

Eventer

CURSO PROFISSIONAL DE GESTÃO E
PROGRAMAÇÃO DE SISTEMAS
INFORMÁTICOS

Eventer.

Josias de Araújo Lopo

Trabalho realizado sob a orientação de:

Prof. António Araújo

Escola Profissional de Braga
Braga, 10 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 ano letivo de 2023/2024 e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.

Dedicatória

Dedico o meu projeto de aptidão profissional a todos os meus professores, aos meus amigos e aos meus pais que me apoiaram no meu percurso de escolar.

Agradecimentos

Gostaria de expressar minha sincera gratidão a todas as pessoas que contribuíram para a realização do projeto da PAP. Em primeiro lugar, gostaria de agradecer ao professor António Araújo, cujo apoio e orientação foram fundamentais para o desenvolvimento. A maneira que me ensinou a aprender a programar foi essencial.

Aos meus pais, Ezequiel e Liliana, apoio constante e encorajamento ao longo de toda a jornada, este projeto não teria sido possível.

Aos meus amigos e colegas, cujo apoio e colaboração foram inestimáveis. Suas ideias, feedbacks e incentivo foram essenciais para o desenvolvimento e aprimoramento do Eventer.

Por fim, gostaria de expressar meus agradecimentos a todos os envolvidos no projeto, cuja as ideias e opiniões foram fundamentais para alcançar os resultados apresentados neste projeto.

Índice

Introdução	10
Fundamentação	10
Objetivos	11
Recursos	13
1. Componente Teórica	14
1.1 Linguagens de Programação	14
1.1.1 <i>React Native</i>	14
1.1.2 <i>JavaScript</i>	14
1.1.3 <i>Html</i>	15
1.1.4 <i>Css</i>	15
1.2 Serviços e Aplicações para o desenvolvimento	15
1.2.1 <i>Visual Studio Code</i>	15
1.2.2 <i>Expo</i>	16
1.2.3 <i>Figma</i>	16
1.2.4 <i>Notion</i>	17
1.2.5 <i>Xcode</i>	17
1.2.6 <i>Android Studio</i>	17
1.2.7 <i>Firebase</i>	18
2. Fazes Desenvolvimento	18
2.1 Esboços (<i>Mockups</i>) no <i>Figma</i>	18

2.2	Entidade visual e <i>design</i>	21
2.2.1	Logo cores e <i>Hall</i> (página entre login e registo).....	21
2.2.2	Abertura da aplicação.....	22
2.3	Desenvolvimento da aplicação.....	22
2.3.1	Aprendizagem <i>React Native</i>	22
2.3.2	Página de Login e Registo	23
2.4	Versão Entregue.....	24
2.5	Problemas e Dificuldades	26
2.6	Website	26
2.6.1	<i>Landing Page</i>	26
3.	Código	28
3.1	Aplicação.....	29
3.1.1	App.js	29
3.1.2	Firebase.js (api do firebase censurada)	41
3.1.3	useAuth.js	42
3.1.4	ChatBubble.js.....	44
3.1.5	ChatGpt.js (Api Censurada).....	46
3.1.6	ChatPage.js	58
3.1.7	ChatsScreen.js.....	71
3.1.8	CreateNote.js	81
3.1.9	Default.js.....	85
3.1.10	Gemini.js (api censurada)	98

3.1.11	Hall.js	110
3.1.12	Home.js	118
3.1.13	List.js.....	137
3.1.14	ListWidget.js	146
3.1.15	Login.js	152
3.1.16	MediaWidget.js	166
3.1.17	Menu.js.....	174
3.1.18	NewPost.js.....	175
3.1.19	NoteDetail.js.....	188
3.1.20	Notes.js.....	197
3.1.21	Profile.js.....	204
3.1.22	Regist.js	216
3.1.23	Todo.js.....	229
3.1.24	ProfilePic.js	231
3.1.25	ProfilePic2.js	235
3.1.26	.gitignore	239
3.1.27	App.json.....	241
3.1.28	babel.config.js	243
3.1.29	metro.config.js	244
3.1.30	package.json.....	244
3.1.31	react-native.config.js	248
3.2	Código Website	248

3.2.1	Index.html.....	248
3.2.2	Mobile.html	256
3.2.3	Redirect.js	263
4.	Outros projetos	264
4.1	NOKAUTE.pt	264
4.2	Innovationpro.pt	264
4.3	jogOleo	265
4.4	Davdsm.pt	265
	Conclusão	265
	Bibliografia.....	266
	Webgrafia	266
	Anexos	267

Índice de figuras

Figura 1 - Logótipo React Native.....	14
Figura 2 - Logótipo JavaScript.....	14
Figura 3 - Logótipo HTML	15
Figura 4 - Logótipo CSS	15
Figura 5 - Logótipo Visual Studio Code	15
Figura 6 - Logótipo Expo	16

Figura 7 - Logótipo Figma	16
Figura 8 - Logótipo AMPPS	17
Figura 9 - Logótipo Xcode	17
Figura 10 - Logótipo Android Studio.....	17
Figura 11 - Logótipo Firebase	18

Introdução

Fundamentação

O Eventer é uma aplicação móvel inovadora desenvolvida para simplificar a gestão pessoal e profissional. Projetado para oferecer uma experiência completa de organização, o Eventer combina diversas funcionalidades essenciais em uma única plataforma acessível de qualquer lugar.

Gerencie seus compromissos diários de forma eficiente com a lista de tarefas integrada, podendo usar tecnologia de inteligência artificial para otimizar suas atividades ou sugerir novas tarefas. Registe ideias e informações importantes de maneira rápida e organizada através do recurso de anotações, garantindo que nada seja esquecido.

O compartilhamento de fotos com descrições detalhadas permite criar um registo visual de momentos significativos ou inspirações. Além disso, as mensagens rápidas facilitam o envio de lembretes e notas importantes para si mesmo.

A integração com inteligência artificial através das *APIs* do *Gemini* e *ChatGPT* eleva ainda mais a experiência do usuário, oferecendo assistência personalizada e interações inteligentes em todos os aspetos da aplicação.

Objetivos

- *Reforçar a capacidade de superar situações inesperadas ou imprevistas nas diferentes fases do projeto.*
- *Desenvolver o espírito crítico e sentido de autonomia.*
- *Desenvolver competências de decisão, escolha e procura de informação.*
- *Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas.*
- *Aumentar a minha capacidade de domínio sobre este tema.*
- *Melhorar a disciplina mental, capacidade de organização, metodologia de trabalho.*
- *Desenvolver competências nos desenvolvimentos de aplicações moveis.*
- *Aplicar conhecimentos e competências adquiridos na área de programação em dispositivos móveis.*
- *Aumentar a capacidade e domínio das linguagens React-Native, HTML , CSS e JavaScript.*
-

Cronograma		
Fase	Calendarização	Elenco de atividades nas diversas fases
Conceção	Setembro 2023	Conhecimento do regulamento
	Setembro 2023	Definição dos critérios da PAP pela direção
	Setembro 2023	Período de negociação / reflexão / decisão sobre os temas / problemas a explorar

	Setembro 2023	Normas para a elaboração de um trabalho científico
	Até 31 outubro 2023	Entrega do projeto de acordo com as normas do regulamento
Desenvolvimento	Outubro 2023	Escolha do tema, baseado no interesse por eventos
	Outubro 2023	Escolha do nome
	Novembro 2023	Decisão final sobre o desenvolvimento da aplicação
	Novembro 2023	Início estudo sobre React Native
	Dezembro 2023	Início do desenvolvimento da App
	Maio 2024	Acabar Aplicação
	Maio 2024	Entrega do relatório
Autoavaliação e Relatório	Janeiro 2024	Defesa intermédia da PAP
	12 maio 2024	Conclusão do projeto e entrega do relatório
	Julho 2024	Defesa final da PAP

Recursos

Recursos		
Humanos	Materiais	Financeiros
Eu	Computador	
Prof. António Araújo	Visual Studio Code, GitHub, Figma, Xcode, Expo, Photoshop, Dribble, Notion	
Prof. Manuel Ferreira	Figma, React Native, HTML, CSS, JavaScript, Nodejs, Firebase	

1. Componente Teórica

Para a realização do meu projeto tive de recorrer ao uso de várias ferramentas, variando desde linguagens de programação para o desenvolvimento em **Android** e **IOS**, como **React Native** e até editores **IDE (integrated Development Enviroment)** como o Visual Code.

1.1 Linguagens de Programação

1.1.1 React Native

Foi usado **React Native (Framework de JavaScript)** na produção da minha PAP porque é mais fácil no desenvolvimento de aplicações multiplataforma, ajuda a manter um desenvolvimento orientado organizado, flexível e extensível. Na minha opinião **React Native** é uma melhor linguagem para lidar com vários processos e desempenhos comparado a **Flutter** por exemplo.

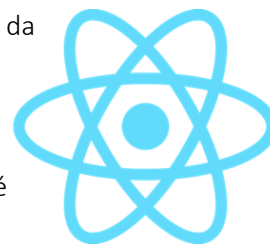


Figura 1 - Logótipo React Native

1.1.2 JavaScript

Foi usado **JavaScript** como linguagem principal para escrever a lógica do aplicativo e para definir a interface do utilizador. **Javascript** é uma linguagem Ubíqua, o desenvolvimento é mais rápido a reutilização de código é mais fácil e tem um ecossistema rico.



Figura 2 - Logótipo JavaScript

1.1.3 Html

Foi usado *HTML* para desenvolvimento da *landing page*, website para promoção da aplicação



Figura 3 - Logótipo HTML

1.1.4 Css

Foi usado CSS para dar estilo ao documento *HTML*



Figura 4 - Logótipo CSS

1.2 Serviços e Aplicações para o desenvolvimento

1.2.1 Visual Studio Code

Foi usado **Visual Code** (*IDE - integrated Development Enviroment*) como editor de código, usado vários plugins desde GitHub Copilot e Prettier.



Figura 5 - Logótipo Visual Studio Code

1.2.2 Expo

Foi usado **Expo Enviroment** como o debugger e execução da APP em tempo real.



Figura 6 - Logótipo Expo

1.2.3 Figma

Foi usado o **Figma** como ferramenta de Design e Apresentação.

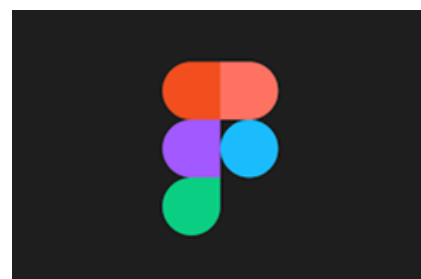


Figura 7 - Logótipo Figma

1.2.4 Notion

Foi usado **Notion** para guardar apontar e organizar dados como um bloco de notas.



Figura 8 - Logótipo AMPPS

1.2.5 Xcode

Foi usado **Xcode** como simulador de iphone para otimizar a responsividade nos dispositivos APPLE.



Figura 9 - Logótipo Xcode

1.2.6 Android Studio

Foi usado **Android Studio** como simulador de android para otimizar a responsividade nos dispositivos Android.



Figura 10 - Logótipo Android Studio

Josias Lopo 17

1.2.7 Firebase

Foi usado **Firebase** base de dados e armazenamento de dados.



Figura 11 - Logótipo
Firebase

2. Fazes Desenvolvimento

2.1 Esboços (*Mockups*) no Figma

A concretização das primeiras ideias e objetivos de funcionalidades foi realizada levando em consideração os aspetos de design, acessibilidade e lógica de páginas. Durante esse processo, trabalhei para garantir que a interface fosse intuitiva e fácil de usar para todos os utilizadores, independentemente de suas necessidades especiais. Além disso, foquei em criar uma estrutura lógica para as páginas, garantindo uma navegação fluída e coerente. Após um trabalho dedicado, alcancei os seguintes resultados:

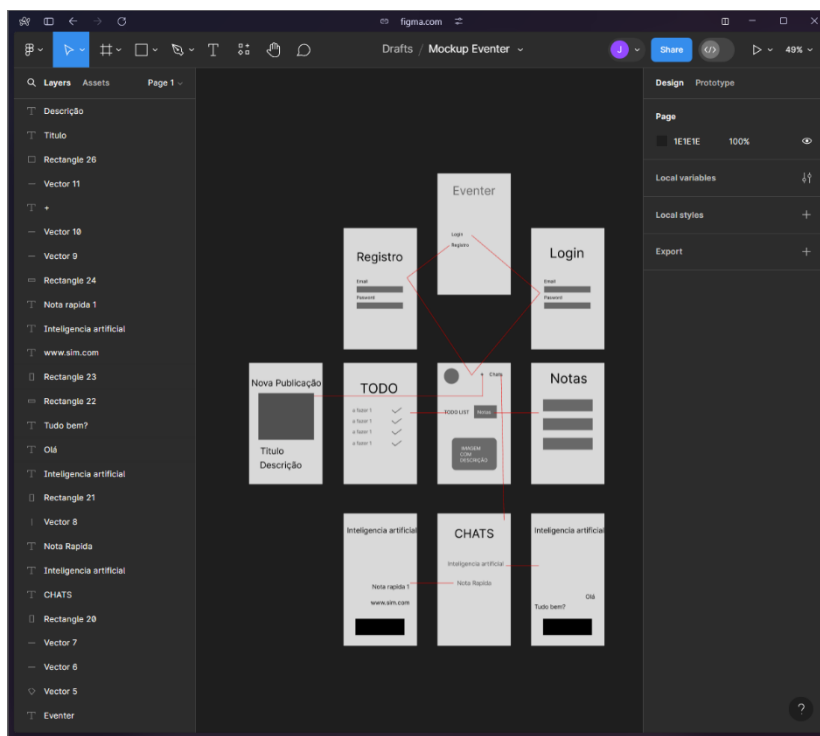


Imagem 1- Mockup 1

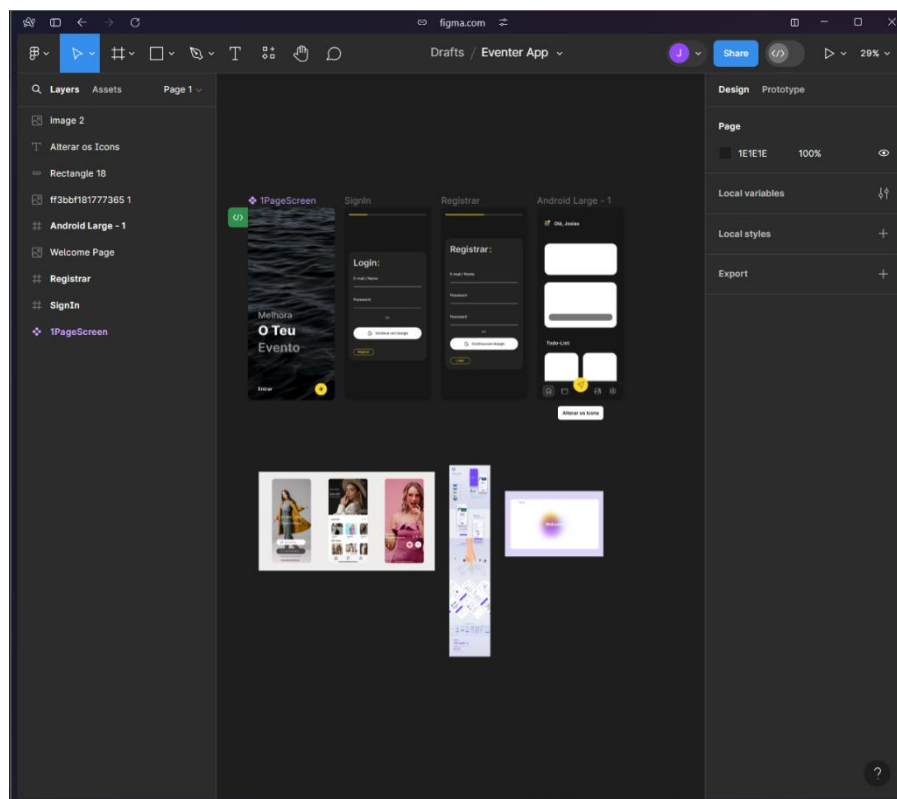


Imagem 2 - Mockup 2

2.2 Entidade visual e design

Desenvolvimento final da entidade visual da aplicação, *design* e logótipo. (Novembro 2023)

2.2.1 Logo cores e Hall (página entre login e registo)

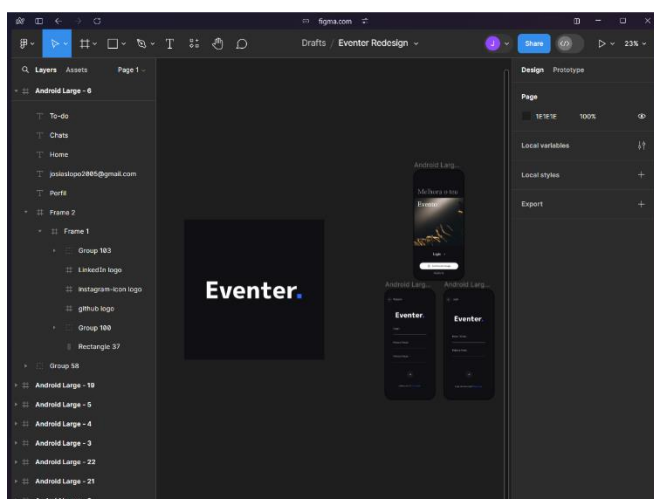


Imagem 3 - Logo e Hall

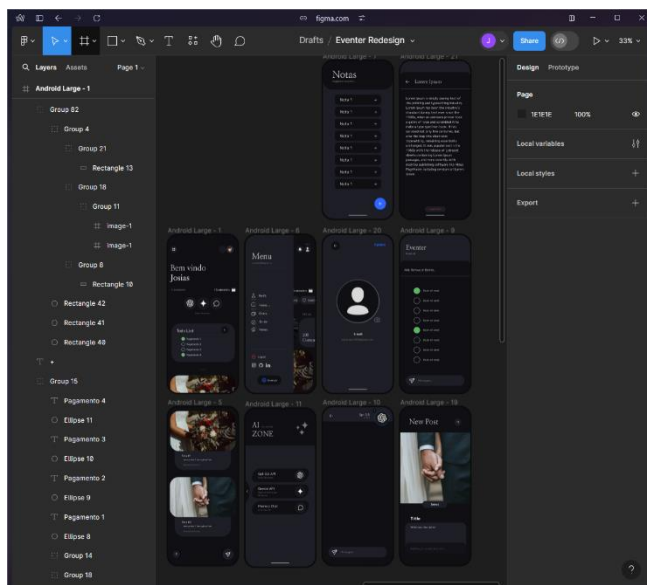


Imagem 4 - Design completo

2.2.2 Abertura da aplicação

A tela inicial da aplicação foi projetada para o carregamento de imagens e a inicialização das páginas, visando garantir que a aplicação inicie sem nenhuma anomalia. Desenvolvi uma abertura para a aplicação de forma profissional e estética, proporcionando uma experiência visual agradável aos utilizadores.

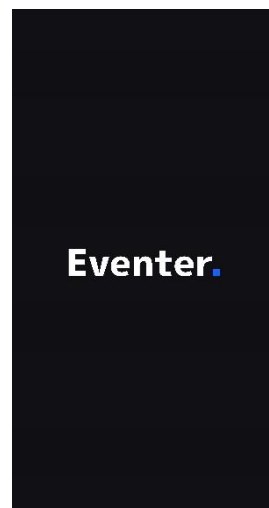


Imagem 5 - Inicialização App

2.3 Desenvolvimento da aplicação

A fase inicial do projeto foi bastante desafiadora, pois eu não possuía conhecimento prévio sobre o desenvolvimento de aplicações nem sobre as linguagens utilizadas. Foi necessário aprender muito em um curto espaço de tempo para superar essas dificuldades.

2.3.1 Aprendizagem React Native

Para aprender a programar em *React Native*, decidi realizar um pequeno projeto de imitação da *Prime Video*. Ao finalizá-lo, adquiri conhecimentos suficientes para desenvolver a minha própria aplicação.

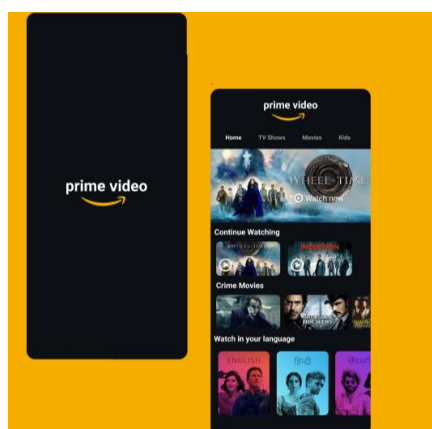


Imagem 6 - Primeira App

2.3.2 Página de Login e Registo

Com a integração de uma autenticação via *Firebase Authentication*, desenvolvi as páginas de login e registo. Ambas possuem campos para inserção de email e senha, sendo o registo utilizado para criar uma conta e o login para iniciar a sessão.

