteiga à frigideira, se necessário, e sumo de limão.</Text>

<Text style={styles.textinho}>8. Cozinhar por mais alguns minutos até que a manteiga de limão esteja levemente caramelizada.</Text>

<Text style={styles.textinho}>9. Retirar o salmão do forno, passar o molho de manteiga de limão e decorar com ervas aromáticas, se desejado.</Text>

<Text style={styles.textinho}>10. Servir de imediato num prato.</Text>


</ScrollView>

</View>

);

case 'Ovos Mexidos com Abacate e Bacon':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Ovos mexidos com abacate e bacon</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

```

Ovos mexidos com abacate e bacon é uma refeição deliciosa e nutritiva,
destacando as proteínas dos ovos, as gorduras saudáveis do abacate e o sabor do bacon.
Uma combinação equilibrada que oferece benefícios para o desenvolvimento muscular,
saúde cardiovascular e saciedade.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Ovos</Text>

<Text style={styles.textinho}>● Abacate maduro</Text>

<Text style={styles.textinho}>● Bacon</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Temperos opcionais: cebolinha, salsa</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Cozinhar o bacon numa frigideira até que fique crocante. Retirar e reservar.</Text>

<Text style={styles.textinho}>2. Na mesma frigideira, remover parte da gordura do bacon, deixando uma quantidade para cozinhar os ovos.</Text>

<Text style={styles.textinho}>3. Bater os ovos numa tigela e despejar na frigideira.</Text>

<Text style={styles.textinho}>4. Mexer os ovos continuamente em fogo médio-baixo até que estejam cozidos, mas ainda cremosos.</Text>

<Text style={styles.textinho}>5. Temperar os ovos com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>6. Enquanto os ovos cozinham, cortar o abacate em fatias ou cubos.</Text>

<Text style={styles.textinho}>7. Servir os ovos mexidos em um prato, cobrir com abacate e despedaçar o bacon por cima.</Text>

<Text style={styles.textinho}>8. Adicionar temperos, como cebola ou salsa.</Text>

<Text style={styles.textinho}>9. Servir imediatamente.</Text>

</ScrollView>

</View>

);

case 'Pizza de Frango com Crosta de Couve-Flor':

return (

```
<View style={{alignContent:'center', alignItems:'center'}}>

  <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Pizza de frango com crosta de couve-flor</Text>

  <ScrollView vertical showsVerticalScrollIndicator>

    <Text style={styles.textinho}>

      Pizza de frango com crosta de couve-flor é uma alternativa saudável,

      destacando a proteína do frango e a crosta de couve-flor com baixo teor de carboidratos.

      Uma opção leve e nutritiva, contribuindo para a saciedade e a saúde muscular.

    </Text>

    { /*16 colheres de sopa */ }

    <Text style={styles.titulo}>Ingredientes:</Text>

    <Text style={styles.textinho}>• 1 cabeça de couve-flor</Text>

    <Text style={styles.textinho}>• 1 ovo</Text>

    <Text style={styles.textinho}>• 16 colheres de sopa de queijo ralado</Text>

    <Text style={styles.textinho}>• 1 colher de chá de temperos (orégano, manjeriçã, tomilho)</Text>

    <Text style={styles.textinho}>• Molho de tomate</Text>

    <Text style={styles.textinho}>• Frango cozido desfiado</Text>

    <Text style={styles.textinho}>• Ingredientes opcionais: tomate, cebola, pimento</Text>

    <Text style={styles.titulo}>Instruções :</Text>

    <Text style={styles.textinho}>1. Pré-aqueça o forno a 200° C.</Text>

    <Text style={styles.textinho}>2. Ralar a couve-flor até obter uma consistência semelhante à farinha.</Text>

    <Text style={styles.textinho}>3. Cozinhar a couve-flor ralada no micro-ondas por cerca de 5 minutos ou até ficar macia.
Deixar esfriar.</Text>

    <Text style={styles.textinho}>4. Misturar a couve-flor com o ovo, queijo ralado e com os temperos, formando uma
massa.</Text>

    <Text style={styles.textinho}>5. Espalhar a massa numa assadeira, formando a crosta da pizza, e levar ao forno por
aproximadamente 15-20 minutos, até dourar.</Text>

    <Text style={styles.textinho}>6. Retirar a crosta do forno e espalhar molho de tomate por cima.</Text>

    <Text style={styles.textinho}>7. Adicionar o frango desfiado e outros ingredientes opcionais.</Text>

    <Text style={styles.textinho}>8. Cubrir com queijo ralado e retornar ao forno até que o queijo derreta e a pizza fique
dourada.</Text>
```

```

<Text style={styles.textinho}>9. Retirar do forno, cortar em fatias e servir imediatamente.</Text>

</ScrollView>

</View>

);

case 'Espinafre Cremoso com Frango':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Espinafre cremoso com frango</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Espinafre cremoso com frango é uma refeição reconfortante e nutritiva,

realçando a proteína magra do frango e os benefícios do espinafre,

rico em ferro e vitaminas. Uma opção deliciosa que contribui para a saúde muscular,

imunológica e digestiva.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Peitos de frango cortados aos cubos</Text>

<Text style={styles.textinho}>• Espinafres frescos</Text>

<Text style={styles.textinho}>• Alho picado</Text>

<Text style={styles.textinho}>• Creme de leite ou iogurte grego</Text>

<Text style={styles.textinho}>• Queijo parmesão ralado</Text>

<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>• Azeite</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Temperar os cubos de frango com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>2. Aquecer o azeite numa frigideira em fogo médio-alto.</Text>

```

<Text style={styles.textinho}>3. Adicionar o frango à frigideira e cozinhar até dourar e estar completamente cozido.
Reservar.</Text>

<Text style={styles.textinho}>4. Na mesma frigideira, adicionar alho picado e refogar até ficar perfumado.</Text>

<Text style={styles.textinho}>5. Adicionar os espinafres na frigideira e cozinhar até murchar.</Text>

<Text style={styles.textinho}>6. Reduzir o fogo e adicionar o creme de leite ou iogurte grego na frigideira e misturar bem.</Text>

<Text style={styles.textinho}>7. Acrescentar o queijo parmesão ralado à mistura, mexer até obter um molho cremoso.</Text>

<Text style={styles.textinho}>8. Devolver o frango à frigideira, garantindo que fique bem misturado com o molho.</Text>

<Text style={styles.textinho}>9. Cozinhar por mais alguns minutos, ajustando os temperos conforme necessário.</Text>

<Text style={styles.textinho}>10. Servir quente, acompanhado de arroz, massa ou vegetais a gosto.</Text>

</ScrollView>

</View>

);

case 'Salada Caprese com Azeite':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Salada caprese com azeite</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

A salada Caprese com azeite é uma opção leve e refrescante,

combinando tomates, queijo mozzarella e manjerição,

regada com azeite de oliva. Rica em vitaminas,

proteínas e gorduras saudáveis,

proporciona uma refeição deliciosa e nutritiva.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Tomates maduros cortados em fatias</Text>

<Text style={styles.textinho}>• Queijo mozzarella fresco, em fatias</Text>

José Carlos da Cunha Rebelo 85

```
<Text style={styles.textinho}>• Folhas de manjeriço</Text>
<Text style={styles.textinho}>• Azeite extra virgem</Text>
<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>
<Text style={styles.textinho}>• Vinagre balsâmico (opcional)</Text>
```

```
<Text style={styles.titulo}>Instruções :</Text>
```

```
<Text style={styles.textinho}>1. Colocar as fatias de tomate e queijo mozzarella alternadamente em um prato ou travessa.</Text>
```

```
<Text style={styles.textinho}>2. Intercalar folhas de manjeriço entre as fatias de tomate e queijo.</Text>
```

```
<Text style={styles.textinho}>3. Regar a salada generosamente com azeite extra virgem.</Text>
```

```
<Text style={styles.textinho}>4. Temperar com sal e pimenta a gosto.</Text>
```

```
<Text style={styles.textinho}>5. Se desejado, adicionar um toque de vinagre balsâmico para intensificar os sabores.</Text>
```

```
<Text style={styles.textinho}>6. Servir imediatamente.</Text>
```

```
</ScrollView>
```

```
</View>
```

```
);
```

```
default:
```

```
return null;
```

```
}
```

```
};
```

```
const styles = StyleSheet.create({
```

```
  container: {
```

```
    flex: 1,
```

```
    backgroundColor: 'black',
```

```
  },
```

```
  imageContainer: {
```

```
    width: responsiveScreenWidth(45),
```

```
    height: responsiveScreenHeight(20),
```

```
    marginBottom: responsiveScreenHeight(2),
```

```
    position: 'relative',
```

```
  }
```

José Carlos da Cunha Rebelo

```
},  
  
image: {  
  width: '100%',  
  height: responsiveScreenHeight(20),  
  borderRadius: 22,  
},  
  
overlay: {  
  ...StyleSheet.absoluteFillObject,  
  backgroundColor: 'rgba(0, 0, 0, 0.4)',  
  borderRadius: 22,  
},  
  
imageText: {  
  position: 'absolute',  
  bottom: 10,  
  left: 0,  
  right: 0,  
  textAlign: 'center',  
  fontWeight: 'bold',  
  fontSize: responsiveScreenFontSize(2.5),  
},  
  
overlayContainer: {  
  ...StyleSheet.absoluteFillObject,  
  justifyContent: 'flex-end',  
  alignItems: 'center',  
},  
  
overlayDescription: {  
  backgroundColor: '#151515',  
  padding: 20,  
  borderRadius: 40,
```

```

    alignItems: 'center',

    width: responsiveScreenWidth(100),

    height: responsiveScreenHeight(60),

    paddingBottom: responsiveScreenHeight(7),

  },

  textinho:{

    color:'#E1E1E1',

    marginBottom: responsiveScreenHeight(0.5)

  },

  titulo:{

    color:'#B8E183',

    fontWeight:'bold',

    marginBottom: responsiveScreenHeight(1),

  }

});

export default Cetogenica;
```

4.1.10.3 *PlanosAlimentacao/mediterranea.js (Página de Dieta mediterrânea)*

```

import React, { useState } from "react";

import { StatusBar } from "expo-status-bar";

import { SafeAreaView } from "react-native-safe-area-context";

import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';

import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import { useNavigation } from '@react-navigation/native';

import seta from ".../assets/images/arrowLefticon.png";

import saladagrega from ".../assets/images/saladagrega.png";

import peixeassado from ".../assets/images/peixeassado.png";

import tabulee from ".../assets/images/tabulee.png";
```

88 José Carlos da Cunha Rebelo

```
import frutasiogurte from '../assets/images/frutasiogurte.png';

import wrapfrango from '../assets/images/wrapcomhummus.png';

const imagesData = [

  { source: saladagrega, description: 'Salada grega' },

  { source: peixeassado, description: 'Peixe assado com limão e ervas' },

  { source: tabule, description: 'Tabule' },

  { source: frutasiogurte, description: 'Frutas frescas com iogurte' },

  { source: wrapfrango, description: 'Wrap de frango com hummus' },

];

export const Mediterranea = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };

  const handleCloseButton = () => {

    setSelectedImage(null);