Durante o processo de criação de uma conta, as senhas são criptografadas visando garantir a segurança e privacidade dos usuários.

josias@gmail.com	✉	12 de mai. de 20...	12 de mai. de 20...	whZWeV4IW8Rd7EGCvGCaV0...
teste2@gamil.com	✉	12 de mai. de 20...	12 de mai. de 20...	y4J1U39pePbVfQot8jiDrYmuY...
macacada@gmail.com	✉	12 de mai. de 20...	12 de mai. de 20...	76FhvZtPiKOUUBnBe8C2ZmM...
teste@gmail.com	✉	12 de mai. de 20...	12 de mai. de 20...	BZvZVfjamJSK8gczJBcHhPBj...

Imagem 7 - Palavra-passe incryptada

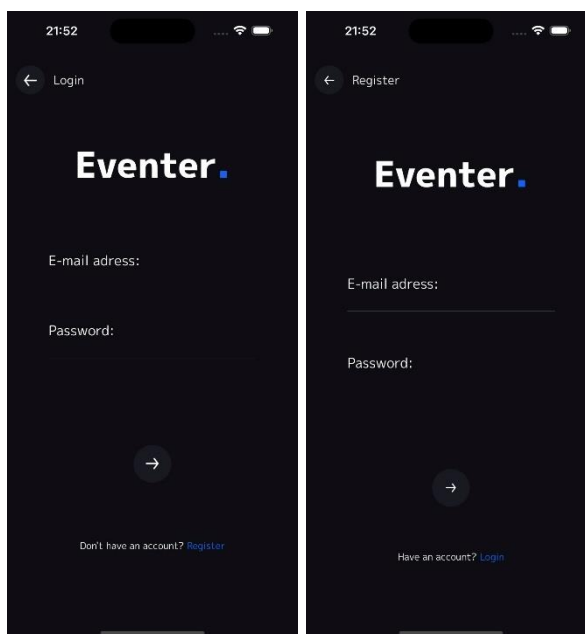


Imagem 9 - Login

Imagem 8 - Registo

2.4 Versão Entregue

Realizada a primeira versão da aplicação antes do prazo de entrega,

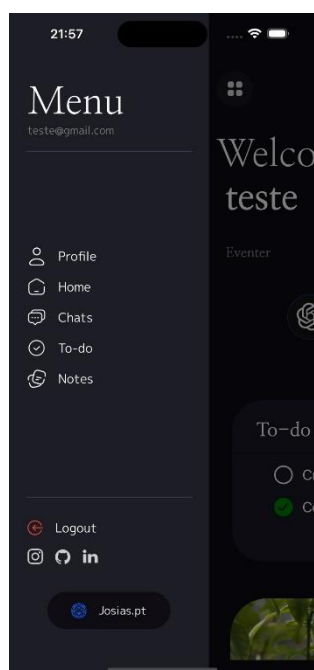


Imagem 10 – Menu lateral

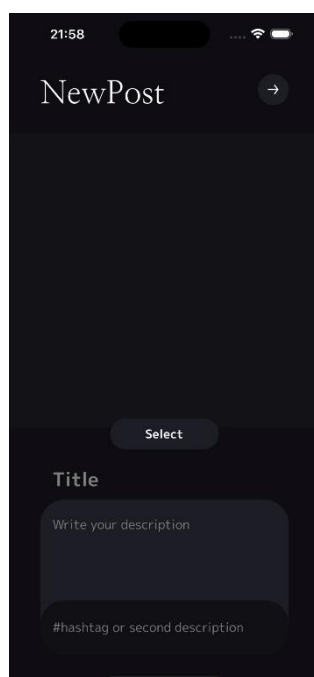


Imagem 13 – Página Nova Publicação

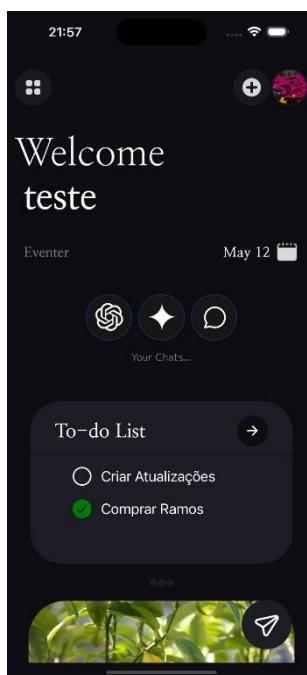


Imagem 11 - Página Inicial



Imagem 12 - Página Chat Rápido

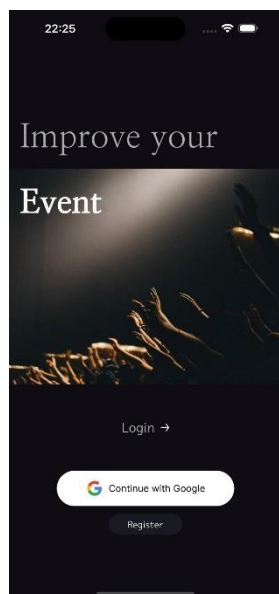


Imagem 16 - Página Hall

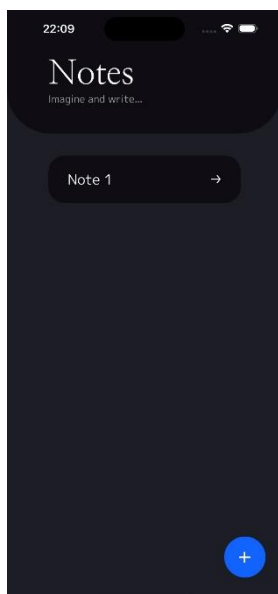


Imagem 15 - Página Notas

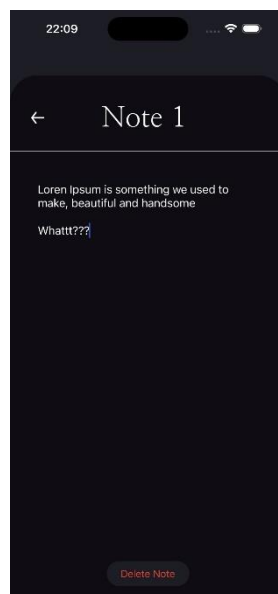


Imagem 14 - Página Nota

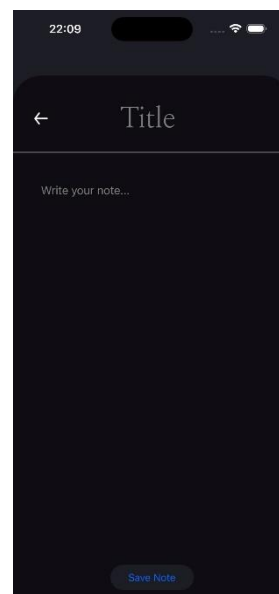


Imagem 17 - Página Criar Nota

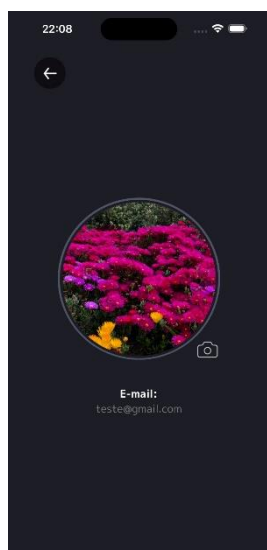


Imagem 18 - Página Perfil



Imagem 21 - Página Gemini

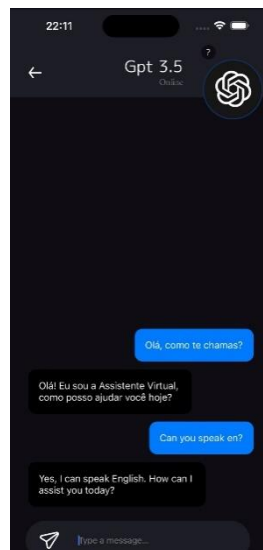


Imagem 19 - Página GPT



Imagem 20 - Página Todo

2.5 Problemas e Dificuldades

Durante o desenvolvimento, deparei-me com diversos desafios para escrever código limpo para *Android* e *iOS* e torná-lo responsivo. Tive de aprender novos métodos e distinguir as duas plataformas, pois cada uma possui seu próprio método de processamento. Uma das maiores dificuldades enfrentadas foi a atualização de pacotes e importações após uma atualização do *Expo* ou do *React Native*. Na primeira vez que isso ocorreu, nada na minha PAP funcionava, levando quase uma semana para resolver até conseguir retomar o desenvolvimento da aplicação.

Os erros relacionados ao *Firebase* foram dos mais problemáticos. Foi necessário otimizar o espaço, compreender lógicas de conexão e diferentes *SDKs* do *Firebase*. Sempre que tentava adicionar uma nova funcionalidade, a importação do *Firebase* no código precisava ser modificada.

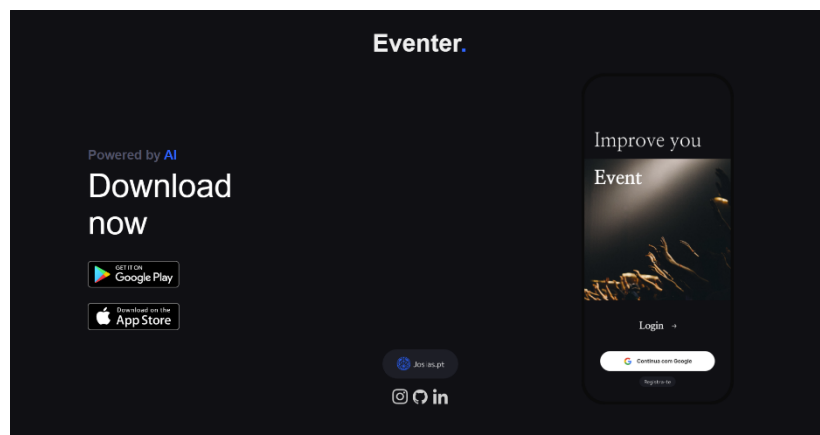
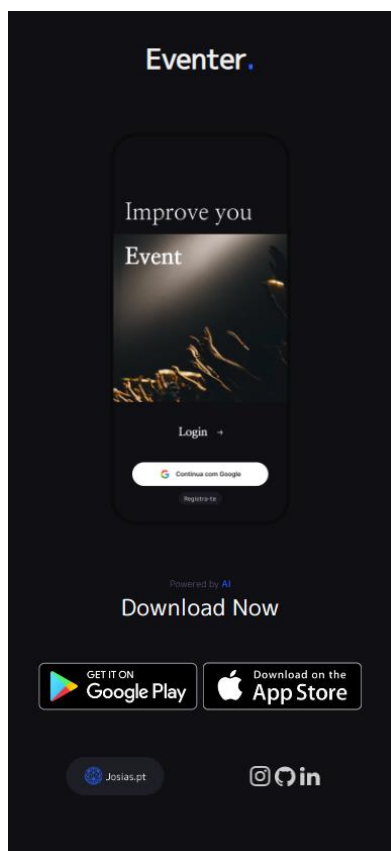
As *APIs* de inteligência artificial também foram desafiadoras, começando com a do *GPT-3.5*, que não mantinha um histórico de mensagens e não retornava nenhuma mensagem após a primeira interação. No entanto, a *API* do *Gemini* foi ainda mais complicada, pois só funciona fora da Europa e não mantém um histórico (característica própria da *API*). Além disso, a documentação do *Gemini* é fraca e a comunidade é limitada, o que torna a implementação mais difícil.

2.6 Website

Desenvolvi um website para promoção da aplicação, tem como base redirecionar para os links das lojas que a aplicação será publicada.

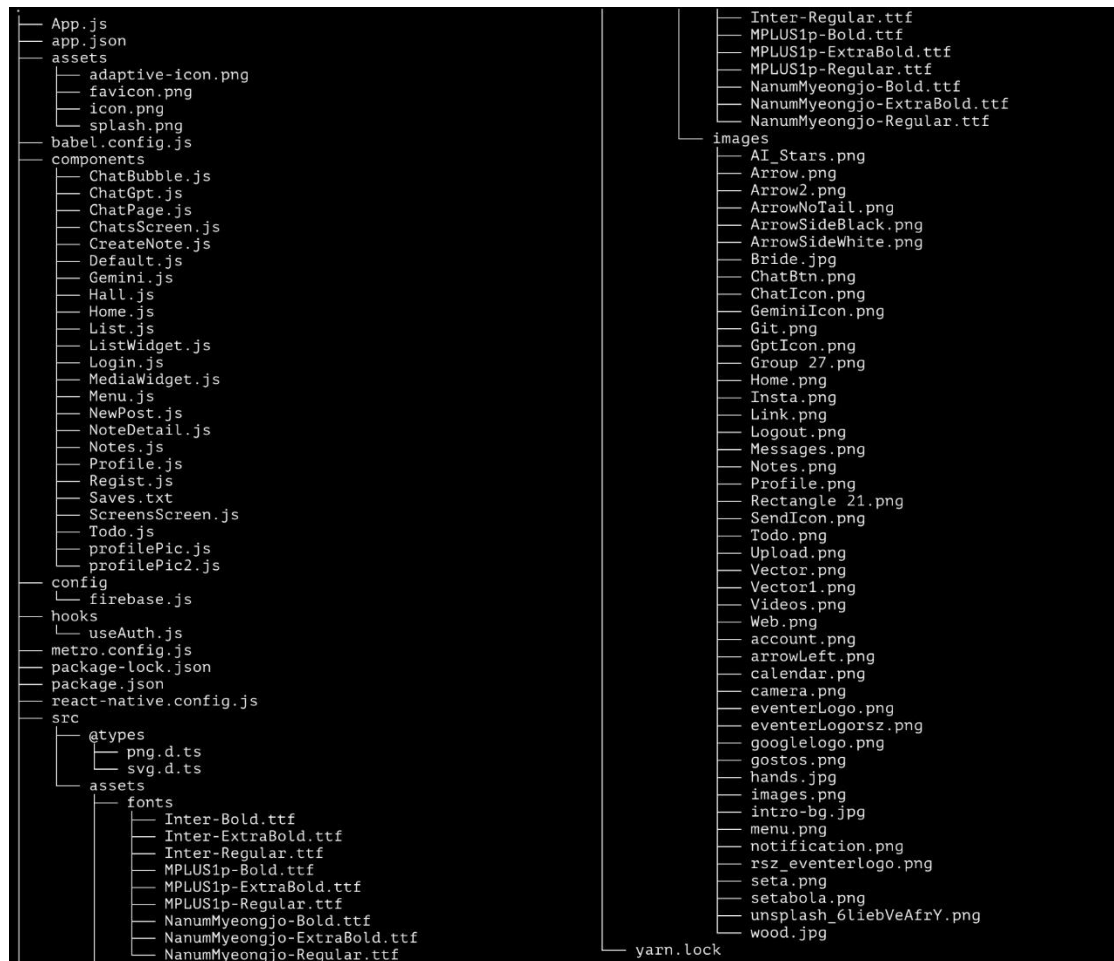
2.6.1 Landing Page

O website é de uma página só (*one page only*), contem uma foto do hall de entrada da app com o slogan da aplicação seguido dos botões de download e links para as minhas redes.



3. Código

O código está formatado no seguinte esquema:



Todos os ficheiros não descritos são parte da arquitetura do *React Native*, o que faz a aplicação conseguir iniciar, são ficheiros comuns em *Expo*. Em *React Native* é boa pratica utilizar os nomes das páginas com início Maiúsculo, tirando páginas de conteúdos não mostrados diretamente.

App.js é onde toda a logica de navegação, inicialização, e menu lateral está.

assets é a pasta onde está guardado a tela de inicialização e ícone da aplicação.

components é a pasta onde todos os ecrãs estão guardados.

config é a pasta onde está guardado o ficheiro de configuração do firebase.

hooks é a pasta onde está guardado o ficheiro de autenticação de utilizadores.

src é a pasta onde está guardada as fontes e as imagens usadas na aplicação e os ícones.

3.1 Aplicação

3.1.1 App.js

```
import React, { useEffect, useState } from 'react';  
  
import { NavigationContainer } from "@react-navigation/native";  
  
import { createNativeStackNavigator } from "@react-navigation/native-stack";  
  
import ScreensScreen from "./components/ScreensScreen";  
  
import Regist from "./components/Regist";  
  
import Hall from "./components/Hall";  
  
import Login from "./components/Login";  
  
import Home from "./components/Home";
```

```
import Todo from './components/Todo';

import Default from './components/Default';

import Gemini from './components/Gemini';

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

import useAuth from './hooks/useAuth';

import { createDrawerNavigator } from '@react-navigation/drawer';

import { View, Text, Linking, Image } from 'react-native';

import { DrawerContentScrollView, DrawerItem } from '@react-navigation/drawer';

import { Avatar, Icon, Title } from 'react-native-paper';

import { TouchableOpacity } from 'react-native-gesture-handler';

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

import { auth } from './config/firebase';

import { signOut } from 'firebase/auth';

import 'firebase/auth';

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

import { Ionicons, Entypo } from '@expo/vector-icons';

import Profile from './components/Profile';

import ChatsScreen from './components/ChatsScreen';

import ChatGpt from './components/ChatGpt';

import ChatPage from './components/ChatPage';

import MediaWidget from './components/MediaWidget';
```

```
import NewPost from './components/NewPost';

import Notes from './components/Notes';

import NoteDetail from './components/NoteDetail';

import CreateNote from './components/CreateNote';

import Git from './src/assets/images/Git.png';

import Link from './src/assets/images/Link.png';

import Insta from './src/assets/images/Insta.png';

import Logout from './src/assets/images/Logout.png';

import Web from './src/assets/images/Web.png';

import ProfileImg from './src/assets/images/Profile.png';

import HomeImg from './src/assets/images/Home.png';

import Messages from './src/assets/images/Messages.png';

import TodoImg from './src/assets/images/Todo.png';

import NotesImg from './src/assets/images/Notes.png';

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

function DrawerContent(props) {

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

  const navigation = useNavigation();
```

```
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 handleLogout = async () => {  
  
  try {  
  
    await signOut(auth);  
  
    props.navigation.dispatch(  
  
      CommonActions.reset({
```

```

    index: 0,

    routes: [{ name: 'Hall' }],

  })

);

} catch (error) {

  console.error("Error logging out:", error);

}

};

return (

  <View style={{flex: 1,}}>

    <DrawerContentScrollView {...props}>

      <View style={{}}>

        <View>

          <View>

            <View style={{flex: 1,paddingLeft: responsiveWidth(2), paddingRight:
responsiveWidth(2)}}>

              <View style={{marginLeft: responsiveWidth(4)}}>

```

```
<Text style={{marginTop: responsiveWidth(7), color: 'white', fontFamily:
'NanumMyeongjo', fontSize: responsiveFontSize(6),}}>Menu</Text>
```

```
<Text numberOfLines={1} style={{ color: '#818181', fontFamily: 'MPLUS1p',
fontSize:responsiveWidth(3.5)}}>{userEmail}</Text>
```

```
</View>
```

```
<View style={{
```

```
borderBottomColor: '#525468',
```

```
borderBottomWidth: 1,
```

```
width: responsiveHeight(25),
```

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

```
marginTop: responsiveHeight(2),
```

```
}}></View>
```

```
<View style={{gap: responsiveWidth(4), height: responsiveHeight(50),
justifyContent: "center", marginLeft: responsiveWidth(4)}}>
```

```
<TouchableOpacity onPress={() => navigation.navigate('Profile')} style={{ gap:
responsiveWidth(4),flexDirection:"row", alignItems: "center", marginRight:
responsiveWidth(4)}}><Image source={ProfileImg} style={{height: responsiveWidth(6), width:
responsiveWidth(6), resizeMode: "contain"}} /><Text style={{color: "white", fontFamily:
"MPLUS1p", fontSize: responsiveFontSize(2)}}>Profile</Text></TouchableOpacity>
```

```
<TouchableOpacity onPress={() => navigation.navigate('Home')} style={{ gap:
responsiveWidth(4),flexDirection:"row", alignItems: "center", marginRight:
responsiveWidth(4)}}><Image source={HomeImg} style={{height: responsiveWidth(6), width:
responsiveWidth(6), resizeMode: "contain",}} /><Text style={{color: "white", fontFamily:
"MPLUS1p", fontSize: responsiveFontSize(2)}}>Home</Text></TouchableOpacity>
```

```
<TouchableOpacity onPress={() => navigation.navigate('ChatsScreen')} style={{
gap: responsiveWidth(4),flexDirection:"row", alignItems: "center", marginRight:
responsiveWidth(4)}}><Image source={Messages} style={{height: responsiveWidth(6), width:
responsiveWidth(6), resizeMode: "contain",}} /><Text style={{color: "white", fontFamily: "MPLUS1p",
fontSize: responsiveFontSize(2)}}>Chats</Text></TouchableOpacity>
```

```
<TouchableOpacity onPress={() => navigation.navigate('Todo')} style={{ gap:
responsiveWidth(4),flexDirection:"row", alignItems: "center", marginRight:
responsiveWidth(4)}}><Image source={TodoImg} style={{height: responsiveWidth(6), width:
responsiveWidth(6), resizeMode: "contain",}} /><Text style={{color: "white", fontFamily: "MPLUS1p",
fontSize: responsiveFontSize(2)}}>To-do</Text></TouchableOpacity>
```

```
<TouchableOpacity onPress={() => navigation.navigate('Notes')} style={{ gap:
responsiveWidth(4),flexDirection:"row", alignItems: "center", marginRight:
responsiveWidth(4)}}><Image source={NotesImg} style={{height: responsiveWidth(6), width:
responsiveWidth(6), resizeMode: "contain",}} /><Text style={{color: "white", fontFamily:
"MPLUS1p", fontSize: responsiveFontSize(2)}}>Notes</Text></TouchableOpacity>
```

```
</View>
```

```
<View style={{
```

```
borderBottomColor: '#525468',
```

```
borderBottomWidth: 1,
```

```
width: responsiveHeight(25),
```

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

```
marginTop: responsiveHeight(2),
```

```
}}></View>
```

```

<View style={{gap: responsiveWidth(4), marginTop:responsiveWidth(7),
marginBottom: responsiveWidth(4)}}>

    <TouchableOpacity          onPress={handleLogout}          style={{marginLeft:
responsiveWidth(4), flexDirection: 'row', overflow: 'hidden', alignItems: 'center', gap:
responsiveWidth(4)}}>

        <Image source={Logout} style={{height: responsiveWidth(5), width:
responsiveWidth(5), resizeMode: "contain",}} />

        <Text style={{color: 'white', fontSize: responsiveFontSize(2), fontFamily:
"MPLUS1p"}}>Logout</Text>

    </TouchableOpacity>

    <View style={{marginLeft: responsiveWidth(4), flexDirection: 'row', overflow:
'hidden', alignItems: 'center', gap: responsiveWidth(4)}}>

        <TouchableOpacity          onPress={()                  =>
Linking.openURL('https://www.instagram.com/josiaslopo_/')}><Image source={Insta} style={{height:
responsiveWidth(5), width: responsiveWidth(5), resizeMode: "contain",}}/></TouchableOpacity>

        <TouchableOpacity          onPress={()                  =>
Linking.openURL('https://github.com/JosiasLopo')}><Image          source={Git}          style={{height:
responsiveWidth(5), width: responsiveWidth(5), resizeMode: "contain",}}/></TouchableOpacity>

        <TouchableOpacity          onPress={()                  =>
Linking.openURL('https://www.josias.pt')}><Image source={Link} style={{height: responsiveWidth(5),
width: responsiveWidth(5), resizeMode: "contain",}}/></TouchableOpacity>

    </View>

```

```

        <TouchableOpacity onPress={() => Linking.openURL('https://www.josias.pt')}
style={{alignSelf: 'center',marginTop: responsiveWidth(5),flexDirection: 'row', width:
responsiveWidth(40),alignItems: 'center',justifyContent: 'center', gap: responsiveWidth(4),
backgroundColor: '#101014', padding: responsiveWidth(3.5), borderRadius: 100,}}>

        <Image source={Web} style={{height: responsiveWidth(5), width:
responsiveWidth(5), resizeMode: "contain",}} />

        <Text style={{color: 'white', fontSize: responsiveFontSize(1.8), fontFamily:
"MPLUS1p"}}>Josias.pt</Text>

    </TouchableOpacity>

</View>

</View>

</View>

</View>

</View>

</DrawerContentScrollView>

</View>

)

}

```

```
const Stack = createNativeStackNavigator();
```

```
const StackNav={()=>{

  const Stack = createNativeStackNavigator();

  return (

    <Stack.Navigator initialRouteName='Home' screenOptions={{ route, navigation }} => ({
headerShown: false })>

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

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

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

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

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

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

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

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

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

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

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

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

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

    </Stack.Navigator>

  )

}
```

```
const DrawerNav={()=>{

  const Drawer = createDrawerNavigator();

  return (

    <Drawer.Navigator

      drawerContent={({props}) => <DrawerContent {...props} />}

      screenOptions={{

        drawerLabel: 'Home',

        headerShown: false,

        overlayColor: 'rgba(0, 0, 0, 0.5)',

        drawerStyle: {

          backgroundColor: '#1D1E26',

        },

        drawerItemStyle: {

          backgroundColor: 'transparent', // Remove background color

          height: 60, // Adjust item height if needed

        },

      }}

    >

    <Drawer.Screen name="Home" component={StackNavigator} />

    </Drawer.Navigator>

  )

}
```

```
export default function App() {  
  
  const { user } = useAuth();  
  
  return (  
  
    <NavigationContainer>  
  
      <StatusBar style="light" />  
  
      {user ? (  
  
        <DrawerNav />  
  
      ) : (  
  
        <Stack.Navigator initialRouteName='Hall' screenOptions={{ route, navigation }} => ({  
headerShown: false })}>  
  
          <Stack.Screen name="Regist" component={Regist} />  
  
          <Stack.Screen name="Hall" component={Hall} />  
  
          <Stack.Screen name="Login" component={Login} />  
  
          <Stack.Screen name="Home" component={Home} />  
  
        </Stack.Navigator>  
  
      )}  
  
    </NavigationContainer>  
  
  );  
  
}
```

3.1.2 Firebase.js (api do firebase censurada)

// Import the functions you need from the SDKs you need

```
import { initializeApp } from "firebase/app";
```

```
import { getAnalytics } from "firebase/analytics";
```

```
import { getFirestore } from 'firebase/firestore';
```

```
import { getAuth } from 'firebase/auth';
```

```
import { getStorage, ref } from 'firebase/storage';
```

Josias Lopo 41

```
const firebaseConfig = {  
  apiKey: "*****",  
  authDomain: "eventer-5e73c.firebaseio.com",  
  projectId: "eventer-5e73c",  
  storageBucket: "eventer-5e73c.appspot.com",  
  messagingSenderId: "*****",  
  appId: "*****"  
};
```

```
const app = initializeApp(firebaseConfig);
```

```
export const auth = getAuth(app);
```

```
export const FIRESTORE_DB = getFirestore(app);
```

```
export const storage = getStorage(app);
```

3.1.3 useAuth.js

```
import { View, Text } from 'react-native'
```

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

42 Josias Lopo

```
import { onAuthStateChanged } from 'firebase/auth';

import { auth } from '../config/firebase';

export default function useAuth() {

  const [user, setUser] = useState(null);

  useEffect(()=>{

    const unsub= onAuthStateChanged(auth, user =>{

      console.log('got user: ',user);

      if(user){

        setUser(user);

      } else {

        setUser(null);

      }

    });

    return unsub;

  }, [])

  return {user}

}

//rncf
```

3.1.4 ChatBubble.js

```
import React from "react";

import {View, Text, StyleSheet, TouchableOpacity} from "react-native";

import ChatGpt from "../Gemini";

import { responsiveWidth } from "react-native-responsive-dimensions";

const ChatBubble = ({ role, text }) => {

  return(

    <View

      style={[

        styles.chatItem,

        role === "user" ? styles.userChatItem : styles.modelChatItem,

      ]}

    >
```

```
<Text style={styles.chatText}>{text}</Text>

</View>

);

};

const styles = StyleSheet.create({

  chatItem: {

    marginBottom: responsiveWidth(3),

    padding: responsiveWidth(4),

    borderRadius: responsiveWidth(3),

    maxWidth: "80%",

    position: "relative",

  },

  userChatItem: {

    alignSelf: "flex-end",

    backgroundColor: "#007AFF",

  },

  modelChatItem: {

    alignSelf: "flex-start",

    backgroundColor: "#000",

  },

},
```

```
chatText: {  
  color: "#fff",  
  fontSize: 16,  
},  
});  
  
export default ChatBubble;
```

3.1.5 ChatGpt.js (Api Censurada)

```
import React, { useState, useEffect } from "react";  
  
import {  
  Text,  
  TextInput,
```

46 Josias Lopo

```
View,
FlatList,
StyleSheet,
TouchableOpacity,
ActivityIndicator,
Image,
} from "react-native";

import axios from "axios";

import AsyncStorage from "@react-native-async-storage/async-storage";
import ChatBubble from "../ChatBubble";

import {
  responsiveFontSize,
  responsiveHeight,
  responsiveWidth,
} from "react-native-responsive-dimensions";

import GptIcon from "../src/assets/images/GptIcon.png";
import Seta2 from "../src/assets/images/arrowLeft.png";
import SendIcon from "../src/assets/images/SendIcon.png";
import { useNavigation } from "@react-navigation/native";

export default function ChatGpt() {
  const navigation = useNavigation();
```

```
const [messages, setMessages] = useState([]);

const [inputText, setInputText] = useState("");

useEffect(() => {

  // Load messages from AsyncStorage when the component mounts

  const loadMessages = async () => {

    try {

      const storedMessages = await AsyncStorage.getItem("messages");

      if (storedMessages) setMessages(JSON.parse(storedMessages));

    } catch (error) {

      console.error("Failed to load messages:", error);

    }

  };

  loadMessages();

}, []);

const sendMessage = async () => {

  const userMessage = { role: "user", content: inputText };

  setInputText("");
```

```
try {  
  
  const response = await axios.post(  
  
    "https://api.openai.com/v1/chat/completions",  
  
    {  
  
      model: "gpt-3.5-turbo",  
  
      messages: [userMessage],  
  
      temperature: 0.5,  
  
    },  
  
    {  
  
      headers: {  
  
        "Content-Type": "application/json",  
  
        Authorization:  
  
          "Bearer *****",  
  
      },  
  
    }  
  
  );  
  
  const botMessage = {  
  
    role: "bot",  
  
    content: response.data.choices[0].message.content,  
  
  };  
  
  const updatedMessages = [...messages, userMessage, botMessage];
```

```
setMessages(updatedMessages);

saveMessages(updatedMessages);

} catch (error) {

  console.error("Error sending message:", error);

}

};

const saveMessages = async (messages) => {

  try {

    await AsyncStorage.setItem("messages", JSON.stringify(messages));

  } catch (error) {

    console.error("Failed to save messages:", error);

  }

};

const renderChatItem = ({ item }) => (

  <ChatBubble role={item.role} text={item.content} />

);

return (

  <View style={styles.container}>

    <View style={styles.header}>
```

```
<TouchableOpacity style={styles.btnArrow} onPress={() => navigation.navigate("Home")}>

  <Image source={Seta2} style={styles.btnArrowImg} />

</TouchableOpacity>

<View

  style={{

    marginRight: responsiveWidth(30),

    marginBottom: responsiveWidth(2),

  }}

>

  <Text style={styles.title}>Gpt 3.5</Text>

  <Text style={styles.titleDescription}>Online</Text>

</View>

</View>

<View

  style={{

    marginTop: responsiveWidth(-20),

    alignItems: "flex-end",

    width: "100%",

    height: responsiveHeight(9),

    marginLeft: responsiveWidth(5),

    zIndex: 1,

  }}

```

>

```
<TouchableOpacity style={{  
  backgroundColor: "#101014",  
  width: responsiveWidth(5.5),  
  height: responsiveWidth(5.5),  
  borderRadius: 100,  
  justifyContent: "center",  
  alignItems: "center",  
  marginRight: responsiveWidth(18),  
}}>
```

```
<Text  
  style={{  
    color: "white",  
    fontSize: responsiveFontSize(1.5),  
    fontFamily: "MPLUS1p",  
  }}>?</Text>
```

```
</TouchableOpacity>
```

```
<Image  
  source={GptIcon}  
  style={{  
    height: responsiveWidth(23),  
    width: responsiveWidth(23),
```

```
        resizeMode: "contain",

      }}

    />

  </View>

  <FlatList

    data={messages}

    renderItem={renderChatItem}

    keyExtractor={({item, index}) => index.toString()}

    contentContainerStyle={styles.chatContainer}

  />

  <View style={styles.inputContainer}>

    <TouchableOpacity style={styles.button} onPress={sendMessage}>

      <Image source={SendIcon} style={styles.buttonText} />

    </TouchableOpacity>

    <TextInput

      style={styles.input}

      placeholder="Type a message..."

      placeholderTextColor="#818181"

      value={inputText}

      onChangeText={setInputText}

    />

  </View>
```

```
</View>

</View>

);

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    padding: responsiveWidth(6),

    backgroundColor: "#101014",

    width: "100%",

    alignItems: "center",

  },

  header: {

    flexDirection: "row",

    paddingBottom: responsiveWidth(2),

    paddingRight: responsiveWidth(4),

    backgroundColor: "#1D1E26",

    marginTop: responsiveWidth(-5.5),

    height: responsiveHeight(15.5),

    width: responsiveWidth(100),
```

54 Josias Lopo

```
alignItems: "flex-end",

justifyContent: "space-between",

zIndex: 1,

},

btnArrow: {

  justifyContent: "center",

  alignItems: "center",

  backgroundColor: "rgba(29, 30, 38, 0.76)",

  marginLeft: responsiveWidth(7),

  height: responsiveWidth(15),

  resizeMode: "contain",

  //borderRadius: '100%',

},

btnArrowImg: {

  width: responsiveWidth(5),

  resizeMode: "contain",

},

title: {

  fontFamily: "MPLUS1p",
```

```
fontSize: responsiveFontSize(3),  
  
color: "white",  
  
textAlign: "center",  
  
},
```

```
titleDescription: {  
  
  fontFamily: "NanumMyeongjo",  
  
  fontSize: responsiveFontSize(1.5),  
  
  color: "#9F9F9F",  
  
  textAlign: "right",  
  
},
```

```
chatContainer: {  
  
  flexGrow: 1,  
  
  justifyContent: "flex-end",  
  
  width: responsiveWidth(86),  
  
  paddingTop: responsiveWidth(14),  
  
},
```

```
inputContainer: {  
  
  flexDirection: "row",  
  
  alignItems: "center",  
  
  justifyContent: "center",
```

```
marginTop: 10,  
  
height: responsiveHeight(7),  
  
marginRight: 10,  
  
padding: responsiveWidth(1),  
  
borderRadius: 100,  
  
backgroundColor: "#1D1E26",  
  
gap: responsiveWidth(3),  
  
},
```

```
input: {  
  
  flex: 4,  
  
  color: "#818181",  
  
},
```

```
button: {  
  
  flex: 1,  
  
  justifyContent: "center",  
  
  alignItems: "center",  
  
},
```

```
buttonText: {  
  
  width: responsiveWidth(8),
```

```
height: responsiveWidth(8),  
  
resizeMode: "contain",  
  
},  
  
});
```

3.1.6 ChatPage.js

```
import React, { useState, useEffect } from 'react';  
  
import { View, TextInput, FlatList, StyleSheet, TouchableOpacity, Text, Image } from 'react-native';  
  
import { getFirestore, collection, addDoc, query, onSnapshot, orderBy } from "firebase/firestore";  
  
import { getAuth } from "firebase/auth";  
  
import {  
  
  responsiveFontSize,  
  
  responsiveHeight,  
  
  responsiveWidth,  
  
} from "react-native-responsive-dimensions";  
  
import ChatIcon from "../src/assets/images/ChatIcon.png";  
  
import Seta2 from "../src/assets/images/arrowLeft.png";  
  
import SendIcon from "../src/assets/images/SendIcon.png";  
  
import { useNavigation } from "@react-navigation/native";
```

```
const db = getFirestore();
```

```
const MyChatScreen = () => {
```

```
  const navigation = useNavigation();
```

```
  const [messages, setMessages] = useState([]);
```

```
  const [newMessage, setNewMessage] = useState("");
```

```
  useEffect(() => {
```

```
    const auth = getAuth();
```

```
    const userId = auth.currentUser.uid;
```

```
    const messagesRef = collection(db, 'users', userId, 'messages');
```

```
    const q = query(messagesRef, orderBy('timestamp', 'asc'));
```

```
    const unsubscribe = onSnapshot(q, (snapshot) => {
```

```
      setMessages(snapshot.docs.map(doc => ({ id: doc.id, ...doc.data() })));
```

```
    });
```

```
    return () => unsubscribe();
```

```

    }, []);

const sendMessage = async () => {

    if (newMessage.length === 0) return;

    try {

        const auth = getAuth();

        const userId = auth.currentUser.uid;

        await addDoc(collection(db, 'users', userId, 'messages'), {

            text: newMessage,

            from: 'me',

            timestamp: Date.now(),

        });

        setNewMessage("");

    } catch (error) {

        console.error('Error sending message: ', error);

    }

};

return (

    <View style={styles.container}>

```

```
<View style={styles.header}>

  <TouchableOpacity style={styles.btnArrow} onPress={() => navigation.navigate("Home")}>

    <Image source={Seta2} style={styles.btnArrowImg} />

  </TouchableOpacity>

</View>

style={{

  marginRight: responsiveWidth(30),

  marginBottom: responsiveWidth(2),

}}

>

<Text style={styles.title}>Notes</Text>

<Text style={styles.titleDescription}>Save your stuff</Text>

</View>

</View>

<View

style={{

  marginTop: responsiveWidth(-20),

  alignItems: "flex-end",

  width: "100%",

  height: responsiveHeight(9),
```

```

marginLeft: responsiveWidth(5),

zIndex: 1,

}}

>

<TouchableOpacity style={{

  backgroundColor: "#101014",

  width: responsiveWidth(5.5),

  height: responsiveWidth(5.5),

  borderRadius: 100,

  justifyContent: "center",

  alignItems: "center",

  marginRight: responsiveWidth(18),

  }}>

<Text

  style={{

    color: "white",

    fontSize: responsiveFontSize(1.5),

    fontFamily: "MPLUS1p",

    }}>?</Text>

</TouchableOpacity>

<Image

  source={ChatIcon}

```

```

    style={{
      height: responsiveWidth(23),
      width: responsiveWidth(23),
      resizeMode: "contain",
    }}
  />
</View>

<FlatList
  data={messages}
  keyExtractor={item => item.id}
  contentContainerStyle={styles.chatContainer}
  renderItem={({ item }) => (
    <View style={[styles.message, item.from === 'me' ? styles.myMessage :
styles.otherMessage]}>
      <Text style={{alignSelf: 'flex-end', color: "white", fontFamily: 'MPLUS1p', fontSize:
responsiveFontSize(2)}}>{item.text}</Text>
      <Text style={styles.timestamp}>
        {new Date(item.timestamp).toLocaleDateString() + ' ' + new
Date(item.timestamp).toLocaleTimeString()}
      </Text>
    </View>
  )}
/>
</View>

```

```

    })

  />

<View style={styles.inputContainer}>

  <TouchableOpacity style={styles.button} onPress={sendMessage}>

    <Image source={SendIcon} style={styles.buttonText} />

  </TouchableOpacity>

  <TextInput

    style={styles.input}

    placeholder="Type a fast note..."

    placeholderTextColor="#818181"

    value={newMessage}

    onChangeText={setNewMessage}

  />

</View>

</View>

);

};