  };

  return (

    <SafeAreaView style={styles.container}>

      <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(23.5), marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft: responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3) }}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>

          <View>

            <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height: responsiveScreenHeight(5) }}

          />


```

```

</View>

</TouchableWithoutFeedback>

<Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.3), fontWeight: 'bold', marginRight: responsiveScreenWidth(7)
}}>Dieta mediterranea</Text>

</View>

<ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3), paddingRight:
responsiveScreenWidth(3) }}>

  <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

    {imagesData.map((image, index) => (

      <TouchableWithoutFeedback key={index} onPress={() => handleImageClick(image.description)}>

        <View style={styles.imageContainer}>

          <Image source={image.source} style={styles.image} />

          <View style={styles.overlay} />

          <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage === image.description ? 'black'
: 'white' }}>{image.description}</Text>

        </View>

      </TouchableWithoutFeedback>

    ))}

  </View>

  {selectedImage && (

    <View style={styles.overlayContainer}>

      <View style={styles.overlayDescription}>

        {renderContent(selectedImage)}

      <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor: '#B8E183', borderRadius: 10}}>

        <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>

      </TouchableOpacity>

    </View>

  </View>

)}

</ScrollView>

</SafeAreaView>

```

```

    );
};

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Salada grega':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Salada grega</Text>
          <ScrollView vertical showsVerticalScrollIndicator>

            <Text style={styles.textinho}>
              A salada grega é uma mistura saudável e vibrante,
              repleta de pepino, tomate, azeitonas, queijo feta,
              cebola roxa e temperada com azeite de oliva e orégano. Rica em vitaminas,
              antioxidantes e proteínas, é uma opção nutritiva que promove a saúde cardiovascular e oferece sabores frescos.
            </Text>

            <Text style={styles.titulo}>Ingredientes:</Text>
            <Text style={styles.textinho}>• Tomates maduros cortados em cubos</Text>
            <Text style={styles.textinho}>• Pepino cortado em rodela</Text>
            <Text style={styles.textinho}>• Cebola roxa fatiada finamente</Text>
            <Text style={styles.textinho}>• Azeitonas sem caroço</Text>
            <Text style={styles.textinho}>• Queijo feta em pedaços</Text>
            <Text style={styles.textinho}>• Orégãos</Text>
            <Text style={styles.textinho}>• Azeite</Text>
            <Text style={styles.textinho}>• Sumo de limão</Text>
            <Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

            <Text style={styles.titulo}>Instruções :</Text>

```

```

<Text style={styles.textinho}>1. Numa tigela grande, combinar os tomates, pepino, azeitonas, queijo feta e cebola
roxa.</Text>

<Text style={styles.textinho}>2. Polvilhar os orégãos sobre a mistura.</Text>

<Text style={styles.textinho}>3. Regar generosamente com azeite.</Text>

<Text style={styles.textinho}>4. Espremer um limão sobre a salada.</Text>

<Text style={styles.textinho}>5. Temperar com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>6. Delicadamente, misturar todos os ingredientes até que estejam bem combinados.</Text>

<Text style={styles.textinho}>7. Deixar a salada descansar no frigorífico por alguns minutos para intensificar os
sabores.</Text>

<Text style={styles.textinho}>8. Servir fria como acompanhamento ou como prato principal.</Text>

</ScrollView>

</View>

);

case 'Peixe assado com limão e ervas':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Peixe assado com limão e ervas</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Peixe assado com limão e ervas é uma opção saudável e saborosa,

destacando a proteína magra do peixe e os benefícios dos ácidos graxos omega-3.

Uma escolha nutritiva que promove a saúde cardiovascular e cognitiva.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Postas de peixe (como salmão, tilápia, linguado)</Text>

<Text style={styles.textinho}>• Limão</Text>

<Text style={styles.textinho}>• Ervas aromáticas(alecrim, tomilho, salsa)</Text>

<Text style={styles.textinho}>• Azeite extra virgem</Text>

```

```

<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>• Alho picado (opcional)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer o forno a 200° C.</Text>

<Text style={styles.textinho}>2. Temperar as postas de peixe com sal, pimenta, alho picado (opcional) e sumo de
limão.</Text>

<Text style={styles.textinho}>3. Colocar as postas numa assadeira forrada com papel alumínio ou papel manteiga.</Text>

<Text style={styles.textinho}>4. Regar as postas com azeite.</Text>

<Text style={styles.textinho}>5. Adicionar ervas aromáticas picadas, como alecrim, tomilho e salsa, sobre o peixe.</Text>

<Text style={styles.textinho}>6. Cortar fatias de limão e colocar sobre as postas.</Text>

<Text style={styles.textinho}>7. Assar no forno por cerca de 15-20 minutos, ou até que o peixe esteja cozido.</Text>

</ScrollView>

</View>

);

case 'Tabule':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Tabule</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Tabule é uma salada do Oriente Médio, leve e refrescante,

composta por trigo bulgur (opcional), tomate, pepino, hortelã,

salsa e cebola, temperada com azeite de oliva e suco de limão.

Rica em fibras, vitaminas e antioxidantes, é uma opção saudável e deliciosa.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• 200g de trigo bulgur (ou quinoa cozida)</Text>

```

<Text style={styles.textinho}>● 250g de tomate cortado em cubos pequenos</Text>

<Text style={styles.textinho}>● 150g de pepino descascado e cortado em cubos pequenos</Text>

<Text style={styles.textinho}>● 100g de cebola roxa finamente picada</Text>

<Text style={styles.textinho}>● 30g de hortelã picada</Text>

<Text style={styles.textinho}>● 30g de salsa picada</Text>

<Text style={styles.textinho}>● Sumo de limão</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Colocar o trigo bulgur numa tigela e cobrir com água a ferver. Deixar descansar por cerca de 20 minutos ou até que esteja macio. Escorrer o excesso de água.</Text>

<Text style={styles.textinho}>2. Numa tigela grande, misturar o trigo bulgur (ou quinoa cozida), tomate, pepino, cebola roxa, hortelã e salsa.</Text>

<Text style={styles.textinho}>3. Numa tigela pequena, fazer o molho, misturando o sumo do limão, azeite, sal e pimenta.</Text>

<Text style={styles.textinho}>4. Despejar o molho sobre os ingredientes da salada e misturar bem.</Text>

<Text style={styles.textinho}>5. Deixar o tabule descansar no frigorífico por pelo menos 30 minutos para que os sabores misturem.</Text>

<Text style={styles.textinho}>6. Servir frio como acompanhamento ou prato principal.</Text>

</ScrollView>

</View>

);

case 'Frutas frescas com iogurte':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Frutas frescas com iogurte</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Frutas frescas com iogurte oferecem uma opção saudável,

rica em proteínas, vitaminas e fibras. Beneficia a saúde muscular,

sistema imunológico e digestivo.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Frutas frescas à escolha(ex. morango, mirtilo, banana, kiwi)</Text>

<Text style={styles.textinho}>● Iogurte natural ou grego</Text>

<Text style={styles.textinho}>● Mel (opcional)</Text>

<Text style={styles.textinho}>● Granola (opcional)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Lavar e preparar as frutas, cortando-as em pedaços se necessário.</Text>

<Text style={styles.textinho}>2. Num recipiente, colocar uma porção de iogurte.</Text>

<Text style={styles.textinho}>3. Adicionar as frutas frescas por cima do iogurte.</Text>

<Text style={styles.textinho}>4. Regar com mel para um toque de doçura.</Text>

<Text style={styles.textinho}>5. Para mais textura, adicionar granola por cima.</Text>

<Text style={styles.textinho}>6. Misturar delicadamente para combinar os sabores.</Text>

<Text style={styles.textinho}>7. Servir imediatamente ou reservar no frigorífico para ficar com uma textura mais sólida.</Text>

</ScrollView>

</View>

);

case 'Wrap de frango com hummus':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Wrap de frango com hummus</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Wrap de frango com hummus combina a

proteína magra do frango com a cremosidade

José Carlos da Cunha Rebelo 95

e benefícios nutricionais do hummus. Uma opção equilibrada e deliciosa, rica em proteínas e fibras.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Tortilhas integrais ou outro tipo de wraps</Text>

<Text style={styles.textinho}>• Peitos de frango cozidos e cortados em tiras</Text>

<Text style={styles.textinho}>• Hummus caseiro ou comprado pronto</Text>

<Text style={styles.textinho}>• Vegetais (alface, tomate, pepino, cenoura)</Text>

<Text style={styles.textinho}>• Molho de iogurte (opcional)</Text>

<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Aquecer as tortilhas ou wraps de acordo com as instruções da embalagem.</Text>

<Text style={styles.textinho}>2. Espalhar uma generosa camada de hummus sobre cada tortilha.</Text>

<Text style={styles.textinho}>3. Colocar tiras de frango cozido sobre o hummus.</Text>

<Text style={styles.textinho}>4. Adicionar os vegetais.</Text>

<Text style={styles.textinho}>5. Se desejado, regar com molho de iogurte para um toque extra de sabor.</Text>

<Text style={styles.textinho}>6. Temperar com sal e pimenta.</Text>

<Text style={styles.textinho}>7. Dobrar as extremidades da tortilha e enrolar firmemente para formar o wrap.</Text>

<Text style={styles.textinho}>8. Cortar ao meio diagonalmente e servir.</Text>

</ScrollView>

</View>

);

default:

return null;

}

};

const styles = StyleSheet.create({

```
container: {  
  flex: 1,  
  backgroundColor: 'black',  
},  
imageContainer: {  
  width: responsiveScreenWidth(45),  
  height: responsiveScreenHeight(20),  
  marginBottom: responsiveScreenHeight(2),  
  position: 'relative',  
},  
image: {  
  width: '100%',  
  height: responsiveScreenHeight(20),  
  borderRadius: 22,  
},  
overlay: {  
  ...StyleSheet.absoluteFillObject,  
  backgroundColor: 'rgba(0, 0, 0, 0.4)',  
  borderRadius: 22,  
},  
imageText: {  
  position: 'absolute',  
  bottom: 10,  
  left: 0,  
  right: 0,  
  textAlign: 'center',  
  fontWeight: 'bold',  
  fontSize: responsiveScreenFontSize(2.5),  
},  
  
overlayContainer: {
```

```
...StyleSheet.absoluteFillObject,  
  
justifyContent: 'flex-end',  
  
alignItems: 'center',  
  
},  
  
overlayDescription: {  
  
  backgroundColor: '#151515',  
  
  padding: 20,  
  
  borderRadius: 40,  
  
  alignItems: 'center',  
  
  width: responsiveScreenWidth(100),  
  
  height: responsiveScreenHeight(60),  
  
  paddingBottom: responsiveScreenHeight(7),  
  
},  
  
textinho:{  
  
  color:'#E1E1E1',  
  
  marginBottom: responsiveScreenHeight(0.5)  
  
},  
  
titulo:{  
  
  color:'#B8E183',  
  
  fontWeight:'bold',  
  
  marginBottom: responsiveScreenHeight(1),  
  
  }  
  
});  
  
export default Mediterranea;
```

4.1.10.4 PlanosAlimentacao/proteina.js (Página de Dieta de proteina)

```
import React, { useState } from "react";

import { StatusBar } from "expo-status-bar";

import { SafeAreaView } from "react-native-safe-area-context";

import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';

import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import { useNavigation } from '@react-navigation/native';

import seta from "../../assets/images/arrowLefticon.png";

import saladaatum from "../../assets/images/saladaatum.png";

import peito from "../../assets/images/peito.png";

import omelete from "../../assets/images/omelete.png";

import iogurte from "../../assets/images/iogurte.png";

import salmao from "../../assets/images/salmao2.png";

const imagesData = [

  { source: peito, description: 'Peito de frango grelhado' },

  { source: omelete, description: 'Omelete de claras de ovo' },

  { source: saladaatum, description: 'Salada de atum com feijão branco' },

  { source: iogurte, description: 'Iogurte grego com frutas e nozes' },

  { source: salmao, description: 'Salmão assado com espargos' },

];

export const Proteina = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };

};
```

```
const handleCloseButton = () => {

  setSelectedImage(null);

};

return (

  <SafeAreaView style={styles.container}>

    <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(23.5),
marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft:
responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3) }}>

      <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>

        <View>

          <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height: responsiveScreenHeight(5) }}
/>

        </View>

      </TouchableWithoutFeedback>

      <Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.5), fontWeight: 'bold' }}>Dieta rica em proteínas</Text>

    </View>

    <ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3), paddingRight:
responsiveScreenWidth(3) }}>

      <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

        {imagesData.map((image, index) => (

          <TouchableWithoutFeedback key={index} onPress={() => handleImageClick(image.description)}>

            <View style={styles.imageContainer}>

              <Image source={image.source} style={styles.image} />

              <View style={styles.overlay} />

              <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage === image.description ? 'black'
: 'white' }}>{image.description}</Text>

            </View>

          </TouchableWithoutFeedback>

        ))}

      </View>

      {selectedImage && (
```

```

<View style={styles.overlayContainer}>
  <View style={styles.overlayDescription}>
    {renderContent(selectedImage)}
    <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor: '#B8E183', borderRadius: 10}}>
      <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>
    </TouchableOpacity>
  </View>
</View>
  </ScrollView>
</SafeAreaView>
);
};

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Peito de frango grelhado':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Peito de frango grelhado</Text>
          <ScrollView vertical showsVerticalScrollIndicator>
            <Text style={styles.textinho}>
              O peito de frango grelhado é uma excelente fonte de proteína magra,
              essencial para a construção muscular. Baixo em gordura saturada,
              é uma opção saudável para manter ou perder peso. Rico em nutrientes como vitaminas B,
              ferro e zinco, contribui para a saúde geral.
              Ao ser grelhado, mantém-se suculento sem adição excessiva de gorduras,
              tornando-o uma escolha nutritiva, saborosa e versátil.
            </Text>
          </ScrollView>
        </View>
      );
    default:
      return null;
  }
};

```

```

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Peitos de frango sem pele e desossados</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Temperos opcionais: alho em pó, cebola em pó, ervas aromáticas (como tomilho ou
alecrim)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer a grelha em fogo médio.</Text>

<Text style={styles.textinho}>2. Temperar os peitos de frango com sal, pimenta e outros temperos à escolha. Regar com um
pouco de azeite.</Text>

<Text style={styles.textinho}>3. Colocar os peitos de frango na grelha pré-aquecida. Cozinhar por 6-8 minutos de cada lado,
ou até que o frango atinja uma temperatura interna segura de 74 °C.</Text>

<Text style={styles.textinho}>4. Virar apenas uma vez para obter aquelas linhas de grelha atraentes. Evitar perfurar a carne,
pois isso pode resultar na perda de sucos.</Text>

<Text style={styles.textinho}>5. Retirar o frango da grelha e deixar descansar por alguns minutos antes de cortar. Isto ajuda
a garantir um frango mais suculento.</Text>

</ScrollView>

</View>

);

case 'Omelete de claras de ovo':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Omelete de claras de ovo</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

A omelete de claras de ovo é uma opção leve e saudável,

rica em proteínas de alta qualidade e baixa em gordura.

Versátil, pode ser preparada com vegetais ou queijo magro.

Contribui para uma dieta equilibrada,

```

sendo uma escolha popular de baixas calorias e alta em nutrientes.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Claras de ovos (quantidade desejada)</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Opções de ingredientes adicionais: vegetais (tomate, espinafre, cogumelos), queijo magro (opcional)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Em uma tigela, bater as claras dos ovos até ficarem bem misturadas. Adicionar uma pitada de sal e pimenta.</Text>

<Text style={styles.textinho}>2. Pré-aquecer uma frigideira antiaderente em fogo médio.</Text>

<Text style={styles.textinho}>3. Despejar as claras dos ovos batidas na frigideira pré-aquecida.</Text>

<Text style={styles.textinho}>4. Adicionar os ingredientes opcionais, sobre as claras.</Text>

<Text style={styles.textinho}>5. Cozinhar as claras até que as bordas estejam firmes. Com uma espátula, dobrar a omelete ao meio.</Text>

<Text style={styles.textinho}>6. Deixar a cozinhar até que o centro esteja completamente cozido.</Text>

<Text style={styles.textinho}>7. Colocar a omelete em um prato e servir imediatamente.</Text>

</ScrollView>

</View>

);

case 'Salada de atum com feijão branco':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Salada de atum com feijão branco</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

A salada de atum com feijão branco é uma opção nutritiva,

combinando proteínas magras do atum,

José Carlos da Cunha Rebelo 103

fibras do feijão branco e uma variedade de nutrientes de vegetais.

Baixa em calorias, é uma escolha leve e saudável,

ideal para refeições equilibradas e rápidas.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Atum em lata escorrido</Text>

<Text style={styles.textinho}>● Feijão branco cozido</Text>

<Text style={styles.textinho}>● Vegetais frescos (tomate, pepino, cebola)</Text>

<Text style={styles.textinho}>● Azeitonas</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sumo de limão</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Em uma tigela, misturar o atum escorrido com o feijão branco cozido.</Text>

<Text style={styles.textinho}>2. Adicionar os vegetais frescos picados, como tomate, pepino, cebola, e azeitonas.</Text>

<Text style={styles.textinho}>3. Temperar a salada com azeite, suco de limão, sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>4. Misturar bem todos os ingredientes para garantir uma distribuição uniforme dos sabores.</Text>

<Text style={styles.textinho}>5. Servir imediatamente ou refrigerar por algum tempo antes de servir para realçar os sabores.</Text>

</ScrollView>

</View>

);

case 'iogurte grego com frutas e nozes':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>iogurte grego com frutas e nozes</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

O iogurte grego com frutas e nozes é uma opção saudável e equilibrada,
combinando a cremosidade do iogurte grego com a doçura das frutas e a crocância das nozes.
Rico em proteínas do iogurte e nutrientes das frutas e nozes,
proporciona uma refeição nutritiva e saborosa.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Iogurte grego natural</Text>

<Text style={styles.textinho}>• Frutas frescas (à escolha)</Text>

<Text style={styles.textinho}>• Nozes (ou outros frutos secos)</Text>

<Text style={styles.textinho}>• Mel</Text>

<Text style={styles.textinho}>• Granola</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Em uma tigela, colocar uma porção de iogurte grego natural.</Text>

<Text style={styles.textinho}>2. Adicionar as frutas frescas cortadas em pedaços pequenos por cima do iogurte.</Text>

<Text style={styles.textinho}>3. Acrescente nozes picadas para adicionar uma textura crocante e benefícios nutricionais.</Text>

<Text style={styles.textinho}>4. Regar com um fio de mel para um toque de doçura extra.</Text>

<Text style={styles.textinho}>5. Adicionar granola por cima para complementar a textura e sabor.</Text>

<Text style={styles.textinho}>6. Misturar delicadamente os ingredientes na tigela.</Text>

<Text style={styles.textinho}>7. Servir imediatamente em um prato ou guardar no frigorífico para ficar com uma textura mais consistente.</Text>

</ScrollView>

</View>

);

case 'Salmão assado com espargos':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

José Carlos da Cunha Rebelo 105

```
<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Salmão assado com espargos</Text>
```

```
<ScrollView vertical showsVerticalScrollIndicator>
```

```
<Text style={styles.textinho}>
```

Salmão assado com espargos é uma refeição saudável,

combinando a suculência do salmão, rico em proteínas e ômega-3,

com a frescura dos espargos, ricos em fibras e vitaminas.

Uma opção deliciosa que promove benefícios para a saúde cardiovascular e imunológica.

```
</Text>
```

```
<Text style={styles.titulo}>Ingredientes:</Text>
```

```
<Text style={styles.textinho}>• Postas de salmão (quantidade desejada)</Text>
```

```
<Text style={styles.textinho}>• Espargos frescos</Text>
```

```
<Text style={styles.textinho}>• Azeite</Text>
```

```
<Text style={styles.textinho}>• Limão</Text>
```

```
<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>
```

```
<Text style={styles.textinho}>• Temperos opcionais (alho picado, salsa)</Text>
```

```
<Text style={styles.titulo}>Instruções :</Text>
```

```
<Text style={styles.textinho}>1. Pré-aquecer o forno a 200° C.</Text>
```

```
<Text style={styles.textinho}>2. Numa assadeira, colocar as postas de salmão e os espargos.</Text>
```

```
<Text style={styles.textinho}>3. Regar o salmão e os espargos com azeite e sumo de limão por cima.</Text>
```

```
<Text style={styles.textinho}>4. Temperar com sal, pimenta e outros temperos a gosto.</Text>
```

```
<Text style={styles.textinho}>5. Assar no forno por 15-20 minutos, ou até que o salmão e os espargos estejam no ponto.</Text>
```

```
<Text style={styles.textinho}>6. Retirar do forno e servir de imediato. Se necessário decorar com rodela de limão e ervas aromáticas.</Text>
```

```
</ScrollView>
```

```
</View>
```

```
);
```

```
default:
```

```
        return null;
    }
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'black',
  },
  imageContainer: {
    width: responsiveScreenWidth(45),
    height: responsiveScreenHeight(20),
    marginBottom: responsiveScreenHeight(2),
    position: 'relative',
  },
  image: {
    width: '100%',
    height: responsiveScreenHeight(20),
    borderRadius: 22,
  },
  overlay: {
    ...StyleSheet.absoluteFillObject,
    backgroundColor: 'rgba(0, 0, 0, 0.4)',
    borderRadius: 22,
  },
  imageText: {
    position: 'absolute',
    bottom: 10,
    left: 0,
    right: 0,
    textAlign: 'center',
  },
});
```

```
fontWeight: 'bold',

fontSize: responsiveScreenFontSize(2.5),

},

overlayContainer: {

  ...StyleSheet.absoluteFillObject,

  justifyContent: 'flex-end',

  alignItems: 'center',

},

overlayDescription: {

  backgroundColor: '#151515',

  padding: 20,

  borderRadius: 40,

  alignItems: 'center',

  width: responsiveScreenWidth(100),

  height: responsiveScreenHeight(60),

  paddingBottom: responsiveScreenHeight(7),

},

textinho:{

  color:'#E1E1E1',

  marginBottom: responsiveScreenHeight(0.5)