```

```

const styles = StyleSheet.create({

  container: {

    flex: 1,

```

```
padding: responsiveWidth(6),

backgroundColor: "#101014",

width: "100%",

alignItems: "center",

},

message: {

  backgroundColor: 'white',

  padding: 10,

  borderRadius: 8,

  marginVertical: 5,

  maxWidth: '80%',

},

myMessage: {

  alignSelf: 'flex-end',

  backgroundColor: '#1563FF',

},

otherMessage: {

  alignSelf: 'flex-start',

  backgroundColor: '#e8e8e8'

},

inputContainer: {

  flexDirection: 'row',
```

```
alignItems: 'center',

backgroundColor: 'white',

padding: 10,

borderTopWidth: 1,

borderTopColor: '#e8e8e8',

},

input: {

  flex: 1,

  borderWidth: 1,

  borderColor: '#ccc',

  borderRadius: 5,

  padding: 8,

  marginRight: 10

},

sendButton: {

  color: '#007bff',

  fontSize: 16,

},

timestamp: {

  fontSize: responsiveFontSize(1.3),

  fontStyle: 'italic',
```

```
color: 'black',

marginTop: 5,

alignSelf: 'flex-end'

},

header: {

  flexDirection: "row",

  paddingBottom: responsiveWidth(2),

  paddingRight: responsiveWidth(4),

  backgroundColor: "#1D1E26",

  marginTop: responsiveWidth(-5.5),

  height: responsiveHeight(15.5),

  width: responsiveWidth(100),

  alignItems: "flex-end",

  justifyContent: "space-between",

  zIndex: 1,

},

btnArrow: {

  justifyContent: "center",

  alignItems: "center",

  backgroundColor: "rgba(29, 30, 38, 0.76)",
```

```
marginLeft: responsiveWidth(7),  
  
height: responsiveWidth(15),  
  
resizeMode: "contain",  
  
//borderRadius: '100%',  
  
},
```

```
btnArrowImg: {  
  
  width: responsiveWidth(5),  
  
  resizeMode: "contain",  
  
},
```

```
title: {  
  
  fontFamily: "MPLUS1p",  
  
  fontSize: responsiveFontSize(3),  
  
  color: "white",  
  
  textAlign: "center",  
  
  alignSelf: 'flex-end'  
  
},
```

```
titleDescription: {  
  
  fontFamily: "NanumMyeongjo",  
  
  fontSize: responsiveFontSize(1.5),
```

```
color: "#9F9F9F",  
  
textAlign: "right",  
  
alignSelf: 'flex-end'  
  
},  
  
chatContainer: {  
  
  flexGrow: 1,  
  
  justifyContent: "flex-end",  
  
  width: responsiveWidth(86),  
  
  paddingTop: responsiveWidth(14),  
  
},  
  
inputContainer: {  
  
  flexDirection: "row",  
  
  alignItems: "center",  
  
  justifyContent: "center",  
  
  marginTop: 10,  
  
  height: responsiveHeight(7),  
  
  marginRight: 10,  
  
  padding: responsiveWidth(1),  
  
  borderRadius: 100,  
  
  backgroundColor: "#1D1E26",  
  
  gap: responsiveWidth(3),
```

},

input: {

flex: 4,

color: "#818181",

},

button: {

flex: 1,

justifyContent: "center",

alignItems: "center",

},

buttonText: {

width: responsiveWidth(8),

height: responsiveWidth(8),

resizeMode: "contain",

},

});

export default MyChatScreen;

70 Josias Lopo

3.1.7 ChatsScreen.js

```
// App.js

import React from 'react';

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

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

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

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

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

import ChatGptIcon from '../src/assets/images/GptIcon.png';

import Gemini from '../src/assets/images/GeminIcon.png';

import Chat from '../src/assets/images/ChatIcon.png';

import Arrow from '../src/assets/images/ArrowNoTail.png';

import headerImg from '../src/assets/images/AI_Stars.png';
```

```
export default function ChatsScreen() {  
  
  const navigation = useNavigation();  
  
  const insets = useSafeAreaInsets();  
  
  const showAlert = () => {  
  
    Alert.alert(  
  
      'Alert Title',  
  
      'My Alert Message',  
  
      [  
  
        { text: 'OK', onPress: () => console.log('OK Pressed') }  
  
      ],  
  
      { cancelable: false }  
  
    );  
  
  };  
  
  return (  
  
    <View style={ [styles.container, {paddingTop: insets.top}] }>  
  
      <View style={{flex: 1, flexDirection: 'row'}}>  
  
        <View style={styles.header}>  
  
          <View style={styles.TxtView}>  
  
            <Text style={styles.headerTxt}> AI </Text>  
  
            <Text style={styles.headerTxtDot}>Ask and Add</Text>  
  
          </View>  
  
        </View>  
  
      </View>  
  
    );  
  
  );  
}
```

```
</View>
```

```
<Text style={styles.headerTxt2}> ZONE</Text>
```

```
</View>
```

```
<View style={
```

```
{flex: 1,
```

```
alignItems: 'flex-start',
```

```
justifyContent: 'center',
```

```
backgroundColor: '#101014',
```

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



```
<Image source={headerImg} style={styles.headerImg}/>
```

```
</View>
```

```
</View>
```

```
<View style={{flex: 3.5, width: '100%', flexDirection: 'row'}}>
```

```
<View style={{height: '100%', backgroundColor: '#1D1E26', flex: 1, alignItems: 'center',  
justifyContent: 'center', marginBottom: responsiveWidth(10), paddingBottom: insets.bottom}}>
```

```
<TouchableOpacity style={styles.backBtn} onPress={() => navigation.navigate('Home')}>
```

```
<Text onPress={showAlert} style={{flex: 1, color: 'white', width: '100%',fontSize:  
responsiveFontSize(1), transform: [{ rotate: '-90deg'}], textAlign: 'center'}}>Easter Egg</Text>
```

```

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

</TouchableOpacity>

</View>

<View style={{[styles.chatContainer, {paddingBottom: insets.bottom}]}}>

  <TouchableOpacity style={styles.chatView} onPress={() => navigation.navigate('ChatGpt')}>

    <View style={{marginLeft: responsiveWidth(5)}}>

      <Text style={styles.AiTitle}>Gpt 3.5 API</Text>

      <Text style={styles.AiDescription}>Most Optimized</Text>

    </View>

    <View style={{[styles.logolImage, {marginRight: responsiveWidth(5)}}}>

      <Image source={ChatGptIcon} style={styles.menuBtnImg}/>

    </View>

  </TouchableOpacity>

  <TouchableOpacity style={styles.chatView} onPress={() => navigation.navigate('Gemini')}>

    <View style={{marginLeft: responsiveWidth(5)}}>

      <Text style={styles.AiTitle}>Gemini</Text>

      <Text style={styles.AiDescription}>Regional And Longer Responses</Text>

    </View>

    <View style={{[styles.logolImage, {marginRight: responsiveWidth(5)}}}>

```

```

    <Image source={Gemini} style={styles.menuBtnImg}/>

  </View>

</TouchableOpacity>

<TouchableOpacity style={styles.chatView} onPress={() => navigation.navigate('ChatPage')}>

  <View style={{marginLeft: responsiveWidth(5)}}>

    <Text style={styles.AiTitle}>Fast Notes</Text>

    <Text style={styles.AiDescription}>Chat Yourself</Text>

  </View>

  <View style={[styles.logolImage, {marginRight: responsiveWidth(5)}}>

    <Image source={Chat} style={styles.menuBtnImg}/>

  </View>

</TouchableOpacity>

</View>

</View>

</View>

);

};

```

```
const styles = StyleSheet.create({  
  
  container: {  
  
    flex: 1,  
  
    alignItems: 'center',  
  
    justifyContent: 'center',  
  
    backgroundColor: "#101014",  
  
  },  
  
  header:{  
  
    flex: 1,  
  
    alignItems: 'flex-start',  
  
    justifyContent: 'center',  
  
    backgroundColor: '#101014',  
  
    width: '100%',  
  
  },  
  
  headerImg:{  
  
    flex: 1,  
  
    width: responsiveWidth(40),  
  
    resizeMode: "contain",  
  
    marginLeft: responsiveWidth(7),
```

```
},  
  
TxtView:{  
  
  flexDirection: "row",  
  
  alignItems: 'baseline',  
  
},  
  
headerTxt:{  
  
  color:"white",  
  
  fontSize: responsiveFontSize(5),  
  
  fontFamily: "NanumMyeongjo",  
  
},  
  
headerTxtDot:{  
  
  color:"#818181",  
  
  fontSize: responsiveFontSize(1.4),  
  
  fontFamily: "MPLUS1p",  
  
},  
  
  
  
headerTxt2:{  
  
  color:"white",  
  
  fontSize: responsiveFontSize(5),  
  
  fontFamily: "NanumMyeongjo",  
  
},  
  
chatContainer: {
```

```
flex: 11,  
  
width: '100%',  
  
backgroundColor: '#1D1E26',  
  
alignItems: 'center',  
  
justifyContent: 'center'  
  
},  
  
chatView:{  
  
  flexDirection: 'row',  
  
  width: responsiveHeight(40),  
  
  height: responsiveHeight(10),  
  
  backgroundColor: '#101014',  
  
  justifyContent: 'space-between',  
  
  alignItems: 'center',  
  
  borderTopLeftRadius: responsiveWidth(14),  
  
  borderTopRightRadius: responsiveWidth(6.5),  
  
  borderBottomLeftRadius: responsiveWidth(6.5),  
  
  borderBottomRightRadius: responsiveWidth(14),  
  
  marginBottom: responsiveWidth(4),  
  
  marginRight: responsiveWidth(3),  
  
},
```

```
logolImage:{  
  
  height: responsiveHeight(10),  
  
  justifyContent: 'center',  
  
},  
  
menuBtnImg:{  
  
  height: responsiveWidth(17),  
  
  width: responsiveWidth(17),  
  
  resizeMode: "contain",  
  
},  
  
backBtn: {  
  
  flexDirection: 'row',  
  
  height: responsiveHeight(17),  
  
  width: responsiveWidth(17),  
  
  backgroundColor: '#101014',  
  
  borderTopRightRadius: 100,  
  
  borderBottomRightRadius: 100,  
  
  overflow: 'hidden',  
  
  marginRight: responsiveWidth(14),  
  
  marginBottom: responsiveWidth(12),  
  
  justifyContent: 'center',
```

```
alignItems: 'center',  
},
```

```
Arrow:{  
  
  flex: 1,  
  
  height: responsiveWidth(4),  
  
  resizeMode: "contain",  
  
  marginRight: responsiveWidth(-1),  
},
```

```
AiTitle: {  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(2.3),  
  
  fontFamily: "MPLUS1p",  
  
  marginLeft: responsiveWidth(5),  
},
```

```
AiDescription: {  
  
  color: '#818181',  
  
  fontSize: responsiveFontSize(1.5),  
  
  fontFamily: "MPLUS1p",  
  
  marginLeft: responsiveWidth(5),
```

80 Josias Lopo

```
width: responsiveWidth(40),  
  
},  
  
});
```

3.1.8 CreateNote.js

```
import React, { useState } from 'react';  
import { View, Text, TextInput, StyleSheet, Button, Alert, TouchableOpacity, Image } from 'react-native';  
import { collection, addDoc } from 'firebase/firestore';  
import { FIRESTORE_DB, auth } from '../config/firebase';  
import { useNavigation } from '@react-navigation/native';  
import { responsiveFontSize, responsiveHeight, responsiveScreenFontSize, responsiveWidth } from  
"react-native-responsive-dimensions";  
import { useSafeAreaInsets } from 'react-native-safe-area-context';  
import Seta2 from "../src/assets/images/arrowLeft.png";  
  
export default function CriarNota({ navigation }) {  
  const [titulo, setTitulo] = useState("");  
  const [conteudo, setConteudo] = useState("");  
  const insets = useSafeAreaInsets();
```

```
const [isPlaceholderTitle, setIsPlaceholderTitle] = useState(true);
const [isPlaceholder, setIsPlaceholder] = useState(true);

const handleGoBack = () => {
  navigation.goBack();
};

const criarNota = async () => {
  try {
    await addDoc(collection(FIRESTORE_DB, 'notas'), {
      titulo,
      conteudo,
      userId: auth.currentUser.uid,
    });
    navigation.goBack();
  } catch (error) {
    console.error("Error creating a note:", error);
  }
};

const handleTitleChange = (title) => {
  setTitulo(title);
  if (title === "") {
    setIsPlaceholderTitle(true);
  } else {
    setIsPlaceholderTitle(false);
  }
};

const handleTextChange = (text) => {
  setConteudo(text);
  if (text === "") {
    setIsPlaceholder(true);
  } else {
    setIsPlaceholder(false);
  }
};

return (
  <View style={[styles.container, {paddingTop: insets.top}]}>
    <View style={[styles.notesContainer, {paddingBottom: insets.bottom - 10}]}>
```

```

<View style={styles.title}>
  <TouchableOpacity style={styles.btnArrow} onPress={handleGoBack} >
    <Image source={Seta2} style={styles.btnArrowImg} />
  </TouchableOpacity>
  <View style={styles.inputContainer}>
    <TextInput
      style={styles.input}
      placeholder="Title"
      placeholderTextColor="#818181"
      value={titulo}
      onChangeText={(title) => {
        setTitulo(title);
        handleTitleChange(title);
      }}
      selectionColor="white"

    />
  </View>
</View>

  <TextInput
    style={styles.inputArea}
    multiline={true}
    placeholder="Write your note..."
    placeholderTextColor="#818181"
    value={conteudo}
    onChangeText={(text) => {
      setConteudo(text);
      handleTextChange(text);
    }}
    selectionColor="white"
  />

  <View style={styles.deleteBtn}>
    <TouchableOpacity onPress={criarNota}>
      <Text style={styles.deleteBtnText}>Save Note</Text>
    </TouchableOpacity>
  </View>
</View>
</View>
);
}

```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#1D1E26',
  },

  notesContainer: {
    flex: 1,
    backgroundColor: '#101014',
    marginTop: responsiveWidth(10),
    borderTopRightRadius: responsiveWidth(10),
    borderTopLeftRadius: responsiveWidth(10),
  },

  title: {
    flex: 1,
    flexDirection: 'row',
    marginBottom: responsiveWidth(10),
    borderBottomWidth: 1,
    borderBottomColor: '#ccc',
    alignItems: 'center',
    //backgroundColor: 'tomato'
  },

  btnArrow: {
    marginTop: responsiveWidth(3),
    justifyContent: "center",
    alignItems: "center",
    width: responsiveWidth(10),
    height: responsiveWidth(10),
    resizeMode: "contain",
    marginLeft: responsiveWidth(5)
  },

  btnArrowImg: {
    width: responsiveWidth(5),
    resizeMode: "contain",
  },

  inputContainer: {
```

```

    flex: 1,
    alignItems: 'center',
  },

  input: {
    fontSize: responsiveFontSize(5),
    fontFamily: 'NanumMyeongjo',
    color: 'white',
    marginTop: responsiveWidth(3),
    paddingRight: responsiveWidth(18)
  },

  inputArea: {
    flex: 5,
    fontSize: 18,
    color: 'white',
    paddingLeft: responsiveWidth(10),
    paddingRight: responsiveWidth(10)
    //textAlignVertical: 'top', // Para Android
  },

  deleteBtn: {
    fontSize: responsiveFontSize(2),
    padding: 10,
    borderRadius: 100,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#1D1E26',
    width: responsiveWidth(30),
    alignSelf: 'center',
    marginBottom: responsiveWidth(2)
  },
  deleteBtnText: {
    color: '#1563FF',
    fontSize: responsiveFontSize(1.8),
  }
});

```

3.1.9 Default.js

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

import { View, StyleSheet, Image, Text, TouchableOpacity, ScrollView, Alert, TextInput, Platform }
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 { getFirestore, collection, addDoc } from 'firebase/firestore';

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

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

import Seta2 from "../src/assets/images/arrowLeft.png";

import camera from "../src/assets/images/camera.png";

import account from "../src/assets/images/account.png";

import { auth } from '../config/firebase';

import ProfileScreen from './profilePic2';

const Profile = () => {

  const insets = useSafeAreaInsets();

  const db = getFirestore(app);
```

```
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 };

  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 = "profilePic";  
  
  const storageRef = ref(storage, `profilePic/${uid}/${filename}`);  
  
  try {  
  
    uploadBytes(storageRef, blob);  
  
  } catch (e) {  
  
    console.log("Upload Error:", e);  
  
  }  
  
  setUploading(false);  
  
  Alert.alert(  
  
    'Photo uploaded!'
```

```
);  
  
setImage(null);  
  
};  
  
const [userEmail, setUserEmail] = useState("");  
  
useEffect(() => {  
  
  const unsubscribe = auth.onAuthStateChanged((user) => {  
  
    if (user) {  
  
      setUserEmail(user.email);  
  
    } else {  
  
      setUserEmail("");  
  
    }  
  
  });  
  
  return () => unsubscribe();  
  
}, []);  
  
return (  
  
  <View style={[styles.container, { paddingTop: insets.top, paddingBottom: insets.bottom}]}>
```

```
<View style={styles.header}>

<TouchableOpacity style={styles.btnArrow} onPress={() => navigation.navigate("Home")}>

  <Image source={Seta2} style={styles.btnArrowImg} />

</TouchableOpacity>

{image && (

  <TouchableOpacity onPress={uploadImage}>

    <Text style={styles.updateBtn}>Update</Text>

  </TouchableOpacity>

)}

</View>

<View style={styles.profilePic}>

  <View style={styles.imageContainer}>

    <View style={styles.imageContainer2}>

      <View style={styles.onChangeImage}>

        {image && <Image source={{ uri: image.uri }} style={{ width: "100%", height: "100%" }} />}

      </View>

    <View style={styles.background}>

      <ProfileScreen/>

    </View>

  </View>

</View>
```

```

    </View>

    </View>

</View>

<View style={styles.cameraImg}>

    <TouchableOpacity onPress={pickImage}>

        <Image source={camera} style={styles.cameraImg2} />

    </TouchableOpacity>

</View>

</View>

<View style={styles.content}>

    <Text style={styles.emailTxt}>E-mail:</Text>

    <Text style={styles.userTxt}>{userEmail}</Text>

</View>

</View>

);

};

export const styles = StyleSheet.create({

```

```
container: {  
  
  flex: 1,  
  
  backgroundColor: '#1D1E26',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},  
  
header: {  
  
  flex: 1,  
  
  flexDirection: 'row',  
  
  width: '80%',  
  
  justifyContent: "space-between",  
  
  alignItems: 'flex-start',  
  
  paddingTop: responsiveWidth(4)  
  
},
```

```
btnArrow: {  
  
  justifyContent: "center",  
  
  alignItems: "center",  
  
  backgroundColor: "#101014",  
  
  width: responsiveWidth(12),  
  
  height: responsiveWidth(12),
```

```
      resizeMode: "contain",

      borderRadius: 100,

      alignSelf: 'flex-start'
    },

    btnArrowImg: {

      width: responsiveWidth(5),

      resizeMode: "contain",

    },

    updateBtn: {

      color: '#1563FF',

      fontFamily: 'MPLUS1pBold',

      fontSize: responsiveFontSize(2),

      padding: 10,

      marginTop: responsiveWidth(1),

      borderRadius: 10,

      overflow: 'hidden'

    },

    profilePic: {

      justifyContent: 'center',
```

```
alignItems: 'center',  
  
marginTop: responsiveWidth(10),  
  
//backgroundColor: 'tomato',  
  
},
```

```
imageContainer: {  
  
  padding: 5,  
  
  width: responsiveWidth(62),  
  
  height: responsiveWidth(62),  
  
  borderRadius: 200,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  backgroundColor: '#525468',  
  
  overflow: 'hidden',  
  
  alignSelf: 'center',  
  
  zIndex: 1,  
  
},
```

```
imageContainer2: {  
  
  padding: 4,  
  
  width: responsiveWidth(60.5),
```

```
height: responsiveWidth(60.5),  
  
borderRadius: 200,  
  
justifyContent: 'center',  
  
alignItems: 'center',  
  
backgroundColor: '#33365A',  
  
overflow: 'hidden'  
  
},
```

```
onChangelImage: {  
  
width: '100%',  
  
height: '100%',  
  
borderRadius: 200,  
  
position: 'absolute',  
  
zIndex: 2,  
  
overflow: "hidden",  
  
},
```

```
background: {  
  
height: '100%',  
  
width: '100%',
```

```
resizeMode: "cover",  
  
overflow: "hidden",  
  
borderRadius: 200,  
  
position: 'absolute',  
  
zIndex: 1,  
  
},
```

```
cameraImg: {  
  
  width: responsiveWidth(10),  
  
  height: responsiveWidth(10),  
  
  zIndex: 2,  
  
  marginLeft: responsiveWidth(52),  
  
  position: 'relative',  
  
  bottom: responsiveWidth(9),  
  
  alignSelf: 'center',  
  
},
```

```
cameraImg2: {  
  
  width: responsiveWidth(10),  
  
  height: responsiveWidth(10),  
  
},
```

```
content: {  
  
  flex: 1.5,  
  
  justifyContent: 'flex-start',  
  
  alignItems: 'center',  
  
},  
  
emailTxt: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1pBold',  
  
  fontSize: responsiveFontSize(2),  
  
  overflow: 'hidden'  
  
},  
  
userTxt: {  
  
  color: '#818181',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(2),  
  
  overflow: 'hidden'  
  
}  
  
});
```

export default Profile;

3.1.10 Gemini.js (api censurada)

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

import {

  Text,

  TextInput,

  View,

  FlatList,

  StyleSheet,

  TouchableOpacity,

  ActivityIndicator,

  Image,

} from "react-native";

import axios from "axios";

import ChatBubble from "../ChatBubble";

import {

  responsiveFontSize,

  responsiveHeight,

  responsiveWidth,

} from "react-native-responsive-dimensions";
```

98 Josias Lopo

```
import Gemini1 from "../src/assets/images/GeminiIcon.png";

import Seta2 from "../src/assets/images/arrowLeft.png";

import SendIcon from "../src/assets/images/SendIcon.png";

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

export default function Gemini() {

  const navigation = useNavigation();

  const [chat, setChat] = useState([]);

  const [userInput, setUserInput] = useState("");

  const [loading, setLoading] = useState(false);

  const [error, setError] = useState(null);

  const [isSpeaking, setIsSpeaking] = useState(false);

  const API_KEY = "*****",

  const handleUserInput = async () => {

    let updatedChat = [

      ...chat,

      {

        role: "user",
```

```
    parts: [{ text: userInput }],  
  
  },  
  
];  
  
setLoading(true);  
  
try {  
  
  const response = await axios.post(  
  
    `https://generativelanguage.googleapis.com/v1beta/models/gemini-  
pro:generateContent?key=${API_KEY}`,  
  
    {  
  
      contents: updatedChat,  
  
    }  
  
  );  
  
  
  console.log("Gemini Pro API Response", response.data);  
  
  
  const modelResponse =  
  
    response.data?.candidates?.[0]?.content.parts?.[0]?.text || "";  
  
  
  if (modelResponse) {  
  
    const updatedChatWithModel = [
```

```

    ...updatedChat,

    {

      role: "model",

      parts: [{ text: modelResponse }],

    },

  ];

  setChat(updatedChatWithModel);

  setUserInput("");

}

} catch (error) {

  console.error("Error Calling Gemini Pro Api: ", error);

  console.error("Error Response: ", error.response);

  setError("An error occurred. Please try again.");

} finally {

  setLoading(false);

}

};

```

```

const renderChatItem = ({ item }) => (

  <ChatBubble

```

```
role={item.role}

text={item.parts[0].text}

/>

);

return (

<View style={styles.container}>

  <View style={styles.header}>

    <TouchableOpacity style={styles.btnArrow} onPress={() => navigation.navigate("Home")}>

      <Image source = {Seta2} style={styles.btnArrowImg}/>

    </TouchableOpacity>

  <View

    style={{

      marginRight: responsiveWidth(30),

      marginBottom: responsiveWidth(2),

    }}

  >

    <Text style={styles.title}>Gemini</Text>

    <Text style={styles.titleDescription}>Online</Text>

  </View>

</View>

<View

  <View
```

```

style={{
  marginTop: responsiveWidth(-20),
  alignItems: "flex-end",
  width: "100%",
  height: responsiveHeight(9),
  marginLeft: responsiveWidth(5),
  zIndex: 1,
}}
>
<TouchableOpacity style={{
  backgroundColor: "#101014",
  width: responsiveWidth(5.5),
  height: responsiveWidth(5.5),
  borderRadius: 100,
  justifyContent: "center",
  alignItems: "center",
  marginRight: responsiveWidth(18),
}}>
<Text
  style={{
    color: "white",
    fontSize: responsiveFontSize(1.5),

```

```
        fontFamily: "MPLUS1p",  
      }}>?</Text>  
  
</TouchableOpacity>  
  
<Image  
  source={Gemini1}  
  style={{  
    height: responsiveWidth(23),  
    width: responsiveWidth(23),  
    resizeMode: "contain",  
  }}  
</Image>  
  
</View>  
  
<FlatList  
  data={chat}  
  renderItem={renderChatItem}  
  keyExtractor={({item, index}) => index.toString()}  
  contentContainerStyle={styles.chatContainer}  
</FlatList>  
  
<View style={styles.inputContainer}>  
  <TouchableOpacity style={styles.button} onPress={handleUserInput}>  
    <Image source={SendIcon} style={styles.buttonText}/>  
  </TouchableOpacity>  
</View>
```

10 Josias Lopo

```

</TouchableOpacity>

<TextInput
  style={styles.input}
  placeholder="Type a message..."
  placeholderTextColor="#818181"
  value={userInput}
  onChangeText={setUserInput}
/>

</View>

{loading && <ActivityIndicator style={styles.loading} color="#333" />}

{error && <Text style={styles.error}>{error}</Text>}

</View>

);
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: responsiveWidth(6),
    backgroundColor: "#101014",
    width: "100%",
    alignItems: "center",

```

},

```
header: {  
  
  flexDirection: "row",  
  
  paddingBottom: responsiveWidth(2),  
  
  paddingRight: responsiveWidth(4),  
  
  backgroundColor: "#1D1E26",  
  
  marginTop: responsiveWidth(-5.5),  
  
  height: responsiveHeight(15.5),  
  
  width: responsiveWidth(100),  
  
  alignItems: "flex-end",  
  
  justifyContent: "space-between",  
  
  zIndex: 1,  
  
},
```

```
btnArrow: {  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
  marginLeft: responsiveWidth(7),  
  
  height: responsiveWidth(15),  
  
  resizeMode: "contain",
```

},

```
btnArrowImg: {  
  width: responsiveWidth(5),  
  resizeMode: "contain",  
},
```

```
title: {  
  fontFamily: "MPLUS1p",  
  fontSize: responsiveFontSize(3),  
  color: "white",  
  textAlign: "center",  
},
```

```
titleDescription: {  
  fontFamily: "NanumMyeongjo",  
  fontSize: responsiveFontSize(1.5),  
  color: "#9F9F9F",  
  textAlign: "right",  
},
```

```
chatContainer: {
```

```
flexGrow: 1,  
  
justifyContent: "flex-end",  
  
width: responsiveWidth(86),  
  
paddingTop: responsiveWidth(14),  
  
},  
  
inputContainer: {  
  
  flexDirection: "row",  
  
  alignItems: "center",  
  
  justifyContent: 'center',  
  
  marginTop: 10,  
  
  height: responsiveHeight(7),  
  
  marginRight: 10,  
  
  padding: responsiveWidth(1),  
  
  borderRadius: 100,  
  
  backgroundColor: "#1D1E26",  
  
  gap: responsiveWidth(3),  
  
},  
  
input: {  
  
  flex: 4,  
  
  color: '#818181',  
  
},
```

10 Josias Lopo

```
button: {  
  
  flex: 1,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},  
  
buttonText: {  
  
  width: responsiveWidth(8),  
  
  height: responsiveWidth(8),  
  
  resizeMode: "contain",  
  
},  
  
loading: {  
  
  marginTop: 10,  
  
},  
  
error: {  
  
  color: "red",  
  
  marginTop: 10,  
  
},  
  
});
```

3.1.11 Hall.js

```
import React from "react";

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

import { useFonts } from 'expo-font';

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-dimensions";

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

import Seta from "../src/assets/images/seta.png";

import GLogo from "../src/assets/images/googlelogo.png";

export default function Hall({ navigation }) {

  const insets = useSafeAreaInsets();

  const [loaded] = useFonts({

    NanumMyeongjo: require("../src/assets/fonts/NanumMyeongjo-Regular.ttf"),
```

```

NanumMyeongjoBold: require('../src/assets/fonts/NanumMyeongjo-Bold.ttf'),
NanumMyeongjoEBold: require('../src/assets/fonts/NanumMyeongjo-ExtraBold.ttf'),
Inter: require('../src/assets/fonts/Inter-Regular.ttf'),
InterBold: require('../src/assets/fonts/Inter-Bold.ttf'),
InterEBold: require('../src/assets/fonts/Inter-ExtraBold.ttf'),
MPLUS1p: require('../src/assets/fonts/MPLUS1p-Regular.ttf'),
MPLUS1pBold: require('../src/assets/fonts/MPLUS1p-Bold.ttf'),
MPLUS1pEBold: require('../src/assets/fonts/MPLUS1p-ExtraBold.ttf'),
});

if (!loaded){
  return null;
}

return (

<View style={{styles.container,{paddingTop: insets.top + responsiveWidth(5), paddingBottom:
insets.bottom}}}>

<View style={{styles.title, {paddingLeft: insets.left + responsiveWidth(4)}}>

<Text style={styles.txt1}>Improve your</Text>

```

```
</View>
```

```
<ImageBackground
```

```
  source={require('../src/assets/images/intro-bg.jpg')} style={[styles.bgImg,{paddingLeft: insets.left  
+ responsiveWidth(4)}}]>
```

```
  <Text style={styles.txt2}>Event</Text>
```

```
</ImageBackground>
```

```
<TouchableOpacity style={styles.containerOu} onPress={() => navigation.navigate("Login")}>
```

```
  <Text style={styles.txtOu}>Login</Text>
```

```
  <View style={styles.btnArrow}>
```

```
    <Image source = {Seta} style={styles.btnArrowImg}/>
```

```
  </View>
```

```
</TouchableOpacity>
```

```
<View style={styles.googleRegisBtn}>
```

```
  <TouchableOpacity style={styles.googleBtn} onPress={() => navigation.navigate("Home")}>
```

```
    <Image source = {GLogo} style={styles.googleBtnLogo}/>
```

```
    <Text style={styles.googleBtnTxt}>Continue with Google</Text>
```

```
  </TouchableOpacity>
```

```
<TouchableOpacity style={styles.regisBtn} onPress={() => navigation.navigate("Regist")}>
```

```
<Text style={styles.regisBtnTxt}>Register</Text>

</TouchableOpacity>

</View>
```

```
</View>
```

```
);
```

```
};
```

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

```
  container: {
```

```
    flex: 1,
```

```
    backgroundColor: '#101014',
```

```
  },
```

```
  title: {
```

```
    flex: 1,
```

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

```
  },
```

```
  bgImg:{
```

```
    flex:2,
```

},

```
txt1: {  
  
  paddingBottom: responsiveWidth(5),  
  
  fontSize: responsiveFontSize(6),  
  
  color: 'rgba(255,255,255,0.67)',  
  
  fontFamily: 'NanumMyeongjo',  
  
},
```

```
txt2:{  
  
  paddingTop: responsiveWidth(5),  
  
  fontSize: responsiveFontSize(6),  
  
  color: 'white',  
  
  fontFamily: 'NanumMyeongjoEBold',  
  
},
```

```
containerOu: {  
  
  flex: 0.8,  
  
  flexDirection: "row",  
  
  justifyContent: 'center',  
  
  alignItems: 'center',
```

11 Josias Lopo

```
gap: responsiveWidth(2),

},

txtOu: {

  color: '#B6B6B6',

  fontSize: responsiveFontSize(2.3),

  fontFamily: 'MPLUS1p',

},

btnArrowImg:{

  justifyContent: 'center',

  alignItems: 'center',

  width: responsiveWidth(3.2),

  resizeMode: "contain",

},

googleRegisBtn: {

  flex: 1,

  gap: responsiveWidth(3),

  alignItems: 'center',

},
```

```
googleBtn: {  
  
  flexDirection: 'row',  
  
  backgroundColor: 'white',  
  
  alignItems: "center",  
  
  width: responsiveHeight(30),  
  
  height: responsiveHeight(6),  
  
  justifyContent: 'center',  
  
  //borderRadius: responsiveWidth(6.5),  
  
  borderRadius: 100,  
  
  gap: responsiveWidth(2)  
  
},
```

```
googleBtnLogo: {  
  
  width: responsiveWidth(6.5),  
  
  height: responsiveHeight(3),  
  
  resizeMode: "contain",  
  
},
```

```
googleBtnTxt: {  
  
  color: 'black',  
  
  fontFamily: 'Inter',  
  
  fontSize: responsiveFontSize(1.7),
```

},

```
regisBtn: {  
  
  backgroundColor: 'rgba(29,30,38,0.88)',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  height: responsiveHeight(3.5),  
  
  width: responsiveWidth(27),  
  
  //borderRadius: responsiveWidth(4)  
  
  borderRadius: 100,  
  
},
```

```
regisBtnTxt: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1p',  
  
  
  
},
```

```
});
```

3.1.12 Home.js

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

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

import { useFonts } from 'expo-font';

import { responsiveFontSize, responsiveHeight, responsiveScreenFontSize, responsiveWidth } from "react-native-responsive-dimensions";

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

import menu from '../src/assets/images/menu.png';

import Upload from '../src/assets/images/Upload.png';

import calendar from '../src/assets/images/calendar.png';

import bride from '../src/assets/images/Bride.jpg';

import hands from '../src/assets/images/hands.jpg';

import wood from '../src/assets/images/wood.jpg';

import chatBtn from '../src/assets/images/ChatBtn.png';

import { AuthCredential, signInOut } from 'firebase/auth';

import { auth } from '../config/firebase';

import * as firebase from 'firebase/app';

import 'firebase/auth';
```

11 Josias Lopo

```
import ListWidget from './ListWidget'; // Import your List component here

import { Header, createStackNavigator } from '@react-navigation/stack';

import { Ionicons, Entypo } from '@expo/vector-icons';

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

import ChatGptIcon from '../src/assets/images/GptIcon.png';

import Gemini from '../src/assets/images/Geminilcon.png';

import Chat from '../src/assets/images/ChatIcon.png';

import PhotosPage from './MediaWidget';

import ProfileScreen from './profilePic';


const Stack = createStackNavigator();


export default function Home() {

  const navigation = useNavigation();

  const currentDate = new Date();

  const options = { month: 'long', day: 'numeric' };

  const formattedDate = currentDate.toLocaleDateString('en-US', options);


  const handleOpenCalendar = () => {

    Linking.openURL('content://com.android.calendar/time/');

    Linking.openURL('calshow://');
```

```
};

const [userEmail, setUserEmail] = useState("");

useEffect(() => {

  const unsubscribe = auth.onAuthStateChanged((user) => {

    if (user) {

      setUserEmail(user.email);

    } else {

      setUserEmail("");

    }

  });

  return () => unsubscribe();

}, []);

const handleLogout = async() =>{

  await signOut (auth);
```

```
}
```

```
const insets = useSafeAreaInsets();
```

```
const [selectedButton, setSelectedButton] = useState(0);
```

```
const handlePress = (index) => {
```

```
  setSelectedButton(index);
```

```
  console.log(`Button ${index + 1} pressed`);
```

```
};
```

```
const [loaded] = useFonts({
```

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

```
  NanumMyeongjoBold: require('../src/assets/fonts/NanumMyeongjo-Bold.ttf'),
```

```
  NanumMyeongjoEBold: require('../src/assets/fonts/NanumMyeongjo-ExtraBold.ttf'),
```

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

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

```
  InterEBold: require('../src/assets/fonts/Inter-ExtraBold.ttf'),
```

```
  MPLUS1p: require('../src/assets/fonts/MPLUS1p-Regular.ttf'),
```

```
  MPLUS1pBold: require('../src/assets/fonts/MPLUS1p-Bold.ttf'),
```

```
  MPLUS1pEBold: require('../src/assets/fonts/MPLUS1p-ExtraBold.ttf'),
```

```
});
```

```
if (!loaded){  
    return null;  
}
```

```
const data = [  
    { text: 'Todos', image: null},  
    { text: 'Videos', image: require('../src/assets/images/Videos.png') },  
    { text: 'Imagens', image: require('../src/assets/images/images.png') },  
    { text: 'Gostos', image: require('../src/assets/images/gostos.png') },  
];
```

```
return (  
    <View style={[styles.container,{paddingTop: insets.top, paddingBottom: insets.bottom - 10}]}>  
        <ScrollView style={[styles.container,{paddingTop: responsiveWidth(5)}]}  
            stickyHeaderIndices={[]}>
```

```
            <View style={[styles.headerBtns, {paddingRight: responsiveWidth(2.5), paddingLeft:  
responsiveWidth(2.5)}}]>
```

```

<View>

  <TouchableOpacity style={styles.menuBtn} onPress={() => navigation.toggleDrawer()}>

    <Image source={menu} style={styles.menuBtnImg}/>

  </TouchableOpacity>

</View>

<View style={styles.headerBtnsRight}>

  <TouchableOpacity          style={styles.menuBtnRight}          onPress={()          =>
navigation.navigate("NewPost")}>

    <Image source = {Upload} style={styles.UploadBtnImg}/>

  </TouchableOpacity>

  <TouchableOpacity          style={styles.menuBtnRight}          onPress={()          =>
navigation.navigate("Profile")}>

    <View style={styles.accountBtnImg}>

      <ProfileScreen/>

    </View>

  </TouchableOpacity>

</View>

</View>

```

```
<View style={styles.welcomeTxt}>
```

```
  <Text style={styles.Welcome}>Welcome</Text>
```

```
  <Text style={styles.userTxt}>{userEmail.split('@')[0].replace(/\./g, '')}</Text>
```

```
</View>
```

```
<View style={[styles.eventTypeDate, { marginBottom: responsiveWidth(-2.5),paddingLeft:
responsiveWidth(5.5), paddingEnd: responsiveWidth(5.5)}}]>
```

```
  <Text style={styles.eventTypeDisplay}>Eventer</Text>
```

```
  <TouchableOpacity style={styles.eventTypeDate} onPress={handleOpenCalendar}>
```

```
    <Text style={styles.eventDate}>{formattedDate}</Text>
```

```
    <Image source = {calendar} style={styles.calendarImg}/>
```

```
  </TouchableOpacity>
```

```
</View>
```

```
<View style={styles.ChatsView}>
```

```
  <TouchableOpacity style={styles.ChatsContainer}>
```

```
    <Image /*source={}*/ style={styles.ChatImage}/>
```

```
  </TouchableOpacity>
```

```
</View>
```

```
<View style={{
```

```

flex: 1,

flexDirection: 'row',

justifyContent: 'center',

alignItems: 'center',

marginTop: responsiveWidth(5),

marginBottom: responsiveWidth(3),

}}>

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

  <Image source={ChatGptIcon} style={{height: responsiveWidth(15), width:
responsiveWidth(17), resizeMode: "contain",}}/>

</TouchableOpacity>

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

  <Image source={Gemini} style={{height: responsiveWidth(15), width:
responsiveWidth(17), resizeMode: "contain",}}/>

</TouchableOpacity>

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

  <Image source={Chat} style={{height: responsiveWidth(15), width: responsiveWidth(17),
resizeMode: "contain",}}/>

</TouchableOpacity>

</View>

```

```
<View style={styles.ChatsText}><Text style={{color: '#818181', fontFamily:'MPLUS1p'}}>Your  
Chats...</Text></View>
```

```
<View style={styles.WidgetsContainer}>

  <View style={styles.WidgetsView}>

    <View style={styles.ToDoHeader}>

      <View style={{justifyContent:'left', alignItems:'left'}}>

        <Text style={styles.ToDoTitle}>To-do List</Text>

      </View>

      <View style={{justifyContent:'right', alignItems:'right'}}>

        <TouchableOpacity style={styles.TODOAddBtn}><Text style={styles.TODOAddText}
onPress={() => navigation.navigate("Todo")}><Ionicons name="arrow-forward"
size={responsiveWidth(5)} color="white" /></Text></TouchableOpacity>

      </View>

    </View>

  </View>

</View>
```

```
<View style={styles.TODOView}>

  <Stack.Navigator screenOptions={{headerShown: false}}>

    <Stack.Screen name="MyTodos" component={ListWidget} />
```

```

        </Stack.Navigator>

    </View>

    <View style={{flexDirection: 'row', marginTop: responsiveWidth(5), gap: 1,}}>

        <View style={{backgroundColor:"#1D1E26", width: responsiveWidth(2.5), height:
responsiveWidth(2.5), borderRadius: responsiveWidth(10)}}></View>

        <View style={{backgroundColor:"#1D1E26", width: responsiveWidth(2.5), height:
responsiveWidth(2.5), borderRadius: responsiveWidth(10)}}></View>

        <View style={{backgroundColor:"#1D1E26", width: responsiveWidth(2.5), height:
responsiveWidth(2.5), borderRadius: responsiveWidth(10)}}></View>

    </View>

    <PhotosPage/>

</View>

</ScrollView>

<TouchableOpacity style={styles.chatBtn} onPress={() => navigation.navigate('ChatPage')}>

    <Image source = {chatBtn} style={styles.chatBtnImg}/>

</TouchableOpacity>

</View>

);
}