},

titulo:{

  color:'#B8E183',

  fontWeight:'bold',

  marginBottom: responsiveScreenHeight(1),

}

});
```

export default Proteina;

4.1.10.5 PlanosAlimentacao/vegetariana.js (Página de Dieta vegetariana)

```
import React, { useState } from "react";

import { StatusBar } from "expo-status-bar";

import { SafeAreaView } from "react-native-safe-area-context";

import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';

import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import { useNavigation } from '@react-navigation/native';

import seta from "../../assets/images/arrowLefticon.png";

import curry from "../../assets/images/curry2.png";

import quinoacomlegumes from "../../assets/images/salada.png";

import hamburguerfeijao from "../../assets/images/hamburguerfeijao.png";

import wraptofu from "../../assets/images/wraptofu.png";

import saladalentilha from "../../assets/images/saladalentilha.png";

const imagesData = [

  { source: curry, description: 'Curry de grão-de-bico' },

  { source: quinoacomlegumes, description: 'Tigela de quinoa com legumes assados' },

  { source: hamburguerfeijao, description: 'Hambúguer de feijão preto' },

  { source: wraptofu, description: 'Wrap de tofu com vegetais' },

  { source: saladalentilha, description: 'Salada de lentilhas com abacate' },

];
```

```
export const Vegetariana = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };

  const handleCloseButton = () => {

    setSelectedImage(null);

  };

  return (

    <SafeAreaView style={styles.container}>

      <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(23.5), marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft: responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3) }}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuAlimentacao')}>

          <View>

            <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height: responsiveScreenHeight(5) }} />

          </View>

        </TouchableWithoutFeedback>

        <Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.3), fontWeight: 'bold', marginRight: responsiveScreenWidth(7) }}>Dieta vegetariana</Text>

      </View>

    </SafeAreaView>

  );
}
```

```

<ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3),
paddingRight: responsiveScreenWidth(3) }}>

  <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

    {imagesData.map((image, index) => (

      <TouchableWithoutFeedback                                key={index}                                onPress={()                                ==>
handleImageClick(image.description)}>

        <View style={styles.imageContainer}>

          <Image source={image.source} style={styles.image} />

          <View style={styles.overlay} />

          <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage
=== image.description ? 'black' : 'white' }}>{image.description}</Text>

        </View>

      </TouchableWithoutFeedback>

    ))}

  </View>

  {selectedImage && (

    <View style={styles.overlayContainer}>

      <View style={styles.overlayDescription}>

        {renderContent(selectedImage)}

        <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor:
'#B8E183', borderRadius: 10}}>

          <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>

        </TouchableOpacity>

      </View>

    </View>

  )}

</ScrollView>

</SafeAreaView>

```

```

    );
  };

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Curry de grão-de-bico':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5),
marginBottom: responsiveScreenHeight(3)}}>Curry de grão-de-bico</Text>
          <ScrollView vertical showsVerticalScrollIndicator>

            <Text style={styles.textinho}>

              O curry de grão-de-bico é um prato nutritivo,

              rico em proteínas vegetais, fibras e uma variedade de especiarias.

              Oferece benefícios anti-inflamatórios e antioxidantes,

              sendo uma opção saudável e saborosa.

            </Text>

            <Text style={styles.titulo}>Ingredientes:</Text>

            <Text style={styles.textinho}>● 2 latas de grão-de-bico, escorrido e lavado</Text>

            <Text style={styles.textinho}>● 1 cebola média picada</Text>

            <Text style={styles.textinho}>● 2 dentes de alho picados</Text>

            <Text style={styles.textinho}>● 1 colher de sopa de gengibre ralado</Text>

            <Text style={styles.textinho}>● 1 lata de tomate sem pele picado</Text>

            <Text style={styles.textinho}>● 1 lata de leite de coco</Text>

            <Text style={styles.textinho}>● 2 colheres de sopa de curry em pó</Text>

```

<Text style={styles.textinho}>● 1 colher de chá de açafrão</Text>

<Text style={styles.textinho}>● 1 colher de chá de cominho em pó</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Coentros para decorar (opcional)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Numa panela grande, refogar a cebola até ficar macia.</Text>

<Text style={styles.textinho}>2. Adicionar alho e gengibre, refogando por mais 1-2 minutos.</Text>

<Text style={styles.textinho}>3. Adicionar as especiarias (curry, açafrão, cominho, coentros) e mexer bem para liberar os aromas.</Text>

<Text style={styles.textinho}>4. Acrescentar tomates sem pele e cozinhar por alguns minutos até que comecem a desmanchar.</Text>

<Text style={styles.textinho}>5. Adicionar grão-de-bico escorrido e leite de coco. Temperar com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>6. Deixar o curry ferver e, de seguida, reduzir o fogo, deixando a cozinhar 15-20 minutos para apurar os sabores.</Text>

<Text style={styles.textinho}>7. Ajustar os temperos conforme necessário.</Text>

<Text style={styles.textinho}>8. Servir o curry de grão-de-bico sobre arroz, quinoa ou com pão naan. Decorar com coentros, se desejado.</Text>

</ScrollView>

</View>

);

case 'Tigela de quinoa com legumes assados':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5),
marginBottom: responsiveScreenHeight(3)}}>Tigela de quinoa com legumes assados</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Tigela de quinoa com legumes assados é uma refeição equilibrada e nutritiva,
destacando a quinoa como fonte de proteínas e fibras,
enquanto os legumes assados adicionam vitaminas e antioxidantes.
Opção saudável para saciedade e energia sustentável.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● 1 chávena de quinoa lavada</Text>

<Text style={styles.textinho}>● Legumes variados (abóbora, cenoura, pimentão, brócolos),
cortados em pedaços</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● Temperos a gosto (como alho em pó, cúrcuma,
paprica)</Text>

<Text style={styles.textinho}>● Folhas de espinafre (opcional)</Text>

<Text style={styles.textinho}>● Molho de tahine (opcional)</Text>

<Text style={styles.textinho}>● Sementes de girassol ou abóbora torradas (opcional)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer o forno a 200°C.</Text>

<Text style={styles.textinho}>2. Cozinhar a quinoa conforme as instruções da
embalagem.</Text>

<Text style={styles.textinho}>3. Numa assadeira, distribuir os legumes cortados. Regar com azeite, temperar com sal, pimenta e outros temperos a gosto.</Text>

<Text style={styles.textinho}>4. Assar os legumes no forno pré-aquecido por cerca de 20-25 minutos ou até ficarem dourados e macios.</Text>

<Text style={styles.textinho}>5. Em cada tigela, colocar uma porção de quinoa cozida.</Text>

<Text style={styles.textinho}>6. Adicionar os legumes assados por cima da quinoa.</Text>

<Text style={styles.textinho}>7. Assar no forno por cerca de 15-20 minutos, ou até que o peixe esteja cozido.</Text>

<Text style={styles.textinho}>8. Finalizar com um fio de molho de tahine e sementes de girassol ou abóbora torradas.</Text>

<Text style={styles.textinho}>9. Misturar todos os ingredientes na tigela.</Text>

</ScrollView>

</View>

);

case 'Hambúrguer de feijão preto':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Hambúrguer de feijão preto</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Hambúrguer de feijão preto é uma alternativa vegetariana rica em proteínas e fibras, destacando os benefícios do feijão preto, como contribuições para a saúde cardiovascular e digestiva.

Saboroso e versátil, é uma escolha saudável para diversas preferências alimentares.

</Text>

José Carlos da Cunha Rebelo 115

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● 2 latas de feijão preto escorrido e lavado</Text>

<Text style={styles.textinho}>● 1 cebola média picada</Text>

<Text style={styles.textinho}>● 2 dentes de alho picados</Text>

<Text style={styles.textinho}>● 1 pimentão vermelho picado</Text>

<Text style={styles.textinho}>● 1 colher de chá de cominho em pó</Text>

<Text style={styles.textinho}>● 1 colher de chá de paprica</Text>

<Text style={styles.textinho}>● 1/2 colher de chá de malagueta em pó (opcional, para um toque de picância)</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>● 1/2 chávena de aveia em flocos</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Numa frigideira, refogar a cebola, alho e pimentão em um pouco de azeite até ficarem macios.</Text>

<Text style={styles.textinho}>2. Numa tigela grande, amassar o feijão preto com um garfo ou esmagar com as mãos (podendo deixar alguns grãos inteiros). </Text>

<Text style={styles.textinho}>3. Adicionar os vegetais refogados ao feijão preto amassado.</Text>

<Text style={styles.textinho}>4. Temperar com cominho, paprica, malagueta em pó, sal e pimenta. Misturar bem.</Text>

<Text style={styles.textinho}>5. Acrescentar a aveia em flocos e misturar até obter uma consistência que permita formar hambúrgueres.</Text>

<Text style={styles.textinho}>6. Dividir a mistura em porções e molde os hambúrgueres.</Text>

<Text style={styles.textinho}>7. Grelhar os hambúrgueres por cerca de 4-5 minutos de cada lado, até que fiquem dourados e cozidos por dentro.</Text>

<Text style={styles.textinho}>8. Servir os hambúrgueres de feijão preto em pães integrais com acompanhamentos de sua escolha.</Text>

</ScrollView>

</View>

);

case 'Wrap de tofu com vegetais':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5),
marginBottom: responsiveScreenHeight(3)}}>Wrap de tofu com vegetais</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Wrap de tofu com vegetais é uma opção vegetariana

rica em proteínas e nutrientes. O tofu fornece proteína

vegetal, enquanto os vegetais adicionam vitaminas,

minerais e fibras. Esta escolha saudável é versátil e deliciosa.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● 200g de tofu firme cortado em tiras</Text>

<Text style={styles.textinho}>● Abóbora, cenoura e pimentão cortados em tiras</Text>

<Text style={styles.textinho}>● Tortilhas ou wraps integrais</Text>

<Text style={styles.textinho}>● Sementes de gergelim</Text>

<Text style={styles.textinho}>● Molho de tahine (opcional)</Text>

<Text style={styles.textinho}>● Espinafres</Text>

José Carlos da Cunha Rebelo 117

<Text style={styles.textinho}>● Pimenta preta</Text>

<Text style={styles.textinho}>● Alho em pó</Text>

<Text style={styles.textinho}>● Molho de soja</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Numa frigideira, aquecer um pouco de azeite em fogo médio.</Text>

<Text style={styles.textinho}>2. Adicionar as tiras de tofu à frigideira e temperar com molho de soja, alho em pó e pimenta preta. Cozinhar até dourar levemente.</Text>

<Text style={styles.textinho}>3. Retirar o tofu da frigideira e reservar.</Text>

<Text style={styles.textinho}>4. Na mesma frigideira, adicionar mais azeite se necessário e refogar os vegetais (abóbora, cenoura e pimentão) até ficarem macios.</Text>

<Text style={styles.textinho}>5. Em cada tortilha, colocar uma porção de tofu cozido e uma porção dos vegetais refogados.</Text>

<Text style={styles.textinho}>6. Adicionar folhas de espinafre fresco.</Text>

<Text style={styles.textinho}>7. Se desejado, regar com molho de tahine e polvilhar com sementes de gergelim.</Text>

<Text style={styles.textinho}>8. Enrolar as tortilhas formando wraps.</Text>

<Text style={styles.textinho}>9. Servir imediatamente.</Text>

</ScrollView>

</View>

);

case 'Salada de lentilhas com abacate':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

```
<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5),
marginBottom: responsiveScreenHeight(3)}}>Salada de lentilhas com abacate</Text>
```

```
<ScrollView vertical showsVerticalScrollIndicator>
```

```
<Text style={styles.textinho}>
```

Salada de lentilhas com abacate combina a
proteína vegetal das lentilhas com as gorduras
saudáveis do abacate, oferecendo uma opção rica
em fibras e nutrientes essenciais. Uma escolha
deliciosa e nutritiva para uma refeição saudável.