```

```
const buttonWidth = 124;
```

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

```
  container: {
```

```
    flex: 1,
```

```
    backgroundColor: '#101014',
```

```
  },
```

```
  chatBtn: {
```

```
    position: 'absolute',
```

```
    bottom: responsiveWidth(10),
```

```
    right: responsiveWidth(8),
```

```
    backgroundColor: '#1D1E26',
```

```
    justifyContent: 'center',
```

```
    alignItems: 'center',
```

```
    width: responsiveWidth(16),
```

```
    height: responsiveWidth(16),
```

```
    resizeMode: "contain",
```

```
    borderRadius: 100,
```

12 Josias Lopo

},

```
chatBtnImg: {  
  width: responsiveWidth(9),  
  resizeMode: "contain",  
  marginTop: responsiveWidth(1),  
  marginRight: responsiveWidth(0.5),  
},
```

```
headerBtns: {  
  flex: 0.35,  
  flexDirection: "row",  
  marginBottom: responsiveWidth(8)  
},
```

```
headerBtnsRight: {  
  flex: 1,  
  height: 'auto',  
  flexDirection: "row",  
  justifyContent: "flex-end",  
  gap: responsiveWidth(0.3),  
},
```

```
menuBtnRight: {  
  
  backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  width: responsiveWidth(12),  
  
  height: responsiveWidth(12),  
  
  resizeMode: "contain",  
  
  //borderRadius: '100%',  
  
  borderRadius: 100,  
  
},
```

```
menuBtn: {  
  
  backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  width: responsiveWidth(12),  
  
  height: responsiveWidth(12),  
  
  resizeMode: "contain",  
  
  //borderRadius: '100%',  
  
  borderRadius: 100,
```

},

```
menuBtnImg: {  
  width: responsiveWidth(4.7),  
  resizeMode: "contain",  
},
```

```
UploadBtnImg: {  
  width: responsiveWidth(6),  
  resizeMode: "contain",  
},
```

```
accountBtnImg: {  
  width: responsiveWidth(11),  
  height: responsiveWidth(11),  
  resizeMode: "cover",  
  overflow: "hidden",  
  borderRadius: 100,  
  position: 'absolute',  
},
```

```
welcomeTxt: {
```

```
flex:0.6,  
  
justifyContent: 'center',  
  
alignItems: 'left',  
  
paddingLeft: responsiveWidth(2.5),  
  
paddingRight: responsiveWidth(2.5),  
  
},
```

```
Welcome: {  
  
  color: 'white',  
  
  fontFamily: 'NanumMyeongjo',  
  
  fontSize: responsiveFontSize(6)  
  
},
```

```
userTxt:{  
  
  color: 'white',  
  
  fontFamily: 'NanumMyeongjoBold',  
  
  fontSize: responsiveFontSize(6),  
  
  marginLeft: responsiveWidth(3)  
  
},
```

```
eventTypeDate: {  
  
  flex:1,
```

13 Josias Lopo

```
flexDirection: "row",  
  
alignItems: 'center',  
  
//backgroundColor: 'tomato',  
  
gap: responsiveWidth(1.5),  
  
},  
  
eventTypeDisplay: {  
  
  flex: 1,  
  
  textAlign:'left',  
  
  fontFamily: 'NanumMyeongjoBold',  
  
  color: '#818181',  
  
  fontSize: responsiveFontSize(2.2),  
  
},  
  
eventDate: {  
  
  flex: 1,  
  
  textAlign:'right',  
  
  fontFamily: 'NanumMyeongjoBold',  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(2.2),  
  
},
```

```
calendarImg: {  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  width: responsiveWidth(8),  
  
  resizeMode: "contain",  
  
},  
  
ChatsText:{  
  
  flex: 1,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  marginBottom: responsiveWidth(10),  
  
},
```

```
WidgetsContainer: {  
  
  justifyContent: 'center',
```

13 Josias Lopo

```
alignItems: 'center',  
  
marginTop: responsiveWidth(2),  
  
},
```

```
WidgetsView: {  
  
  width: "85%",  
  
  height: responsiveHeight(8),  
  
  backgroundColor: '#1D1E26',  
  
  paddingTop: responsiveWidth(2),  
  
  justifyContent: "center",  
  
  borderTopLeftRadius: responsiveWidth(10),  
  
  borderTopRightRadius: responsiveWidth(10),  
  
  overflow: 'hidden',  
  
  marginBottom: 1,  
  
},
```

```
TodoView: {  
  
  height: responsiveWidth(35),  
  
  width: '85%',  
  
},
```

```
ToDoHeader: {  
  
  flex:0.5,  
  
  flexDirection: "row",  
  
  alignItems: "center",  
  
  justifyContent: "space-between"  
  
},
```

```
ToDoTitle: {  
  
  color: 'white',  
  
  fontFamily: 'NanumMyeongjoBold',  
  
  fontSize: responsiveFontSize(3),  
  
  paddingLeft: responsiveWidth(8),  
  
},
```

```
TodoAddBtn: {  
  
  backgroundColor: '#101014',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  width: responsiveWidth(10),  
  
  height: responsiveWidth(10),  
  
  resizeMode: "contain",
```

```
borderRadius: 100,

marginRight: responsiveWidth(8),

},

});
```

3.1.13 List.js

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

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

import { collection, addDoc, onSnapshot, doc, updateDoc, deleteDoc } from 'firebase/firestore';

import { FIRESTORE_DB } from '../config/firebase';

import { Ionicons, Entypo } from '@expo/vector-icons';

import { responsiveFontSize, responsiveWidth } from 'react-native-responsive-dimensions';

import { getAuth, onAuthStateChanged, currentUser } from 'firebase/auth';

const List = () => {

  const [todo, setTodo] = useState("");

  const [todos, setTodos] = useState([]);

  const [user, setUser] = useState(null);

  useEffect(() => {
```

```
const auth = getAuth();

const unsubscribe = onAuthStateChanged(auth, (currentUser) => {

  setUser(currentUser);

});

return () => unsubscribe();

}, []);

const addTodo = async () => {

  if (user) {

    try {

      const docRef = await addDoc(collection(FIRESTORE_DB, 'todos'), {

        title: todo,

        done: false,

        email: user.email

      });

      setTodo("");

      console.log('Document written with ID: ', docRef.id);

    } catch (e) {

      console.error('Error adding document: ', e);

    }

  } else {
```

13 Josias Lopo

```
}  
  
};  
  
useEffect(() => {  
  
  const todoRef = collection(FIRESTORE_DB, 'todos');  
  
  const unsubscribe = onSnapshot(todoRef, (snapshot) => {  
  
    const todos = [];  
  
    snapshot.docs.forEach((doc) => {  
  
      if (doc.data().email === user?.email) {  
  
        todos.push({  
  
          id: doc.id,  
  
          ...doc.data(),  
  
        });  
  
      }  
  
    });  
  
    setTodos(todos);  
  
  });  
  
  return () => unsubscribe();  
  
}, [user]);
```

```
const toggleDone = async (itemId, done) => {

  const todoRef = doc(FIRESTORE_DB, 'todos', itemId);

  await updateDoc(todoRef, { done: !done });

};

const deleteItem = async (itemId) => {

  const todoRef = doc(FIRESTORE_DB, 'todos', itemId);

  await deleteDoc(todoRef);

};

const renderTodo = ({ item }) => {

  return (

    <View style={[styles.todoContainer, { borderRadius:"100%"}]}>

      <TouchableOpacity onPress={() => toggleDone(item.id, item.done)} style={styles.todo}>

        {item.done ? <Ionicons name="checkmark-circle" size={32} color="green" /> : <Ionicons
name="radio-button-off" size={32} color="white" />}

        <Text style={styles.todoText}>{item.title}</Text>

      </TouchableOpacity>

      <Ionicons name="close-circle" size={responsiveWidth(5.8)} color="#FF6060" onPress={() =>
deleteItem(item.id)} />

    </View>

  );

};
```

```
};  
  
//FF6060  
  
return (  
  
  <View style={styles.container}>  
  
    <View style={styles.containerTodo}>  
  
  
  
      <View style={styles.content}>  
  
        {todos.length > 0 && (  
  
          <FlatList style={styles.Flat}  
  
            data={todos}  
  
            renderItem={renderTodo}  
  
            keyExtractor={({todo}) => todo.id}  
  
          )/  
  
        )}  
  
      </View>  
  
    <View style={styles.form}>  
  
      <TextInput  
  
        style={styles.input}
```

```

placeholder="Start typing..."

placeholderTextColor="gray"

onChangeText={({text}) => setTodo(text)}

value={todo}

/>

<TouchableOpacity style={styles.ButtonAdd} onPress={addTodo} disabled={todo === ""}>

  <Text style={styles.TextAdd}>ADD</Text>

</TouchableOpacity>

</View>

</View>

</View>

);

};

```

```

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#1D1E26',

  },

  containerTitle: {

```

```
flex: 0.2,  
  
justifyContent: 'flex-start',  
  
},  
  
containerTodo: {  
  
  flex: 1,  
  
  paddingLeft: responsiveWidth(10),  
  
  paddingRight: responsiveWidth(10),  
  
  paddingTop: responsiveWidth(5),  
  
  backgroundColor: '#101014',  
  
  borderTopLeftRadius: responsiveWidth(10),  
  
  borderTopRightRadius: responsiveWidth(10),  
  
  
  
},  
  
content: {  
  
  flex: 1,  
  
},  
  
form: {  
  
  flex: 0.1,
```

```
marginVertical: responsiveWidth(5),  
  
flexDirection: 'row',  
  
alignItems: 'center',  
  
},
```

```
input: {  
  
  flex: 1,  
  
  height: responsiveWidth(15),  
  
  borderRadius: responsiveWidth(100),  
  
  backgroundColor: "#1D1E26",  
  
  color: 'white',  
  
  paddingLeft: responsiveWidth(5),  
  
  paddingRight: responsiveWidth(5),  
  
  fontFamily: 'MPLUS1p',  
  
},
```

```
ButtonAdd: {  
  
  paddingLeft: responsiveWidth(5),  
  
  paddingRight: responsiveWidth(5),  
  
},
```

```
TextAdd: {
```

14 Josias Lopo

```
color:"blue",

fontFamily:"MPLUS1p",

fontSize: responsiveFontSize(2)

},

todo: {

  flexDirection: 'row',

  flex: 1,

  alignItems: 'center',

},

todoText: {

  flex: 1,

  paddingHorizontal: 4,

  color: 'white',

  fontSize: responsiveFontSize(2.3),

  paddingLeft: 10,

},

todoContainer: {

  flexDirection: 'row',

  alignItems: 'center',
```

```
padding: 10,  
  
marginVertical: 4,  
  
},  
  
Flat: {  
  
  flex: 1,  
  
}  
  
});
```

```
export default List;
```

3.1.14 ListWidget.js

```
import React, { useState, useEffect } from 'react';  
  
import { View, StyleSheet, TextInput, Button, FlatList, TouchableOpacity, Text } from 'react-native';  
  
import { collection, addDoc, onSnapshot, doc, updateDoc, deleteDoc } from 'firebase/firestore';  
  
import { FIRESTORE_DB } from '../config/firebase';  
  
import { Ionicons, Entypo } from '@expo/vector-icons';  
  
import { responsiveFontSize, responsiveWidth } from 'react-native-responsive-dimensions';  
  
import { getAuth, onAuthStateChanged, currentUser } from 'firebase/auth';
```

```
const ListWidget = () => {  
  
  const [todo, setTodo] = useState("");  
  
  const [todos, setTodos] = useState([]);  
  
  const [user, setUser] = useState(null);  
  
  useEffect(() => {  
  
    const auth = getAuth();  
  
    const unsubscribe = onAuthStateChanged(auth, (currentUser) => {  
  
      setUser(currentUser);  
  
    });  
  
    return () => unsubscribe();  
  
  }, []);  
  
  const addTodo = async () => {  
  
    if (user) {  
  
      try {  
  
        const docRef = await addDoc(collection(FIRESTORE_DB, 'todos'), {  
  
          title: todo,  
  
          done: false,  
  
          email: user.email  
  
        });  
  
      }  
  
    }  
  
  }  
  
}
```

```
});  
  
setTodo("");  
  
console.log('Document written with ID: ', docRef.id);  
} catch (e) {  
  
  console.error('Error adding document: ', e);  
  
}  
  
} else {  
  
  
}  
};  
  
useEffect(() => {  
  
  const todoRef = collection(FIRESTORE_DB, 'todos');  
  
  
  
  
  const unsubscribe = onSnapshot(todoRef, (snapshot) => {  
  
    const todos = [];  
  
    snapshot.docs.forEach((doc) => {  
  
      if (doc.data().email === user?.email) {  
  
        todos.push({  
  
          id: doc.id,  
  
          ...doc.data(),  
  
        });  
  
      }  
  
    });  
  
  });  
  
});
```

14 Josias Lopo

```

    }

  });

  setTodos(todos);

});

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

const toggleDone = async (itemId, done) => {

  const todoRef = doc(FIRESTORE_DB, 'todos', itemId);

  await updateDoc(todoRef, { done: !done });

};

const deleteItem = async (itemId) => {

  const todoRef = doc(FIRESTORE_DB, 'todos', itemId);

  await deleteDoc(todoRef);

};

const renderTodo = ({ item }) => {

  return (

    <View style={[styles.todoContainer, { borderRadius:"100%"}]}>

    <TouchableOpacity onPress={() => toggleDone(item.id, item.done)} style={styles.todo}>

```

```

        {item.done ? <Ionicons name="checkmark-circle" size={responsiveWidth(7.5)}
color="green" /> : <Ionicons name="radio-button-off" size={responsiveWidth(7.5)} color="white" />}

        <Text style={styles.todoText}>{item.title}</Text>

        </TouchableOpacity>

        </View>

    );

};

return (

    <View style={styles.container}>

        <View style={{height: '100%', width: '100%', backgroundColor: '#1D1E26',
borderBottomLeftRadius: responsiveWidth(10), borderBottomRightRadius: responsiveWidth(10)}}>

            <View style={styles.content}>

                {todos.length > 0 && (

                    <FlatList style={styles.Flat}

                        data={todos}

                        renderItem={renderTodo}

                        keyExtractor={({todo} => todo.id}

                    />

                )}

            </View>

        </View>

    )
}

```

```
</View>

</View>

</View>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#101014',

  },

  content: {

    padding: responsiveWidth(3),

    paddingBottom: 0,

  },

  todo: {

    flexDirection: 'row',

    flex: 1,
```

```
alignItems: 'center',  
  
marginBottom: responsiveWidth(3),  
  
},  
  
todoText: {  
  
  flex: 1,  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(2.3),  
  
  paddingLeft: 10,  
  
},  
  
todoContainer: {  
  
  flexDirection: 'row',  
  
  alignItems: 'center',  
  
  marginLeft: responsiveWidth(10),  
  
},  
  
});
```

export default ListWidget;

3.1.15 Login.js

15 Josias Lopo

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

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

import { useFonts } from 'expo-font';

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

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

import Seta2 from "../src/assets/images/arrowLeft.png";

import GLogo from "../src/assets/images/googlelogo.png";

import { auth } from "../config/firebase";

import { signInWithEmailAndPassword } from "firebase/auth";

export default function Login({ navigation }) {

  const insets = useSafeAreaInsets();

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [errorMessage, setErrorMessage] = useState("");
```

```
const handleSubmit = async () => {

  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  if (!email || !password) {

    setErrorMessage('Please enter your email and password.');
```

```
  } else if (!emailRegex.test(email)) {

    setErrorMessage('Please enter a valid email address.');
```

```
  } else {

    try {

      await signInWithEmailAndPassword(auth, email, password);

      setErrorMessage('');
```

```
    } catch (err) {

      setErrorMessage('Invalid email or password.');
```

```
    }

  }

};

const [loaded] = useFonts({

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

  NanumMyeongjoBold: require('../src/assets/fonts/NanumMyeongjo-Bold.ttf'),

  NanumMyeongjoEBold: require('../src/assets/fonts/NanumMyeongjo-ExtraBold.ttf'),
```

```

Inter: require('../src/assets/fonts/Inter-Regular.ttf'),
InterBold: require('../src/assets/fonts/Inter-Bold.ttf'),
InterEBold: require('../src/assets/fonts/Inter-ExtraBold.ttf'),
MPLUS1p: require('../src/assets/fonts/MPLUS1p-Regular.ttf'),
MPLUS1pBold: require('../src/assets/fonts/MPLUS1p-Bold.ttf'),
MPLUS1pEBold: require('../src/assets/fonts/MPLUS1p-ExtraBold.ttf'),
});

if (!loaded){
  return null;
}

return (

  <View style={[styles.container,{paddingTop: insets.top + responsiveWidth(5), paddingBottom:
insets.bottom, paddingLeft: insets.left + responsiveWidth(3), paddingRight: insets.left +
responsiveWidth(3)}}>

    <TouchableOpacity style={styles.btnBack} onPress={() => navigation.navigate("Hall")}>

      <View style={styles.btnArrow}>

```

```

<Image source = {Seta2} style={styles.btnArrowImg}/>

</View>

<Text style={{styles.txtLoginBtn, {marginLeft: responsiveWidth(3)}}>Login</Text>

</TouchableOpacity>

<View style={styles.loginContent}>

  <View style={styles.logoContent}>

    <Text style={styles.logo}>

      Eventer

    </Text>

    <Text style={styles.logoDot}>.</Text>

  </View>

  <View

    style={{styles.userData, { flex: 1 }}}

  >

    <View style={styles.Name}>

      <Text style={styles.userNamePassword}>E-mail adress:</Text>

```

```
<TextInput placeholder="" style = {styles.userInput }
value={email}
onChangeText={value => setEmail(value)}/>
</View>
```

```
<View style={styles.Password}>
  <Text style={styles.userNamePassword}>Password:</Text>
  <TextInput
    placeholder=""
    style={styles.userInput}
    value={password}
    onChangeText={value => setPassword(value)}
    secureTextEntry={true}
  />
</View>
</View>
```

```
{errorMessage !== " " && (
  <Text style={styles.errorMessage}>{errorMessage}</Text>
)}
```

```
</View>
```

```
<View style={styles.googleRegisBtn}>
```

```
<View style={styles.Login}>
```

```
<TouchableOpacity
```

```
onPress={handleSubmit}>
```

```
<View style={styles.btnArrowRotate}>
```

```
<Image source = {Seta2} style={styles.btnArrowImg}/>
```

```
</View>
```

```
</TouchableOpacity>
```

```
</View>
```

```
<TouchableOpacity style={styles.regisBtn} onPress={() => navigation.navigate("Regist")}>
```

```
<Text style={styles.regisBtnTxt}>Don't have an account? </Text>
```

```
<Text style={styles.regisBtnLink}>Register</Text>
```

```
</TouchableOpacity>
```

```
</View>
```

```
</View>
```

```
);
```

```
};
```

15 Josias Lopo

```
const styles = StyleSheet.create({  
  
  container: {  
  
    flex: 1,  
  
    backgroundColor: '#101014',  
  
  },  
  
  scrollContainer: {  
  
    flexGrow: 0.1,  
  
  },  
  
  btnBack:{  
  
    width: responsiveWidth(25),  
  
    flexDirection: "row",  
  
    alignItems: "center",  
  
  },  
  
  btnArrow: {  
  
    justifyContent: 'center',
```

```
alignItems: 'center',  
  
backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
width: responsiveWidth(10),  
  
height: responsiveWidth(10),  
  
resizeMode: "contain",  
  
borderRadius: 100,  
  
},
```

```
btnArrowImg: {  
  
  
  
  
  
  
  
  
  
width: responsiveWidth(4.5),  
  
resizeMode: "contain",  
  
},
```

```
btnArrowRotate: {  
  
justifyContent: 'center',  
  
alignItems: 'center',  
  
backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
width: responsiveWidth(13),  
  
height: responsiveWidth(13),  
  
resizeMode: "contain",  
  
borderRadius: 100,
```

16 Josias Lopo

```
transform: [{ rotate: '180deg' }],  
  
},
```

```
txtLoginBtn: {  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(2),  
  
  fontFamily: 'MPLUS1p',  
  
},
```

```
loginContent: {  
  
  flex: 1.5,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},
```

```
logoContent: {  
  
  flex: 1,  
  
  flexDirection: 'row',  
  
  alignItems: 'center',
```