```
</Text>
```

```
<Text style={styles.titulo}>Ingredientes:</Text>
```

```
<Text style={styles.textinho}>• 1 chávena de lentilhas cozidas</Text>
```

```
<Text style={styles.textinho}>• 1 abacate maduro, cortado em cubos</Text>
```

```
<Text style={styles.textinho}>• Tomate cereja cortado ao meio</Text>
```

```
<Text style={styles.textinho}>• Cebola roxa picada finamente</Text>
```

```
<Text style={styles.textinho}>• Coentros picados (opcional)</Text>
```

```
<Text style={styles.textinho}>• Sumo de limão</Text>
```

```
<Text style={styles.textinho}>• Azeite</Text>
```

```
<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>
```

```
<Text style={styles.titulo}>Instruções :</Text>
```

```
<Text style={styles.textinho}>1. Cozinhar as lentilhas conforme as instruções da embalagem e
reservar para esfriar.</Text>
```

```
<Text style={styles.textinho}>2. Numa tigela grande, misturar as lentilhas cozidas, abacate,
tomate cereja, cebola roxa e coentros.</Text>
```

José Carlos da Cunha Rebelo 119

<Text style={styles.textinho}>3. Temperar a salada com sumo de limão, azeite, sal e pimenta. Misturar bem.</Text>

<Text style={styles.textinho}>4. Deixar a salada marinar no frigorífico por pelo menos 30 minutos para intensificar os sabores.</Text>

<Text style={styles.textinho}>5. Servir a salada de lentilhas com abacate como acompanhamento ou como prato principal.</Text>

```

    </ScrollView>

  </View>

);

default:
  return null;
}
};

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'black',
  },
  imageContainer: {
    width: responsiveScreenWidth(45),
    height: responsiveScreenHeight(20),
    marginBottom: responsiveScreenHeight(2),
    position: 'relative',
  },
  image: {

```

```

width: '100%',

height: responsiveScreenHeight(20),

borderRadius: 22,

},

overlay: {

  ...StyleSheet.absoluteFillObject,

  backgroundColor: 'rgba(0, 0, 0, 0.4)',

  borderRadius: 22,

},

imageText: {

  position: 'absolute',

  bottom: 10,

  left: 0,

  right: 0,

  textAlign: 'center',

  fontWeight: 'bold',

  fontSize: responsiveScreenFontSize(2.5),

},

overlayContainer: {

  ...StyleSheet.absoluteFillObject,

  justifyContent: 'flex-end',

  alignItems: 'center',

},

overlayDescription: {

```

```
    backgroundColor: '#151515',  
    padding: 20,  
    borderRadius: 40,  
    alignItems: 'center',  
    width: responsiveScreenWidth(100),  
    height: responsiveScreenHeight(60),  
    paddingBottom: responsiveScreenHeight(7),  
  },  
  
  textinho:{  
    color:'#E1E1E1',  
    marginBottom: responsiveScreenHeight(0.5)  
  },  
  
  titulo:{  
    color:'#B8E183',  
    fontWeight:'bold',  
    marginBottom: responsiveScreenHeight(1),  
  }  
});  
  
export default Vegetariana;
```

4.1.11 PlanosTreino/inferiores.js (Página de treino de membros inferiores)

```
import React, { useState, useRef } from "react";

import { StatusBar } from "expo-status-bar";

import { SafeAreaView } from "react-native-safe-area-context";

import { StyleSheet, View, Text, TouchableOpacity, Image, ScrollView, TouchableWithoutFeedback } from 'react-native';

import { responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

import { useNavigation } from '@react-navigation/native';

import { Video } from 'expo-av';

import { Audio } from 'expo-av';


import seta from "../../assets/images/arrowLefticon.png";

import esparbobora from "../../assets/images/esparbobora.png";

import frangobroc from "../../assets/images/frangobroc.png";

import wrap from "../../assets/images/wrap.png";

import couveflor from "../../assets/images/couveflor.png";

import camarao from "../../assets/images/camarao.png";


const vidagachamentos = require('../../assets/videos/agachamentos.mp4');


const imagesData = [

  { source: frangobroc, description: 'Agachamentos' },

  { source: esparbobora, description: 'Prensa de pernas' },

  { source: wrap, description: 'Afundo com barra' },

  { source: couveflor, description: 'Elevação de panturrilha' },

  { source: camarao, description: 'Glute bridge' },

];
```

José Carlos da Cunha Rebelo 123

```
export const Inferiores = () => {

  const navigation = useNavigation();

  const [selectedImage, setSelectedImage] = useState(null);

  const videoRef = useRef(null);

  const [currentTime, setCurrentTime] = useState(0);

  const [duration, setDuration] = useState(0.1);

  const [paused, setPaused] = useState(false);

  const [isVideoPlaying, setIsVideoPlaying] = useState(false);

  const [overlay, setOverlay] = useState(false);


  const handlePlayPause = () => {

    setPaused(!paused);

    setIsVideoPlaying(!isVideoPlaying);

  };


  const handleImageClick = (imageDescription) => {

    setSelectedImage(imageDescription);

  };


  const handleCloseButton = () => {

    setSelectedImage(null);

  };


  return (

    <SafeAreaView style={styles.container}>

      <View style={{ width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight: responsiveScreenWidth(17), marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent: 'space-between', alignItems: 'center', paddingLeft: responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3) }}>

        <TouchableWithoutFeedback onPress={() => navigation.navigate('MenuTreinos')}>
```

```

<View>

  <Image source={seta} style={{ resizeMode: 'contain', width: responsiveScreenWidth(7), height:
responsiveScreenHeight(5) }} />

</View>

</TouchableWithoutFeedback>

<Text style={{ color: 'white', fontSize: responsiveScreenFontSize(2.3), fontWeight: 'bold' }}>Treino de membros
inferiores</Text>

</View>

<ScrollView vertical showsVerticalScrollIndicator style={{ paddingLeft: responsiveScreenWidth(3), paddingRight:
responsiveScreenWidth(3) }}>

  <View style={{ flexDirection: 'row', flexWrap: 'wrap', justifyContent: 'space-between' }}>

    {imagesData.map((image, index) => (

      <TouchableWithoutFeedback key={index} onPress={() => handleImageClick(image.description)}>

        <View style={styles.imageContainer}>

          <Image source={image.source} style={styles.image} />

          <View style={styles.overlay} />

          <Text style={{ ...styles.imageText, backgroundColor: 'transparent', color: selectedImage ===
image.description ? 'black' : 'white' }}>{image.description}</Text>

        </View>

      </TouchableWithoutFeedback>

    ))}

  </View>

  {selectedImage && (

    <View style={styles.overlayContainer}>

      <TouchableOpacity onPress={handlePlayPause} style={styles.playPauseButton}>

        <Text style={styles.playPauseButtonText}>

          {isVideoPlaying ? 'Pause' : 'Play'}

        </Text>

      </TouchableOpacity>

```

```

<View style={styles.overlayDescription}>
  {renderContent(selectedImage)}
  <TouchableOpacity onPress={handleCloseButton} style={{padding: 10, backgroundColor: '#B8E183',
borderRadius: 10}}>
    <Text style={{ color: 'black', textAlign: 'center' }}>Fechar Descrição</Text>
  </TouchableOpacity>
</View>
</View>
)}
</ScrollView>
</SafeAreaView>
);
};

```

```

const renderContent = (selectedImage) => {
  switch (selectedImage) {
    case 'Agachamentos':
      return (
        <View style={{alignContent:'center', alignItems:'center'}}>
          <Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Agachamentos</Text>
          <ScrollView vertical showsVerticalScrollIndicator>
            <Text style={styles.textinho}>
              Agachamentos são exercícios fundamentais que
              fortalecem músculos das pernas, glúteos e costas,
              melhoram postura, aumentam mobilidade, estimulam o
              sistema cardiovascular, queimam calorias, têm aplicação
              prática e estimulam a produção de hormônios anabólicos.
            </Text>
          </ScrollView>
        </View>
      )
    default:
      return null;
  }
};

```

```

</Text>

<Text style={styles.titulo}>Como fazer:</Text>

{/**/}

</ScrollView>

</View>

);

case 'Prensa de pernas':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Esparguete de abóbora com pesto</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Esparguete de abóbora com pesto é uma opção leve e saudável,
rica em fibras e antioxidantes. O pesto, feito com manjerição,
azeite, pinhões e queijo parmesão, adiciona sabor.

Embora seja mais baixo em proteínas,
pode ser complementado com fontes adicionais,
como frango grelhado, para uma refeição equilibrada.

Uma escolha deliciosa e nutritiva.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Abóbora esparguete</Text>

```

<Text style={styles.textinho}>• Para o pesto: manjeriço fresco, azeite, pinhões, queijo parmesão, alho, sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Cortar a abóbora esparguete ao meio e retirar as sementes. Cozinhar no forno ou no micro-ondas até que a carne da abóbora esteja macia.</Text>

<Text style={styles.textinho}>2. Em um processador de alimentos, misturar manjeriço fresco, pinhões, alho, queijo parmesão e azeite até obter uma consistência desejada. Temperar com sal e pimenta a gosto.</Text>

<Text style={styles.textinho}>3. Usar um garfo para soltar e desfiar a carne da abóbora parecendo esparguete.</Text>

<Text style={styles.textinho}>4. Misturar o esparguete de abóbora com o pesto preparado.</Text>

<Text style={styles.textinho}>5. Servir imediatamente. Se desejado, adicionar mais queijo parmesão e pinhões por cima para decorar.</Text>

</ScrollView>

</View>

);

case 'Afundo com barra':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom: responsiveScreenHeight(3)}}>Wrap de alface e peru e guacamole</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Wrap de alface e peru com guacamole é uma refeição leve e saudável,

destacando a alface como substituto de massa,

o peru como fonte magra de proteína e o guacamole como fornecedor de ácidos graxos saudáveis.

Opção rica em proteínas, baixa em hidratos de carbono, favorecendo a saúde cardiovascular e proporcionando saciedade com nutrientes essenciais.

</Text>

```

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• Folhas de alface grandes</Text>

<Text style={styles.textinho}>• Peito de peru em fatias</Text>

<Text style={styles.textinho}>• Guacamole (abacate, tomate, cebola, coentros, limão)</Text>

<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>• Temperos opcionais: queijo desfiado, pimenta, alho em pó</Text>


<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Lavar e secar cuidadosamente as folhas de alface, usando-as como "wraps"
para envolver os ingredientes.</Text>

<Text style={styles.textinho}>2. Espalhar uma porção de guacamole no centro de cada folha de
alface.</Text>

<Text style={styles.textinho}>3. Adicionar fatias de peito de peru sobre o guacamole.</Text>

<Text style={styles.textinho}>4. Temperar com sal, pimenta e outros condimentos desejados.</Text>

<Text style={styles.textinho}>5. Adicionar (opcional) uma pitada de queijo desfiado para um toque adicional
de sabor.</Text>

<Text style={styles.textinho}>6. Dobrar as laterais da folha de alface sobre o recheio e enrolar, formando um
wrap.</Text>

<Text style={styles.textinho}>7. Servir imediatamente.</Text>


</ScrollView>

</View>

);

case 'Elevação de panturrilha':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Couve-flor assada com parmesão</Text>

<ScrollView vertical showsVerticalScrollIndicator>

```

<Text style={styles.textinho}>

Couve-flor assada com parmesão é uma opção saudável e saborosa,
combinando a couve-flor nutritiva com o queijo parmesão.

Rica em vitaminas, fibras e proteínas,

é uma alternativa leve e benéfica para a saúde digestiva e óssea.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>• 1 cabeça de couve-flor</Text>

<Text style={styles.textinho}>• Azeite</Text>

<Text style={styles.textinho}>• Sal e pimenta a gosto</Text>

<Text style={styles.textinho}>• Queijo parmesão ralado</Text>

<Text style={styles.textinho}>• Temperos opcionais: alho em pó, ervas secas (alecrim, tomilho)</Text>

<Text style={styles.titulo}>Instruções :</Text>

<Text style={styles.textinho}>1. Pré-aquecer o forno a 200° C.</Text>

<Text style={styles.textinho}>2. Cortar a couve-flor em pequenos floretes.</Text>

<Text style={styles.textinho}>3. Numa tigela, misturar os floretes de couve-flor com azeite, sal, pimenta e outros temperos desejados.</Text>

<Text style={styles.textinho}>4. Espalhar os floretes numa assadeira em uma única camada.</Text>

<Text style={styles.textinho}>5. Assar no forno por 20-25 minutos, virando ocasionalmente, até que a couve-flor fique dourada e macia.</Text>

<Text style={styles.textinho}>6. Retirar do forno e polvilhar generosamente com queijo parmesão ralado.</Text>

<Text style={styles.textinho}>7. Voltar ao forno por mais 5-7 minutos, ou até que o queijo esteja derretido e dourado.</Text>

<Text style={styles.textinho}>8. Servir imediatamente como acompanhamento ou petisco.</Text>

</ScrollView>

```

</View>

);

case 'Glute bridge':

return (

<View style={{alignContent:'center', alignItems:'center'}}>

<Text style={{ color: '#B8E183', fontWeight:'bold', fontSize: responsiveScreenFontSize(2.5), marginBottom:
responsiveScreenHeight(3)}}>Salada de camarão com abacate</Text>

<ScrollView vertical showsVerticalScrollIndicator>

<Text style={styles.textinho}>

Salada de camarão com abacate é uma opção refrescante e nutritiva,

destacando a proteína magra do camarão e as gorduras saudáveis do abacate.

Rica em ácidos graxos omega-3, proteínas, fibras e nutrientes,

essa salada promove a saúde cardiovascular, suporta a construção muscular e oferece benefícios para a
pele e o sistema imunológico.

</Text>

<Text style={styles.titulo}>Ingredientes:</Text>

<Text style={styles.textinho}>● Camarões cozidos e descascados</Text>

<Text style={styles.textinho}>● Abacate cortado aos cubos</Text>

<Text style={styles.textinho}>● Folhas de alface</Text>

<Text style={styles.textinho}>● Tomates cereja, cortados ao meio</Text>

<Text style={styles.textinho}>● Pepino em rodela</Text>

<Text style={styles.textinho}>● Coentro ou salsa</Text>

<Text style={styles.textinho}>● Sumo de limão</Text>

<Text style={styles.textinho}>● Azeite</Text>

<Text style={styles.textinho}>● Sal e pimenta a gosto</Text>

<Text style={styles.titulo}>Instruções :</Text>

```

<Text style={styles.textinho}>1. Numa tigela grande, misturar os camarões cozidos, abacate, folhas de alface ou rúcula, tomates cereja e pepino.</Text>

<Text style={styles.textinho}>2. Adicionar o coentro ou salsa frescos picados para um toque de sabor.</Text>

<Text style={styles.textinho}>3. Temperar a salada com suco de limão, azeite, sal e pimenta a gosto. Misturar suavemente para garantir que todos os ingredientes estejam bem revestidos.</Text>

<Text style={styles.textinho}>4. Deixe a salada no frigorífico por alguns minutos para intensificar os sabores.</Text>

<Text style={styles.textinho}>5. Servir a salada de camarão com abacate como uma refeição leve e deliciosa.</Text>

</ScrollView>

</View>

);

default:

return null;

}

};

const styles = StyleSheet.create({

container: {

flex: 1,

backgroundColor: 'black',

},

imageContainer: {

width: responsiveScreenWidth(45),

height: responsiveScreenHeight(20),

marginBottom: responsiveScreenHeight(2),

position: 'relative',

},

image: {

```
width: '100%',  
  
height: responsiveScreenHeight(20),  
  
borderRadius: 22,  
  
},  
  
overlay: {  
  
  ...StyleSheet.absoluteFillObject,  
  
  backgroundColor: 'rgba(0, 0, 0, 0.4)',  
  
  borderRadius: 22,  
  
},  
  
imageText: {  
  
  position: 'absolute',  
  
  bottom: 10,  
  
  left: 0,  
  
  right: 0,  
  
  textAlign: 'center',  
  
  fontWeight: 'bold',  
  
  fontSize: responsiveScreenFontSize(2.5),  
  
},  
  
  
  
overlayContainer: {  
  
  ...StyleSheet.absoluteFillObject,  
  
  justifyContent: 'flex-end',  
  
  alignItems: 'center',  
  
},  
  
  
  
overlayDescription: {  
  
  backgroundColor: '#151515',  
  
  padding: 20,  
  
  borderRadius: 40,
```

```
alignItems: 'center',  
  
width: responsiveScreenWidth(100),  
  
height: responsiveScreenHeight(60),  
  
paddingBottom: responsiveScreenHeight(7),  
  
},
```

```
textinho:{  
  
  color:'#E1E1E1',  
  
  marginBottom: responsiveScreenHeight(2)  
  
},
```

```
titulo:{  
  
  color:'#B8E183',  
  
  fontWeight:'bold',  
  
  marginBottom: responsiveScreenHeight(1),  
  
},
```

```
playPauseButton: {  
  
  position: 'absolute',  
  
  top: '0%',  
  
  left: '50%',  
  
  transform: [{ translateX: -25 }, { translateY: -25 }], // Ajuste para centralizar o botão  
  
  backgroundColor: 'rgba(255, 255, 255, 0.8)',  
  
  width: 50,  
  
  height: 50,  
  
  borderRadius: 25,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},
```

134 José Carlos da Cunha Rebelo

```
playPauseButtonText: {  
  fontSize: 18,  
  fontWeight: 'bold',  
  color: 'black',  
},  
});
```

```
export default Inferiores;
```

4.1.13 Media/index.js (Página de funcionamento de imagens)

```
import React, { useState, useEffect } from 'react';  
  
import { View, ScrollView, Image, StyleSheet, Text, TouchableOpacity, Alert } from 'react-native';  
  
import { firebase } from '../../config/firebase';  
  
import { getStorage, ref, app, getDownloadURL, listAll, getMetadata, deleteObject } from 'firebase/storage';  
  
import { getAuth } from 'firebase/auth';  
  
import { getFirestore, collection, addDoc, orderBy, query } from 'firebase/firestore';  
  
import { responsiveHeight, responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth, responsiveWidth }  
from 'react-native-responsive-dimensions';  
  
  
import trash from '../../assets/images/trashicon.png';  
  
export const Media = ({ uid }) => {  
  const [imageUrls, setImageUrls] = useState([]);  
  
  const storage = getStorage(app); // Initialize storage at the top level  
  
  
  const fetchImages = async () => {  
    const auth = getAuth();  
  
    const user = auth.currentUser;  
  
    const uid = user.uid;
```

```
const imagesRef = ref(storage, `images/${uid}`);

const listResult = await listAll(imagesRef);

const urlsPromises = listResult.items.map(async imageRef => {

  const url = await getDownloadURL(imageRef);

  const metadata = await getMetadata(imageRef);

  return { url, createdAt: metadata.timeCreated, ref: imageRef }; // Adicionamos a referência do arquivo
});

const urls = await Promise.all(urlsPromises);

// Ordenar as URLs pela data/hora de criação
urls.sort((a, b) => b.createdAt - a.createdAt);

setImageUrls(urls);
};

useEffect(() => {

  fetchImages();

  // Atualiza a lista de imagens a cada 60 segundos
  const intervalId = setInterval(fetchImages, 10000);

  console.log("Atualizando lista de imagens...");

  // Limpa o intervalo quando o componente é desmontado
  return () => clearInterval(intervalId);

}, [storage, uid]); // Adiciona storage e uid como dependências
```

```
const confirmDelete = (index) => {

  Alert.alert(

    'Confirmação',

    'Tem certeza que deseja apagar a foto?',

    [

      { text: 'Cancelar', style: 'cancel' },

      { text: 'Apagar', onPress: () => deletelImage(index) }

    ],

    { cancelable: true }

  );

};

const deletelImage = async (index) => {

  const { ref } = imageUrls[index];

  await deleteObject(ref); // Deletamos o arquivo do storage

  const updatedUrls = [...imageUrls];

  updatedUrls.splice(index, 1);

  setImageUrls(updatedUrls);

};

return (

  <View style={styles.container}>

    <ScrollView>

      {imageUrls.map((image, index) => (

        <View key={index} style={styles.imageContainer}>

          <Image source={{ uri: image.url }} style={styles.image} />

          <TouchableOpacity style={styles.deleteButton} onPress={() => confirmDelete(index)}>

            <Image source={trash} style={styles.trashIcon} />

          </TouchableOpacity>

        )

      )}

    </ScrollView>

  )

);
```

```

        </View>

      )}

    </ScrollView>

  </View>

);

};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 1,
    backgroundColor: '#191919',
    borderRadius: 20,
    paddingTop: responsiveScreenHeight(3),
    alignItems: 'center',
  },
  imageContainer: {
    marginBottom: responsiveScreenHeight(4),
    alignItems: 'center',
    position: 'relative',
  },
  image: {
    width: responsiveScreenWidth(90),
    height: responsiveScreenHeight(25),
    borderRadius: 20,
  },
  deleteButton: {
    position: 'absolute',
    top: responsiveScreenHeight(1),

```

```

right: 10,

backgroundColor: '#B8E183',

alignItems: 'center',

justifyContent: 'center',

height: responsiveScreenHeight(4.5),

width: responsiveScreenWidth(10),

borderRadius: 9999,

},

trashIcon: {

width: responsiveScreenWidth(4),

height: responsiveScreenWidth(4),

resizeMode: 'contain',

},

deleteButtonText: {

color: 'white',

fontWeight: 'bold',

},

});

```

4.1.14 BottomTabNavigation/index.js (Barra de navegação inferior)