},

logo: {

color: 'white',

fontFamily: 'MPLUS1pEBold',

fontSize: responsiveFontSize(6),

},

logoDot: {

color: '#1563FF',

fontFamily: 'MPLUS1pEBold',

fontSize: responsiveFontSize(6),

},

userData: {

flex: 1,

width: responsiveHeight(33),

},

Name:{

flex: 1,

16 Josias Lopo

```
justifyContent: 'center',  
},
```

```
Password:{  
  
  flex: 1,  
  
  justifyContent: 'center',  
  
},
```

```
userNamePassword: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(2.3),  
  
},
```

```
userInput: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(2),  
  
  borderBottomColor: '#525468',  
  
  borderBottomWidth: responsiveWidth(0.10),  
  
},
```

```
Login: {  
  
  paddingTop: responsiveHeight(3),  
  
  flex: 0.70,  
  
  flexDirection: 'row',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},
```

```
googleRegisBtn: {  
  
  flex: 1.3,  
  
  alignItems: 'center',  
  
},
```

```
regisBtn: {  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  height: responsiveHeight(3.5),  
  
  borderRadius: 100,  
  
  flexDirection: 'row',  
  
},
```

```
regisBtnTxt: {
```

16 Josias Lopo

```
color: 'white',

fontFamily: 'MPLUS1p',

},

regisBtnLink: {

color: '#1563FF',

fontFamily: 'MPLUS1p',

},

errorMessage: {

color: '#EA4335',

fontFamily: 'MPLUS1p',

fontSize: responsiveFontSize(1.8),

marginTop: responsiveHeight(1),

},

});
```

3.1.16 MediaWidget.js

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

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

import { getStorage, ref, app, getDownloadURL, listAll, deleteObject } from 'firebase/storage';

import { getAuth } from 'firebase/auth';

import { getFirestore, collection, query, where, getDocs, deleteDoc, doc } from 'firebase/firestore';

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

import { Ionicons } from '@expo/vector-icons';

const PhotosPage = () => {

  const [imageData, setImageData] = useState([]);

  const db = getFirestore(app);

  const storage = getStorage(app);

  useEffect(() => {

    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 imageUrls = await Promise.all(listResult.items.map(async imageRef => {

  const url = await getDownloadURL(imageRef);

  const querySnapshot = await getDocs(query(collection(db, 'photoDescriptions'),
where("imageRef", "=", imageRef.toString())));

  let imageData = {};

  querySnapshot.forEach((doc) => {

    imageData = doc.data();

  });

  return { url, ...imageData };

}));

setImageData(imageUrls);

};
```

```
fetchImages();

const intervalId = setInterval(fetchImages, 10000);

return () => clearInterval(intervalId);
}, [db, storage]));

const handleDelete = async (imageRefString) => {
  Alert.alert(
    "Confirm Delete",
    "Are you sure you want to delete this photo and its description?",
    [
      {
        text: "Cancel",
        style: "cancel"
      },
      {
        text: "Delete",
        onPress: async () => {
          try {
```

```

const imageRef = ref(storage, imageRefString);

await deleteObject(imageRef);

const querySnapshot = await getDocs(query(collection(db, 'photoDescriptions'),
where("imageRef", "==", imageRefString)));

querySnapshot.forEach(async (doc) => {

    await deleteDoc(doc.ref);

});

setImageData(imageData.filter(data => data.imageRef !== imageRefString));

} catch (error) {

    console.error("Error deleting image:", error);

}

},

style: "destructive"

}

]

);

};

```

```
return (

  <View style={styles.container}>

    <ScrollView>

      {imageData.map((data, index) => (

        <View key={index} style={{ justifyContent: 'center', alignItems: 'center' }}>

          <Image source={{ uri: data.url }} style={styles.image} />

          <View style={styles.descriptionContainer}>

            <Text style={styles.title}>{data.title}</Text>

            <Text style={styles.description}>{data.description}</Text>

            <Text style={styles.hashtags}>{data.hashtags}</Text>

            <TouchableOpacity

              style={styles.deleteButton}

              onPress={() => handleDelete(data.imageRef)}

            >

              <Ionicons name="close-circle" size={responsiveWidth(5.8)} color="#898989" />

            </TouchableOpacity>

          </View>

        </View>

      )}

    </ScrollView>

  </View>

)
```

17 Josias Lopo

```
    )}}  
  
  </ScrollView>  
  
  </View>  
  
  );  
};  
  
const { width } = Dimensions.get('window');  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    padding: 10,  
    justifyContent: 'center',  
    alignItems: 'center',  
    width: '100%',  
    backgroundColor: '#101014'  
  },  
  
  image: {  
    width: width - 60,  
    height: responsiveHeight(24),
```

```
borderRadius: responsiveWidth(12),  
  
position: 'relative',  
  
top: 10,  
  
zIndex:1,  
  
},  
  
descriptionContainer: {  
  
  marginBottom: responsiveWidth(5),  
  
  backgroundColor: "#1D1E26",  
  
  width: responsiveWidth(75),  
  
  height: responsiveHeight(16),  
  
  justifyContent: 'flex-start',  
  
  paddingTop: responsiveWidth(5),  
  
  paddingLeft: responsiveWidth(10),  
  
  paddingRight: responsiveWidth(10),  
  
  paddingBottom: responsiveWidth(2),  
  
  borderBottomLeftRadius: responsiveWidth(12),  
  
  borderBottomRightRadius: responsiveWidth(12),  
  
  },
```

```
title: {  
  
  color: "white",  
  
  fontSize: responsiveFontSize(2.5),  
  
  marginBottom: responsiveWidth(1),  
  
  fontFamily: "MPLUS1pBold",  
  
},  
  
description: {  
  
  color: "white",  
  
  fontSize: responsiveFontSize(1.8),  
  
  marginBottom: responsiveWidth(1),  
  
  fontFamily: "MPLUS1p",  
  
  marginLeft: responsiveWidth(2),  
  
  marginBottom: responsiveWidth(2.5)  
  
},  
  
hashtags: {  
  
  fontSize: responsiveFontSize(1.5),  
  
  color: '#898989',  
  
  fontFamily: "MPLUS1p",  
  
  marginBottom: responsiveWidth(3.5),  
  
  marginLeft: responsiveWidth(2)  
  
},
```

```
deleteButton: {  
  position: 'absolute',  
  top: responsiveWidth(6),  
  right: responsiveWidth(10),  
},  
});
```

```
export default PhotosPage;
```

3.1.17 Menu.js

```
import React from 'react';  
import { DrawerItem } from '@react-navigation/drawer';  
import { View } from 'react-native';
```

```
const Menu = ({ navigation }) => {  
  const menuItems = [  
    { title: 'Home', screenName: 'Home' },  
  ];
```

```
  return (  
    <View>
```

17 Josias Lopo

```
{menuItems.map((item) => (
  <DrawerItem
    key={item.title}
    label={item.title}
    onPress={() => navigation.navigate(item.screenName)}
  />
))}
</View>
);
};
```

export default Menu;

3.1.18 NewPost.js

```
import React, {useState} from 'react'

import {View, StyleSheet, Image, Text, TouchableOpacity, SafeAreaView, Alert, TextInput} from
'react-native'

import * as ImagePicker from 'expo-image-picker'

import { firebase } from '../config/firebase'; // Importe o serviço de armazenamento

import { getStorage, ref, app, uploadBytes } from 'firebase/storage';

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

import { getAuth } from 'firebase/auth';
```

```
import { getFirestore, collection, addDoc } from 'firebase/firestore';

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

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

import Seta2 from "../src/assets/images/arrowLeft.png";

const UploadScreen = () => {

  const insets = useSafeAreaInsets();

  const db = getFirestore(app);

  const navigation = useNavigation();

  const [image, setImage] = useState(null)

  const [uploading, setUploading] = useState(false)

  const [title, setTitle] = useState("");

  const [description, setDescription] = useState("");

  const [hashtags, setHashtags] = useState("");
```

```
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 {

    uploadBytes(storageRef, blob);

} catch (e) {

    console.log("Upload Error:", e);

}
```

17 Josias Lopo

```
setUploading(false)

Alert.alert(

  'Photo uploaded!'

);

setImage(null);

try {

const docRef = await addDoc(collection(db, "photoDescriptions"), {

  title: title,

  description: description,

  hashtags: hashtags,

  imageRef: storageRef.toString(),

  userID: uid

});

console.log("Document written with ID: ", docRef.id);

} catch (e) {

console.error("Error adding document: ", e);

}

setTitle("");

setDescription("");

setHashtags("");
```

```
console.log("Image uploaded to:", storageRef.toString());

};

return(
  <SafeAreaView style={{styles.container,{paddingTop: insets.top, paddingBottom: insets.bottom}}}>

    <View style={styles.header}>
      <Text style={styles.headerTxt}>NewPost</Text>

      <TouchableOpacity style={styles.btnArrow} onPress={uploadImage}>
        <Image source={Seta2} style={styles.btnArrowImg} />
      </TouchableOpacity>
    </View>

    <View style={styles.imageContainer}>
      {image && <Image source={{uri: image.uri}} style={{width: "100%", height: "100%"}}/>}
    </View>
  </SafeAreaView>
);
```

```
</View>
```

```
<TouchableOpacity style={styles.selectButton} onPress={pickImage}>
```

```
  <Text style={styles.btnText}>Select</Text>
```

```
</TouchableOpacity>
```

```
<View style={styles.description}>
```

```
  <View style={styles.inputTitle}>
```

```
    <TextInput style={[styles.inputTitleTxt]} onChangeText={setTitle}
```

```
      value={title} placeholder="Title" placeholderTextColor="gray"
```

```
  />
```

```
</View>
```

```
<View style={styles.descriptionContainer}>
```

```
  <View style={styles.inputDescription}>
```

```
    <TextInput style={styles.inputDescriptionTxt} onChangeText={setDescription}
```

```
      value={description} placeholder="Write your description" placeholderTextColor="gray"
```

```
      multiline={true}
```

```
      numberOfLines={4}
```

```
      maxLength={60}
```

```

        textAlignVertical="top"

        flexWrap="wrap"

    />

</View>

<View style={styles.inputHashtag}>

  <TextInput style={styles.inputHashtagTxt} onChangeText={setHashtags} maxLength={35}

    value={hashtags}      placeholder="#hashtag      or      second      description"
placeholderTextColor="gray"

  />

</View>

</View>

</View>

</SafeAreaView>

);

}

```

```

const styles = StyleSheet.create({

  container: {

```

```
flex: 1,  
  
backgroundColor: '#101014'  
  
},  
  
header: {  
  
  width: '80%',  
  
  flexDirection: "row",  
  
  justifyContent: "space-between",  
  
  alignSelf: 'center',  
  
  marginTop: responsiveWidth(6),  
  
  marginBottom: responsiveWidth(7),  
  
},  
  
headerTxt : {  
  
  fontSize: responsiveFontSize(5),  
  
  color: 'white',  
  
  textAlign: 'center',  
  
  fontFamily: 'NanumMyeongjo',  
  
},  
  
btnArrow: {  
  
  justifyContent: "center",
```

```
alignItems: "center",  
  
backgroundColor: "#1D1E26",  
  
marginLeft: responsiveWidth(7),  
  
height: responsiveWidth(10),  
  
width: responsiveWidth(10),  
  
resizeMode: "contain",  
  
borderRadius: responsiveWidth(100),  
  
},
```

```
btnArrowImg: {  
  
width: responsiveWidth(3.5),  
  
resizeMode: "contain",  
  
transform: [{ rotate: '180deg' }],  
  
},
```

```
imageContainer: {  
  
flex: 1.5,  
  
backgroundColor: 'rgba(29, 30, 38, 0.30)',  
  
justifyContent: 'flex-end',  
  
},
```

```
selectButton: {
```

18 Josias Lopo

```

backgroundColor: '#1D1E26',

width: responsiveWidth(35),

height: responsiveWidth(10),

borderRadius: responsiveWidth(100),

justifyContent: 'center',

alignItems: 'center',

zIndex: 1,

position: 'relative',

bottom: responsiveWidth(5), // Define a distância entre o botão e a parte inferior

alignSelf: 'center', // Centraliza o botão horizontalmente

marginTop: responsiveWidth(2)

},

btnText: {

  color: 'white',

  fontSize: responsiveFontSize(1.9),

  fontFamily: 'MPLUS1pBold',

},

description: {

  flex: 1,

  alignItems: 'center',

```

},

```
inputTitle: {  
  width: '80%',  
  justifyContent: 'center',  
  paddingLeft: responsiveWidth(4)  
},
```

```
inputTitleTxt: {  
  color: 'white',  
  fontSize: responsiveFontSize(3),  
  fontFamily: 'MPLUS1pBold',  
},
```

```
descriptionContainer: {  
  width: '80%',  
  alignItems: 'center',  
  marginTop: responsiveWidth(2),  
  backgroundColor: '#1D1E26',  
  borderRadius: responsiveWidth(8),  
},
```

```
inputDescription: {  
  
  width: '100%',  
  
  justifyContent: 'top',  
  
  padding: responsiveWidth(4),  
  
  height: responsiveHeight(15),  
  
},  
  
inputDescriptionTxt: {  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(2),  
  
  fontFamily: 'MPLUS1p',  
  
},  
  
inputHashtag: {  
  
  width: '100%',  
  
  justifyContent: 'center',  
  
  height: responsiveHeight(8),  
  
  padding: responsiveWidth(4),  
  
  backgroundColor: 'rgba(16,16,20,0.84)',  
  
  borderRadius: responsiveWidth(8)  
  
},
```

```
inputHashtagTxt: {  
  
  color: 'gray',  
  
  fontSize: responsiveFontSize(2),  
  
  fontFamily: 'MPLUS1p',  
  
},  
  
})  
  
export default UploadScreen;
```

//lighshot

3.1.19 NoteDetail.js

```
import React, { useState, useEffect } from 'react';  
  
import { View, Text, TextInput, StyleSheet, Button, Alert, TouchableOpacity, Image } from 'react-native';  
  
import { doc, getDoc, updateDoc, deleteDoc } from 'firebase/firestore';  
  
import { FIRESTORE_DB } from '../config/firebase';  
  
import { useNavigation } from '@react-navigation/native';
```

18 Josias Lopo

```
import { responsiveFontSize, responsiveHeight, responsiveScreenFontSize, responsiveWidth } from
"react-native-responsive-dimensions";

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

import Seta2 from "../src/assets/images/arrowLeft.png";

export default function DetalheNota({ route }) {

  const { notald } = route.params;

  const [titulo, setTitulo] = useState("");

  const [conteudo, setConteudo] = useState("");

  const [shouldSave, setShouldSave] = useState(false);

  const navigation = useNavigation();

  const insets = useSafeAreaInsets();

  useEffect(() => {

    const carregarNota = async () => {

      const docRef = doc(FIRESTORE_DB, 'notas', notald);

      const docSnap = await getDoc(docRef);

      if (docSnap.exists()) {

        setTitulo(docSnap.data().titulo);

        setConteudo(docSnap.data().conteudo);

      }

    }

  })
}
```

```
};

carregarNota();

let timeoutId;

if (shouldSave) {

  timeoutId = setTimeout(() => {

    salvarNota();

    setShouldSave(false);

  }, 10000);

}

return () => clearTimeout(timeoutId);

}, [shouldSave]);

const salvarNota = async () => {

  const docRef = doc(FIRESTORE_DB, 'notas', notalId);

  await updateDoc(docRef, { titulo, conteudo });

};

const handleBlur = () => {

  setShouldSave(true);
```

```
};
```

```
const handlePress = () => {  
  
  salvarNota();  
  
  navigation.goBack();  
  
};
```

```
const apagarNota = async () => {  
  
  Alert.alert(  
  
    "Delete Note",  
  
    "Are you sure you want to DELETE?",  
  
    [  
  
      { text: "Cancel", style: "cancel" },  
  
      {  
  
        text: "Delete",  
  
        style: "destructive",  
  
        onPress: async () => {  
  
          try {  
  
            await deleteDoc(doc(FIRESTORE_DB, 'notas', notaid));  
  
            navigation.goBack();  
  
          } catch (error) {  
  
            console.error("Error deleting note:", error);  
  
          }  
  
        }  
  
      ]  
  
    )  
  
  }  
  
};
```

```
    }  
    },  
    },  
  ]  
);  
};  
  
return (  
  <View style={[styles.container, {paddingTop: insets.top}]}>  
    <View style={[styles.notesContainer, {paddingBottom: insets.bottom - 10}]}>  
      <View style={styles.title}>  
        <TouchableOpacity style={styles.btnArrow} onPress={handlePress}>  
          <Image source={Seta2} style={styles.btnArrowImg} />  
        </TouchableOpacity>  
        <View style={styles.inputContainer}>  
          <TextInput  
            style={styles.input}  
            value={titulo}  
            onChangeText={setTitulo}  
            onBlur={handleBlur}  
          />  
        </View>  
      </View>  
    </View>  
  </View>  
)
```

```

    </View>

  </View>

  <TextInput

    style={styles.inputArea}

    multiline={true}

    value={conteudo}

    onChangeText={setConteudo}

    onBlur={handleBlur}

  />

  <View style={styles.deleteBtn}>

    <TouchableOpacity onPress={apagarNota}>

      <Text style={styles.deleteBtnText}>Delete Note</Text>

    </TouchableOpacity>

  </View>

</View>

</View>

);
}

```

```

const styles = StyleSheet.create({

  container: {

```

```
flex: 1,  
  
backgroundColor: '#1D1E26',  
  
},  
  
notesContainer: {  
  
  flex: 1,  
  
  backgroundColor: '#101014',  
  
  marginTop: responsiveWidth(10),  
  
  borderTopRightRadius: responsiveWidth(10),  
  
  borderTopLeftRadius: responsiveWidth(10),  
  
},
```

```
title: {  
  
  flex: 1,  
  
  flexDirection: 'row',  
  
  marginBottom: responsiveWidth(10),  
  
  borderBottomWidth: 1,  
  
  borderBottomColor: '#ccc',  
  
  alignItems: 'center',  
  
},
```

```
btnArrow: {  
  
  marginTop: responsiveWidth(3),  
  
  justifyContent: "center",  
  
  alignItems: "center",  
  
  width: responsiveWidth(10),  
  
  height: responsiveWidth(10),  
  
  resizeMode: "contain",  
  
  marginLeft: responsiveWidth(5)  
  
},
```

```
btnArrowImg: {  
  
  width: responsiveWidth(5),  
  
  resizeMode: "contain",  
  
},
```

```
inputContainer: {  
  
  flex: 1,  
  
  alignItems: 'center',  
  
},
```

```
input: {  
  
  fontSize: responsiveFontSize(5),
```

```
fontFamily: 'NanumMyeongjo',  
  
color: 'white',  
  
marginTop: responsiveWidth(3),  
  
paddingRight: responsiveWidth(18)  
  
},
```

```
inputArea: {  
  
  flex: 5,  
  
  fontSize: 18,  
  
  color: 'white',  
  
  paddingLeft: responsiveWidth(10),  
  
  paddingRight: responsiveWidth(10)  
  
},
```

```
deleteBtn: {  
  
  fontSize: responsiveFontSize(2),  
  
  padding: 10,  
  
  borderRadius: 100,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  backgroundColor: '#1D1E26',
```

19 Josias Lopo

```
width: responsiveWidth(30),

alignSelf: 'center',

marginBottom: responsiveWidth(2)

},

deleteBtnText: {

color: '#EA4335',

fontSize: responsiveFontSize(1.8),

}

});
```

3.1.20 Notes.js

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

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

import { collection, query, where, onSnapshot } from 'firebase/firestore';

import { FIRESTORE_DB, auth } from '../config/firebase';

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

import { responsiveFontSize, responsiveHeight, responsiveScreenFontSize, responsiveWidth } from
"react-native-responsive-dimensions";

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

```
import Seta2 from "../src/assets/images/arrowLeft.png";

export default function ListaNotas() {

  const [notas, setNotas] = useState([]);

  const navigation = useNavigation();

  const insets = useSafeAreaInsets();

  useEffect(() => {

    const unsubscribe = onSnapshot(

      query(collection(FIRESTORE_DB, 'notas'), where('userId', '==', auth.currentUser.uid)),

      (snapshot) => {

        setNotas(snapshot.docs.map(doc => ({ id: doc.id, ...doc.data() })));

      }

    );

    return () => unsubscribe();