```

import * as React from 'react';

import { Home } from "../Home";

import { Settings } from "../Settings";

import { Scan } from '../Scan';

import Upload from '../Upload';

import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth,
responsiveScreenHeight, responsiveScreenFontSize } from 'react-native-responsive-dimensions';

```

José Carlos da Cunha Rebelo 139

```
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";
import { NavigationContainer } from '@react-navigation/native';
import { View, Image } from 'react-native';

import Scanbutt from '../assets/images/scanicon1.png';
import Scanbuttpre from '../assets/images/pressedscan1.png';
import Settingsbutt from '../assets/images/settingsicon1.png';
import Settingsbuttpre from '../assets/images/pressedsettings1.png';
import Homebutt from '../assets/images/homeiconnotgreen.png';
import Homebuttpre from '../assets/images/homeicon1.png';
import post from '../assets/images/post.png';
import postpre from '../assets/images/postpre.png';

const Tab = createBottomTabNavigator();

export function ShowBottomTabs(){
  return(
    <Tab.Navigator      screenOptions={{headerShown:      false,      tabBarShowLabel:      false,
    tabBarStyle:{backgroundColor:'#212121',bottom:      responsiveScreenHeight(2),      position:'absolute',
    borderRadius: 30, width: responsiveScreenWidth(90), alignSelf: 'center', left: responsiveScreenWidth(5),
    borderTopColor: 'transparent', height: responsiveScreenHeight(7)}}}>
      <Tab.Screen
        name='HomeTab'
        component={Home}
        options={{
          tabBarIcon: ({focused}) => {
```

```

    if (focused){

        return <Image source={Homebuttpre} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>

    } else {

        return <Image source={Homebutt} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>

    }

}

}

}

/>

<Tab.Screen

name='ScanTab'

component={Scan}

options={{

    tabBarIcon: ({focused}) => {

        if (focused){

            return <Image source={Scanbuttpre} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>

        } else {

            return <Image source={Scanbutt} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>

        }

    }

}

}

/>

<Tab.Screen

name='UploadTab'

component={Upload}

```

```

options={{
  tabBarIcon: ({focused}) => {
    if (focused){
      return <Image source={postpre} style={{width: responsiveScreenWidth(6), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>
    } else {
      return <Image source={post} style={{width: responsiveScreenWidth(6), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>
    }
  }
}}
/>

<Tab.Screen
  name='SettingsTab'
  component={Settings}
  options={{
    tabBarIcon: ({focused}) => {
      if (focused){
        return <Image source={Settingsbuttpre} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>
      } else {
        return <Image source={Settingsbutt} style={{width: responsiveScreenWidth(6.5), height:
responsiveScreenHeight(3), resizeMode:'contain'}}/>
      }
    }
  }}
/>

</Tab.Navigator>

```

```
)  
}
```

4.1.15 MenuAulas/index.js (Formulário de marcação de aulas)

```
import React, { useState, useEffect } from "react";  
import { Text, View, StyleSheet, KeyboardAvoidingView, Platform, TouchableWithoutFeedback, Alert, ScrollView,  
TouchableOpacity } from "react-native";  
import { SafeAreaView } from "react-native-safe-area-context";  
import { collection, onSnapshot, addDoc } from "firebase/firestore"; // Import addDoc  
import { FIRESTORE_DB, auth } from "../config/firebase"; // Import useAuth para obter o ID do usuário atual  
import * as Notifications from 'expo-notifications';  
import { responsiveFontSize, responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from "react-  
native-responsive-dimensions";  
  
export const MenuAulas = () => {  
  const [selectedAula, setSelectedAula] = useState(null);  
  const [aulas, setAulas] = useState([]);  
  
  useEffect(() => {  
    const fetchAulas = async () => {  
      try {  
        const snapshot = await onSnapshot(collection(FIRESTORE_DB, 'aulas'), (querySnapshot) => {  
          const aulasData = [];  
          querySnapshot.forEach((doc) => {  
            aulasData.push({ id: doc.id, ...doc.data() });  
          });  
          setAulas(aulasData);  
        });  
      } catch (error) {  
        console.error('Erro ao buscar aulas:', error);  
      }  
    };  
  
    fetchAulas();  
  }, []);  
  
  const scheduleNotification = async () => {  
    console.log('Agendando notificação...');  
  
    if (!selectedAula) {  
      Alert.alert('Selecione uma aula antes de submeter!');  
      return;  
    }  
  
    try {  
      setTimeout(async () => {  
        const notificationContent = {  
          title: 'Olá, tens uma aula agora! 📅',  

```

```

    body: `Agora tens uma aula de ${selectedAula.nome}.`,
  };

  // Agendar notificação
  const identifier = await Notifications.scheduleNotificationAsync({
    content: notificationContent,
    trigger: { seconds: 2 }
  });

  // Salvar notificação no Firestore
  await addDoc(collection(FIRESTORE_DB, 'notificacoes'), {
    ...notificationContent,
    aulaId: selectedAula.id,
    userId: auth.currentUser.uid, // Adiciona o ID do usuário ao documento de notificação
    timestamp: new Date()
  });

  console.log('Notificação agendada com sucesso:', identifier);

  Alert.alert(
    'Aula agendada com sucesso!',
    `Aula agendada: ${selectedAula.nome}.`,
  );
}, 1000); // Espera 2 segundos antes de agendar a notificação
} catch (error) {
  console.error("Erro ao agendar notificação:", error);
}
};

return (
  <SafeAreaView style={{ flex: 1, backgroundColor: '#090909' }}>
    <KeyboardAvoidingView
      behavior={Platform.OS == "android" ? "padding" : ""}
      style={{
        flex: 1,
        backgroundColor: "black",
      }}
    >
      <View style={{ flex: 1, alignItems: "center" }}>
        <Text style={styles.textSubHeader}>Marcação de aulas</Text>

        <View style={{backgroundColor: '#111111', width: '100%', height: responsiveScreenHeight(90), paddingBottom: responsiveScreenHeight(8), borderRadius: 20}}>

          <ScrollView style={{ width: "100%", paddingHorizontal: 22, marginTop: 64 }}>
            <Text style={{ fontSize: 18, color: 'white', marginBottom: responsiveScreenHeight(2)}}>Aulas Disponíveis:</Text>
            {aulas.map((aula) => (
              <TouchableOpacity key={aula.id} onPress={() => setSelectedAula(aula)}>
                <View style={selectedAula && selectedAula.id === aula.id ? styles.selectedAulaBtn : styles.aulaBtn}>
                  <Text style={{ color: 'white' }}>{aula.nome}</Text>
                </View>
              </TouchableOpacity>
            ))}
          </ScrollView>
        </View>
      </View>
    </SafeAreaView>
  );

```

```

    })
  </ScrollView>

  <TouchableWithoutFeedback onPress={scheduleNotification}>
    <View style={styles.submitBtn}>
      <Text style={{ color: 'white' }}>Submit</Text>
    </View>
  </TouchableWithoutFeedback>

</View>
</View>
</KeyboardAvoidingView>
</SafeAreaView>
);
};

const styles = StyleSheet.create({
  textSubHeader: {
    fontSize: responsiveScreenFontSize(3),
    color: "white",
    marginTop: responsiveScreenHeight(5),
    marginBottom: responsiveScreenHeight(5),
    fontWeight: 'bold',
  },
  aulaBtn: {
    backgroundColor: "#212121",
    paddingVertical: 22,
    alignItems: "center",
    justifyContent: "center",
    borderRadius: 8,
    paddingVertical: 12,
    marginVertical: 8,
  },
  selectedAulaBtn: {
    backgroundColor: "#090909",
    paddingVertical: 22,
    alignItems: "center",
    justifyContent: "center",
    borderRadius: 8,
    paddingVertical: 12,
    marginVertical: 8,
  },
  submitBtn: {
    alignSelf: 'center',
    width: responsiveScreenWidth(90),
    backgroundColor: "#B8E183",
    paddingVertical: 22,
    alignItems: "center",
    justifyContent: "center",
    borderRadius: 8,
    paddingVertical: 12,
    marginBottom: responsiveScreenHeight(3.5)

```

José Carlos da Cunha Rebelo 145

```
},
});
```

```
export default MenuAulas;
```

4.1.16 MenuTreinos/index.js (Página de planos de treino)

```
import React from 'react';

import { StatusBar } from 'expo-status-bar';

import { SafeAreaView, useSafeAreaInsets, SafeAreaView } from 'react-native-safe-area-context';
import { StyleSheet, Image, View, Text, ScrollView, TouchableWithoutFeedback, } from 'react-native';
import { useNavigation } from '@react-navigation/native';

import { responsiveFontSize, responsiveHeight, responsiveWidth, responsiveScreenWidth, responsiveScreenHeight, responsiveScreenFontSize
} from 'react-native-responsive-dimensions';
```

```
import seta from "../../assets/images/arrowLefticon.png";
import pernas from "../../assets/images/pernas.png";
import bracos from "../../assets/images/bracos.jpg";
import core from "../../assets/images/core.png";
import cardio from "../../assets/images/cardio.png";
import vegan from "../../assets/images/vegan.png";
```

```
export const MenuTreinos = () => {
```

```
  const navigation = useNavigation();
```

```
  return (
```

```
    <SafeAreaView style={styles.container}>
```

```
      <View style={{width: responsiveScreenWidth(100), height: responsiveScreenHeight(5), paddingRight:
responsiveScreenWidth(28.5),marginTop: responsiveScreenHeight(5), flexDirection: 'row', justifyContent:'space-between',
alignItems:'center', paddingLeft: responsiveScreenWidth(5), marginBottom: responsiveScreenHeight(3)}}>
```

```
        <TouchableWithoutFeedback onPress={() => navigation.navigate('Home')}>
```

```
          <Image source={seta} style={{resizeMode:'contain',width: responsiveScreenWidth(7), height: responsiveScreenHeight(5)}}/>
```

```
        </TouchableWithoutFeedback>
```

```

<Text style={{color:'white', fontSize: responsiveScreenFontSize(2.5)}}>Escolha um plano:</Text>

</View>

<ScrollView vertical showsVerticalScrollIndicator style={{paddingLeft: responsiveScreenWidth(5), paddingRight: responsiveScreenWidth(5)}}>

    <TouchableWithoutFeedback onPress={() => navigation.navigate('Inferiores')}>

        <View style={styles.option}>

            <Image source={pernas} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

            <View style={styles.overlay} />

            <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Membros inferiores</Text>

        </View>

    </TouchableWithoutFeedback>

    <View style={styles.option}>

        <Image source={bracos} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

        <View style={styles.overlay} />

        <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Membros superiores</Text>

    </View>

    <View style={styles.option}>

        <Image source={core} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

        <View style={styles.overlay} />

        <Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Abdómen e lombar</Text>

    </View>

    <View style={styles.option}>

        <Image source={cardio} style={{width: responsiveScreenWidth(90), height: responsiveScreenHeight(15), borderRadius: 22, position: 'absolute'}}/>

```

```

<View style={styles.overlay} />

<Text style={{color:'white', alignSelf:'center', fontWeight: 'bold', fontSize: responsiveScreenFontSize(3), marginTop: responsiveScreenHeight(5.5)}}>Treino Cardiovascular</Text>

</View>

</ScrollView>

<StatusBar style="light" />

</SafeAreaView>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: 'black',

  },

  overlay: {

    ...StyleSheet.absoluteFillObject,

    backgroundColor: 'rgba(0, 0, 0, 0.4)', // Ajuste a opacidade conforme necessário

    borderRadius: 22,

  },

  option: {

    width: responsiveScreenWidth(90),

    height: responsiveScreenHeight(15),

    alignContent:'center',

    marginBottom: responsiveScreenHeight(3),

  },

});

```

4.1.17 Notifications/index.js (Página de notificações)

```
import React, { useState, useEffect } from 'react';

import { StyleSheet, View, Text, Image, TouchableOpacity, TouchableWithoutFeedback, FlatList, Alert } from 'react-native';

import seta from "../../assets/images/arrowLefticon.png";

import icondel from "../../assets/images/icons8-delete-24.png";

import { useNavigation } from '@react-navigation/native';

import { collection, onSnapshot, query, where, deleteDoc, doc } from 'firebase/firestore'; // Importar onSnapshot
import { FIRESTORE_DB, auth } from '../../config/firebase'; // Importar auth do Firebase
import { responsiveScreenFontSize, responsiveScreenHeight, responsiveScreenWidth } from 'react-native-responsive-dimensions';
import { SafeAreaView } from 'react-native-safe-area-context';

export const Notifi = () => {

  const [notifications, setNotifications] = useState([]); // Estado para armazenar as notificações
  const navigation = useNavigation();

  useEffect(() => {

    // Função para buscar notificações no Firestore
    const fetchNotifications = async () => {

      try {

        const q = query(collection(FIRESTORE_DB, 'notificacoes'), where('userId', '==', auth.currentUser.uid));

        const unsubscribe = onSnapshot(q, (querySnapshot) => {

          const notificationsData = [];

          querySnapshot.forEach((doc) => {

            notificationsData.push({ id: doc.id, ...doc.data() });

          });

          setNotifications(notificationsData);

        });

        return () => unsubscribe(); // Limpando o listener ao desmontar o componente

      } catch (error) {

        console.error('Erro ao buscar notificações:', error);

      }

    };

  });

}
```

```

    }
  };

  fetchNotifications();
}, []); // Executar apenas uma vez ao montar o componente

const confirmDeleteNotification = async (id) => {
  Alert.alert(
    'Confirmar Exclusão',
    'Tem certeza que deseja excluir esta notificação?',
    [
      {
        text: 'Cancelar',
        style: 'cancel',
      },
      { text: 'Sim', onPress: () => deleteNotification(id) },
    ],
    { cancelable: false }
  );
};

const deleteNotification = async (id) => {
  try {
    await deleteDoc(doc(FIRESTORE_DB, 'notificacoes', id));
  } catch (error) {
    console.error('Erro ao excluir notificação:', error);
  }
};

return (
  <SafeAreaView style={styles.container}>

```

```

<View style={styles.header}>

  <Text style={styles.title}>Notificações</Text>

</View>

<View style={{width:'100%', height: responsiveScreenHeight(85), backgroundColor:'#131313', borderRadius: 20,padding:
responsiveScreenWidth(5), paddingTop: responsiveScreenHeight(4), alignItems: 'center'}}>

  {notifications.length === 0 ? (

    <View style={styles.noNotificationContainer}>

      <Text style={styles.noNotificationText}>Não existem notificações por agora.</Text>

    </View>

  ) : (

    <FlatList

      data={notifications}

      keyExtractor={(item, index) => index.toString()}

      renderItem={({ item }) => (

        <View style={styles.notificationItem}>

          <Text style={{fontSize: responsiveScreenFontSize(2.5), marginTop: responsiveScreenHeight(0.5)}}>🔔</Text>

          <View>

            <Text style={styles.notificationContent}>{item.body}</Text>

            <Text style={styles.notificationDate}>{new Date(item.timestamp.seconds * 1000).toLocaleString()}</Text>

          </View>

          <TouchableWithoutFeedback onPress={() => confirmDeleteNotification(item.id)} style={{marginTop: responsiveScreenHeight(1)}}>

            <Image source={icondel}></Image>

          </TouchableWithoutFeedback>

        </View>

      )}

    </FlatList>

  )}

</View>

```

```
        />

    }}

</View>

</SafeAreaView>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#090909',

  },

  noNotificationContainer: {

    flex: 1,

    alignItems: 'center',

    justifyContent: 'center',

  },

  noNotificationText: {

    color: 'white',

    fontSize: responsiveScreenFontSize(2)

  },

  notificationItem: {

    backgroundColor: '#212121',

    padding: responsiveScreenWidth(4),

    marginBottom: responsiveScreenHeight(2),

    borderRadius: 15,

    flexDirection: 'row',

    alignItems: 'center',

    width: responsiveScreenWidth(90),

    justifyContent: 'space-between',

  },

});
```

```
notificationContent: {  
  color: 'white',  
  fontSize: responsiveScreenFontSize(1.8)  
},  
notificationDate: {  
  color: 'white',  
  fontSize: 14,  
  marginTop: 5,  
},  
  
title: {  
  fontSize: responsiveScreenFontSize(3),  
  color: 'white',  
  alignSelf: 'center',  
  fontWeight: 'bold',  
},  
  
header: {  
  width: '100%',  
  height: responsiveScreenHeight(10),  
  justifyContent: 'center',  
},  
  
});
```

4.1.18 Profile/index.js (Foto de perfil do utilizador)

```
import React, { useState, useEffect } from 'react';  
import { View, Image, StyleSheet } from 'react-native';  
import { getStorage, ref, app, getDownloadURL, listAll } from 'firebase/storage';  
import { getAuth } from 'firebase/auth';  
import { responsiveScreenWidth, responsiveScreenHeight } from 'react-native-responsive-dimensions';
```

José Carlos da Cunha Rebelo 153

```
export const Profile = ({ uid }) => {  
  
  const [imageUrl, setImageUrl] = useState(null);  
  
  const storage = getStorage(app);  
  
  
  const fetchLatestImage = async () => {  
  
    const auth = getAuth();  
  
    const user = auth.currentUser;  
  
    const userId = user.uid;  
  
  
    // Construct the path to the user's profile folder  
  
    const profileRef = ref(storage, `profile/${userId}`);  
  
    const listResult = await listAll(profileRef);  
  
  
    // Sort images by date (assuming creation time is reflected in file name)  
  
    const sortedImages = listResult.items.sort((a, b) => b.fullPath.localeCompare(a.fullPath));  
  
  
    // Get download URL for the most recent image (if any)  
  
    if (sortedImages.length > 0) {  
  
      const mostRecentRef = sortedImages[0];  
  
      const url = await getDownloadURL(mostRecentRef);  
  
      setImageUrl(url);  
  
    } else {  
  
      setImageUrl(null); // Set to null if no images found  
  
    }  
  
  };  
  
  
  useEffect(() => {  
  
    fetchLatestImage();  
  
  
    // Optional update interval (uncomment if needed)
```

```
const intervalId = setInterval(fetchLatestImage, 5000); // Update every 10 seconds

return () => clearInterval(intervalId);

}, [storage, uid]); // Only depend on storage and uid

return (
  <View style={styles.container}>
    {imageUrl && (
      <Image source={{ uri: imageUrl }} style={styles.image} />
    )}
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    alignItems: 'center',
    justifyContent: 'center',
  },
  image: {
    width: responsiveScreenWidth(25),
    aspectRatio: 1, // Manter a proporção da imagem
    resizeMode: 'cover', // Ajustar a imagem para cobrir toda a área disponível
    borderRadius: 9999, // Usar um valor grande o suficiente para garantir o arredondamento completo
    marginBottom: responsiveScreenHeight(1),
  },
});

export default Profile;
```

Conclusão

Concluí este projeto com a sensação de ter superado desafios que ultrapassaram as minhas expectativas. Ao longo dos meus três anos na EPB, deparei-me com uma variedade de obstáculos que jamais imaginei enfrentar. Desde atividades relacionadas ao meu curso até aquelas completamente alheias à minha área de estudo, cada uma delas contribuiu para o meu crescimento pessoal e profissional. Participar ativamente de todas essas atividades não foi apenas uma escolha, mas sim uma parte essencial da minha jornada. Isso permitiu-me não apenas ampliar o meu conhecimento, mas também desenvolver as minhas habilidades interpessoais. Apesar dos contratempos enfrentados durante o desenvolvimento da minha prova de aptidão profissional, sinto que não alcancei exatamente o resultado desejado. Acredito que, se tivesse sido mais eficiente na gestão do tempo, poderia ter implementado funcionalidades adicionais.

Bibliografia

Web grafia

React Native - <https://reactnative.dev/>

JavaScript - <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

Android Studio - <https://developer.android.com/studio/intro>

Firebase - <https://rockcontent.com/br/blog/firebase/>

VS Code - <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>

HTML-<https://developer.mozilla.org/pt-BR/docs/Web/HTML>

CSS - <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>

Netlify - <https://www.netlify.com/>

Anexos

Anexo 1 – Curriculum Vitae


europass



José Rebelo

Nacionalidade: Portuguesa Data de nascimento: 24/03/2005 Sexo: Masculino

Número de telemóvel: (+351) 924041637

Endereço de email: zcarebelo05@gmail.com

Linkedin: <https://www.linkedin.com/in/josé-rebelo-b0963a261/>

Casa: Rua de Boa Vista 1º andar, Braga (Portugal)

SOBRE MIM

Olá, chamo-me José e sou um entusiasta apaixonado por programação. Ao longo da minha jornada digital, adquiri habilidades sólidas em várias linguagens e tecnologias.

Quando se trata de Python, tenho domínio dos conceitos básicos e consigo criar soluções eficientes para desafios simples. Na área de C#, tenho experiência desde a criação de aplicações Windows Forms até a integração de sistemas com bases de dados.

No desenvolvimento web, aprendi linguagens como o PHP e o Javascript combinados com o HTML e CSS para estilizar e encorporar as páginas.

Em relação ao desenvolvimento de aplicações móveis, aprendi conceitos básicos de React Native com Expo.

COMPETÊNCIAS LINGUÍSTICAS

Língua(s) materna(s): português

Outra(s) língua(s):

Inglês

COMPREENSÃO ORAL B1 LEITURA B1 ESCRIVER B1

PRODUÇÃO ORAL B2 INTERAÇÃO ORAL B1

Níveis: A1 e A2: Utilizador de base; B1 e B2: Utilizador independente; C1 e C2: Utilizador avançado

DISTINÇÕES E PRÉMIOS

Microsoft Office Specialist - Excel & Word

Certiport [25/01/2023]

Ligação: <https://files.fm/u/sf2myrscdp>

VOLUNTARIADO

EMACI 2021

[Braga]

EXPERIÊNCIA PROFISSIONAL

Estagiário no departamento de informática

Cruz Vermelha Portuguesa - Delegação de Braga [05/06/2023 – 25/07/2023]

Cidade: Braga

País: Portugal