  }, []);

  return (

    <View style={{styles.container,{paddingBottom: insets.bottom - 10}}>

      <View style={{styles.header,{paddingTop: insets.top}}>
```

```

<Text style={styles.title}>Notes</Text>

<Text style={styles.description}>Imagine and write...</Text>

</View>

<FlatList
  data={notas}
  keyExtractor={({item}) => item.id}
  renderItem={({ item }) => (
    <TouchableOpacity
      style={styles.notaItem}
      onPress={() => navigation.navigate('NoteDetail', { notaid: item.id })}
    >
      <Text style={styles.notaTitulo}>{item.titulo}</Text>
      <View style={styles.btnArrow}>
        <Image source={Seta2} style={styles.btnArrowImg} />
      </View>
    </TouchableOpacity>
  )}
/>

<TouchableOpacity
  style={styles.botaoNovaNota}
  onPress={() => navigation.navigate('CreateNote')}

```

```
>  
  
  <Text style={styles.botaoTexto}>+</Text>  
  
  </TouchableOpacity>  
  
  </View>  
  
);  
}
```

```
const styles = StyleSheet.create({  
  
  container: {  
  
    flex: 1,  
  
    backgroundColor: '#1D1E26',  
  
    justifyContent: 'center',  
  
    alignItems: 'center'  
  
  },  
  
  header: {  
  
    backgroundColor: '#101014',  
  
    marginBottom: responsiveWidth(8),  
  
    width: '100%',  
  
    borderBottomLeftRadius: responsiveWidth(16),  
  
    borderBottomRightRadius: responsiveWidth(16),  
  
    paddingBottom: responsiveWidth(10),
```

20 Josias Lopo

```
paddingLeft: responsiveWidth(15)  
  
},
```

```
title: {  
  
  color: 'white',  
  
  fontSize: responsiveFontSize(6),  
  
  fontFamily: 'NanumMyeongjo',  
  
  marginTop: responsiveWidth(2)  
  
},
```

```
description: {  
  
  color: '#9F9F9F',  
  
  fontSize: responsiveFontSize(1.8),  
  
  fontFamily: 'MPLUS1p',  
  
},
```

```
notaltem: {  
  
  padding: responsiveWidth(4),  
  
  marginBottom: responsiveWidth(5),  
  
  backgroundColor: '#101014',  
  
  borderRadius: responsiveWidth(6),
```

```
flexDirection: 'row',  
  
justifyContent: 'space-between',  
  
width: responsiveWidth(70),  
  
overflow: 'hidden',  
  
alignItems: 'center'  
  
},  
  
notaTitulo: {  
  
  fontSize: responsiveFontSize(2.5),  
  
  fontFamily: 'MPLUS1p',  
  
  color: 'white',  
  
  marginLeft: responsiveWidth(3)  
  
},  
  
btnArrow: {  
  
  justifyContent: "center",  
  
  alignItems: "center",  
  
  width: responsiveWidth(10),  
  
  resizeMode: "contain",  
  
},
```

```
btnArrowImg: {  
  
  width: responsiveWidth(3.5),
```

20 Josias Lopo

```
        resizeMode: "contain",  
  
        transform: [{ rotate: '180deg' }],  
  
    },  
  
    botaoNovaNota: {  
  
        position: 'absolute',  
  
        bottom: responsiveWidth(10),  
  
        right: responsiveWidth(6),  
  
        backgroundColor: '#1563FF', // Cor verde  
  
        borderRadius: 100,  
  
        width: responsiveWidth(15),  
  
        height: responsiveWidth(15),  
  
        alignItems: 'center',  
  
        justifyContent: 'center',  
  
        elevation: 10, // Sombra para Android  
  
    },  
  
    botaoTexto: {  
  
        color: 'white',  
  
        fontSize: responsiveFontSize(4),  
  
        marginBottom: responsiveWidth(1),  
  
    },  
  
});
```

3.1.21 Profile.js

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

import { View, StyleSheet, Image, Text, TouchableOpacity, ScrollView, Alert, TextInput, Platform }
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 { getFirestore, collection, addDoc } from 'firebase/firestore';

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

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

import Seta2 from "../src/assets/images/arrowLeft.png";

import camera from "../src/assets/images/camera.png";

import account from "../src/assets/images/account.png";

import { auth } from '../config/firebase';

import ProfileScreen from './profilePic2';

const Profile = () => {
```

```
const insets = useSafeAreaInsets();

const db = getFirestore(app);

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 };

  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 = "profilePic";
```

```
  const storageRef = ref(storage, `profilePic/${uid}/${filename}`);
```

```
  try {
```

```
    uploadBytes(storageRef, blob);
```

```
  } catch (e) {
```

```
    console.log("Upload Error:", e);
```

```
  }
```

```
  setUploading(false);
```

20 Josias Lopo

```
Alert.alert(  
  'Photo uploaded!'  
);  
setImage(null);  
  
};  
  
const [userEmail, setUserEmail] = useState("");  
  
useEffect(() => {  
  const unsubscribe = auth.onAuthStateChanged((user) => {  
    if (user) {  
      setUserEmail(user.email);  
    } else {  
      setUserEmail("");  
    }  
  });  
  
  return () => unsubscribe();  
}, []);  
  
return (
```

```
<View style={{styles.container, { paddingTop: insets.top, paddingBottom: insets.bottom}}}>
```

```
<View style={styles.header}>
```

```
<TouchableOpacity style={styles.btnArrow} onPress={() => navigation.navigate("Home")}>
```

```
<Image source={Seta2} style={styles.btnArrowImg} />
```

```
</TouchableOpacity>
```

```
{image && (
```

```
<TouchableOpacity onPress={uploadImage}>
```

```
<Text style={styles.updateBtn}>Update</Text>
```

```
</TouchableOpacity>
```

```
})
```

```
</View>
```

```
<View style={styles.profilePic}>
```

```
<View style={styles.imageContainer}>
```

```
<View style={styles.imageContainer2}>
```

```
<View style={styles.onChangeImage}>
```

```
{image && <Image source={{ uri: image.uri }} style={{ width: "100%", height: "100%" }} />}
```

```
</View>
```

20 Josias Lopo

```

<View style={styles.background}>

  <ProfileScreen/>

</View>

</View>

</View>

<View style={styles.cameraImg}>

  <TouchableOpacity onPress={pickImage}>

    <Image source={camera} style={styles.cameraImg2} />

  </TouchableOpacity>

</View>

</View>

<View style={styles.content}>

  <Text style={styles.emailTxt}>E-mail:</Text>

  <Text style={styles.userTxt}>{userEmail}</Text>

</View>

</View>

);

};

```

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

```
  container: {
```

```
    flex: 1,
```

```
    backgroundColor: '#1D1E26',
```

```
    justifyContent: 'center',
```

```
    alignItems: 'center',
```

```
  },
```

```
  header: {
```

```
    flex: 1,
```

```
    flexDirection: 'row',
```

```
    width: '80%',
```

```
    justifyContent: "space-between",
```

```
    alignItems: 'flex-start',
```

```
    paddingTop: responsiveWidth(4)
```

```
  },
```

```
  btnArrow: {
```

```
    justifyContent: "center",
```

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

```
    backgroundColor: "#101014",
```

21 Josias Lopo

```
width: responsiveWidth(12),  
  
height: responsiveWidth(12),  
  
resizeMode: "contain",  
  
borderRadius: 100,  
  
alignSelf: 'flex-start'  
  
},  
  
btnArrowImg: {  
  
  width: responsiveWidth(5),  
  
  resizeMode: "contain",  
  
},  
  
updateBtn: {  
  
  color: '#1563FF',  
  
  fontFamily: 'MPLUS1pBold',  
  
  fontSize: responsiveFontSize(2),  
  
  padding: 10,  
  
  marginTop: responsiveWidth(1),  
  
  borderRadius: 10,  
  
  overflow: 'hidden'  
  
},
```

```
profilePic: {  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  marginTop: responsiveWidth(10),  
  
},
```

```
imageContainer: {  
  
  padding: 5,  
  
  width: responsiveWidth(62),  
  
  height: responsiveWidth(62),  
  
  borderRadius: 200,  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  backgroundColor: '#525468',  
  
  overflow: 'hidden',  
  
  alignSelf: 'center',  
  
  zIndex: 1,  
  
},
```

```
imageContainer2: {  
  
  padding: 4,
```

21 Josias Lopo

```
width: responsiveWidth(60.5),  
height: responsiveWidth(60.5),  
borderRadius: 200,  
justifyContent: 'center',  
alignItems: 'center',  
backgroundColor: '#1D1E26',  
overflow: 'hidden'  
},
```

```
onChangelImage: {  
width: '100%',  
height: '100%',  
borderRadius: 200,  
position: 'absolute',  
zIndex: 2,  
overflow: "hidden",  
},
```

```
background: {  
height: '100%',
```

```
width: '100%',  
  
resizeMode: "cover",  
  
overflow: "hidden",  
  
borderRadius: 200,  
  
position: 'absolute',  
  
zIndex: 1,  
  
},  
  
cameraImg: {  
  
width: responsiveWidth(10),  
  
height: responsiveWidth(10),  
  
zIndex: 2,  
  
marginLeft: responsiveWidth(52),  
  
position: 'relative',  
  
bottom: responsiveWidth(9),  
  
alignSelf: 'center',  
  
},  
  
cameraImg2: {  
  
width: responsiveWidth(10),  
  
height: responsiveWidth(10),  
  
},
```

21 Josias Lopo

```
content: {  
  
  flex: 1.5,  
  
  justifyContent: 'flex-start',  
  
  alignItems: 'center',  
  
},  
  
emailTxt: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1pBold',  
  
  fontSize: responsiveFontSize(2),  
  
  overflow: 'hidden'  
  
},  
  
userTxt: {  
  
  color: '#818181',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(2),  
  
  overflow: 'hidden'  
  
}  
  
});
```

export default Profile;

3.1.22 Regist.js

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

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

import { useFonts } from 'expo-font';

import { responsiveFontSize, responsiveHeight, responsiveWidth } from "react-native-responsive-
dimensions";

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

import Seta2 from "../src/assets/images/arrowLeft.png";

import GLogo from "../src/assets/images/googlelogo.png";

import { createUserWithEmailAndPassword } from "firebase/auth";

import { Auth, auth } from '../config/firebase';

export default function Regist({ navigation }) {

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [errorMessage, setErrorMessage] = useState("");
```

```
const handleSubmit = async () => {  
  
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
  
  if (!email || !password) {  
  
    setErrorMessage('Please enter your email and password.');  } else if (!emailRegex.test(email)) {  
  
    setErrorMessage('Please enter a valid email address.');  } else if (password.length < 6) {  
  
    setErrorMessage('Password should be at least 6 characters long.');  } else {  
  
    try {  
  
      await createUserWithEmailAndPassword(auth, email, password);  
  
      setErrorMessage('');  
  
      navigation.navigate("Login");  
  
    } catch (err) {  
  
      setErrorMessage(err.message);  
  
    }  
  
  }  
  
};
```

```
const insets = useSafeAreaInsets();

const [loaded] = useFonts({
  NanumMyeongjo: require('../src/assets/fonts/NanumMyeongjo-Regular.ttf'),
  NanumMyeongjoBold: require('../src/assets/fonts/NanumMyeongjo-Bold.ttf'),
  NanumMyeongjoEBold: require('../src/assets/fonts/NanumMyeongjo-ExtraBold.ttf'),
  Inter: require('../src/assets/fonts/Inter-Regular.ttf'),
  InterBold: require('../src/assets/fonts/Inter-Bold.ttf'),
  InterEBold: require('../src/assets/fonts/Inter-ExtraBold.ttf'),
  MPLUS1p: require('../src/assets/fonts/MPLUS1p-Regular.ttf'),
  MPLUS1pBold: require('../src/assets/fonts/MPLUS1p-Bold.ttf'),
  MPLUS1pEBold: require('../src/assets/fonts/MPLUS1p-ExtraBold.ttf'),
});

if (!loaded){
  return null;
}

return (
```

```
<View style={{styles.container,{paddingTop: insets.top + responsiveWidth(5), paddingBottom: insets.bottom, paddingLeft: insets.left + responsiveWidth(3), paddingRight: insets.left + responsiveWidth(3)}}>
```

```
<TouchableOpacity style={styles.btnBack} onPress={() => navigation.navigate("Hall")}>
```

```
<View style={styles.btnArrow}>
```

```
<Image source = {Seta2} style={styles.btnArrowImg}/>
```

```
</View>
```

```
<Text style={{styles.txtLoginBtn, {marginLeft: responsiveWidth(3)}}>Register</Text>
```

```
</TouchableOpacity>
```

```
<View style={styles.loginContent}>
```

```
<View style={styles.logoContent}>
```

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

```
Eventer
```

```
</Text>
```

```
<Text style={styles.logoDot}>.</Text>
```

```
</View>
```

```
<View style={styles.userData}>

<View style={styles.Name}>

  <Text style={styles.userNamePassword}>E-mail address:</Text>

  <TextInput

    placeholder=""

    value={email}

    onChangeText={value => setEmail(value)}

    style = {styles.userInput}

  />

</View>

<View style={styles.Password}>

  <Text style={styles.userNamePassword}>Password:</Text>

  <TextInput

    placeholder=""

    style={styles.userInput}

    value={password}

    onChangeText={value => setPassword(value)}

    secureTextEntry={true}

  />
```

```
</View>
```

```
{errorMessage !== "" && (  
  <Text style={styles.errorMessage}>{errorMessage}</Text>  
)}
```

```
</View>
```

```
<View style={styles.googleRegisBtn}>  
  <View style={styles.Login}>  
    <TouchableOpacity  
      onPress={handleSubmit}>  
      <View style={styles.btnArrowRotate}>  
        <Image source = {Seta2} style={styles.btnArrowImg}/>  
      </View>  
    </TouchableOpacity>  
  </View>  
  <TouchableOpacity style={styles.regisBtn} onPress={() => navigation.navigate("Login")}>  
    <Text style={styles.regisBtnTxt}>Have an account? </Text>  
    <Text style={styles.regisBtnLink}>Login</Text>
```

```
</TouchableOpacity>

</View>

</View>

</View>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#101014',

  },

  btnBack:{

    width: responsiveWidth(25),

    flexDirection: "row",
```

```
alignItems: "center",  
  
//backgroundColor: 'black',  
  
},
```

```
btnArrow: {  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
  backgroundColor: 'rgba(29, 30, 38, 0.76)',  
  
  width: responsiveWidth(10),  
  
  height: responsiveWidth(10),  
  
  resizeMode: "contain",  
  
  //borderRadius: '100%',  
  
  borderRadius: 100,  
  
},
```

```
btnArrowImg: {  
  
  width: responsiveWidth(3.5),  
  
  resizeMode: "contain",  
  
},
```

```
btnArrowRotate: {
```

```
justifyContent: 'center',
alignItems: 'center',
backgroundColor: 'rgba(29, 30, 38, 0.76)',
width: responsiveWidth(13),
height: responsiveWidth(13),
resizeMode: "contain",
//borderRadius: '100%',
borderRadius: 100,
transform: [{ rotate: '180deg' }],
},
```

```
txtLoginBtn: {
  color: 'white',
  fontSize: responsiveFontSize(2),
  fontFamily: 'MPLUS1p',
},
```

```
loginContent: {
  flex: 1,
```

22 Josias Lopo

```
//backgroundColor: 'black',  
  
justifyContent: 'center',  
  
alignItems: 'center',  
  
},
```

```
logoContent: {  
  
  flex: 1,  
  
  flexDirection: 'row',  
  
  alignItems: 'center',  
  
},
```

```
logo: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1pEBold',  
  
  fontSize: responsiveFontSize(6),  
  
},
```

```
logoDot: {  
  
  color: '#1563FF',  
  
  fontFamily: 'MPLUS1pEBold',  
  
  fontSize: responsiveFontSize(6),
```

},

```
userData: {  
  
  //backgroundColor: 'gray',  
  
  flex: 1,  
  
  width: responsiveHeight(33),  
  
},
```

```
Name:{  
  
  flex: 1,  
  
  justifyContent: 'center',  
  
},
```

```
Password:{  
  
  flex: 1,  
  
  justifyContent: 'center',  
  
},
```

```
userNamePassword: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1p',
```

22 Josias Lopo

```
fontSize: responsiveFontSize(2.3),  
  
},  
  
userInput: {  
  
  color: 'white',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(2),  
  
  borderBottomColor: '#525468',  
  
  borderBottomWidth: responsiveWidth(0.10),  
  
},  
  
Login: {  
  
  paddingTop: responsiveHeight(3),  
  
  flex: 0.70,  
  
  flexDirection: 'row',  
  
  justifyContent: 'center',  
  
  alignItems: 'center',  
  
},  
  
googleRegisBtn: {  
  
  flex: 1.3,  
  
  alignItems: 'center',
```

},

```
regisBtn: {  
  justifyContent: 'center',  
  alignItems: 'center',  
  height: responsiveHeight(3.5),  
  //borderRadius: responsiveWidth(4)  
  borderRadius: 100,  
  flexDirection: 'row',  
},
```

```
regisBtnTxt: {  
  color: 'white',  
  fontFamily: 'MPLUS1p',  
  
},
```

```
regisBtnLink: {  
  color: '#1563FF',  
  fontFamily: 'MPLUS1p',  
  
},
```

22 Josias Lopo

```
errorMessage: {  
  
  color: '#EA4335',  
  
  fontFamily: 'MPLUS1p',  
  
  fontSize: responsiveFontSize(1.8),  
  
  marginTop: responsiveHeight(1),  
  
},  
  
});
```

3.1.23 Todo.js

```
import React from "react";  
  
import { View, StyleSheet } from 'react-native';  
  
import { useSafeAreaInsets } from 'react-native-safe-area-context';  
  
import { Header, createStackNavigator } from '@react-navigation/stack';  
  
import List from './List'; // Import your List component here  
  
import { Text } from 'react-native';  
  
import { responsiveFontSize, responsiveWidth } from "react-native-responsive-dimensions";  
  
  
const Stack = createStackNavigator();
```

```
export default function Default() {  
  
  const insets = useSafeAreaInsets();  
  
  return (  
  
    <View style={styles.container,{ paddingBottom: insets.bottom, }}>  
  
      <View style={{paddingTop: insets.top + responsiveWidth(3), padding: responsiveWidth(5),  
backgroundColor:"#1D1E26"}}>  
  
        <Text style={styles.txt1}>Eventer</Text>  
  
        <Text style={styles.txt2}>TODO LIST</Text>  
  
      </View>  
  
      <Stack.Navigator screenOptions={{headerShown: false}}>  
  
        <Stack.Screen name="MyTodos" component={List} />  
  
      </Stack.Navigator>  
  
    </View>  
  
  );  
}
```

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

```
backgroundColor: '#101014',

},

txt1: {

  fontSize: responsiveFontSize(6),

  color: '#fff',

  fontFamily: 'NanumMyeongjo',

},

txt2:{

  paddingTop: responsiveWidth(1),

  fontSize: responsiveFontSize(2),

  color: '#fff',

  fontFamily: 'MPLUS1p',

},

});
```

3.1.24 ProfilePic.js

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

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

import { firebase } from '../config/firebase';
```

```
import { getAuth } from 'firebase/auth';

import { getStorage, ref, getDownloadURL } from 'firebase/storage';

import { responsiveHeight, responsiveWidth } from 'react-native-responsive-dimensions';

import account from '../src/assets/images/account.png';

const ProfileScreen = () => {

  const [profilePic, setProfilePic] = useState(null);

  const auth = getAuth();

  useEffect(() => {

    const currentUser = auth.currentUser;

    if (currentUser) {

      const uid = currentUser.uid;

      const storage = getStorage(firebase);

      const storageRef = ref(storage, `profilePic/${uid}/profilePic`);

      const fetchProfilePic = () => {

        getDownloadURL(storageRef)

          .then(url => {

            setProfilePic(url);
```

```

    })

    .catch(error => {

        console.error('Error fetching image:', error);

    });

};

fetchProfilePic();

const interval = setInterval(fetchProfilePic, 10000);

return () => clearInterval(interval);

}

}, []);

return (

<View style={styles.container}>

    <View style={styles.imageContainer}>

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

        {profilePic && <Image source={{ uri: profilePic }} style={styles.profilePic} />}

    </View>

</View>

```

```
</View>

</View>

);

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

    alignItems: 'center',

    justifyContent: 'center',

  },

  imageContainer: {

    width: responsiveWidth(12),

    height: responsiveWidth(12),

    position: 'relative',

    alignItems: 'center',

    justifyContent: 'center'

  },

  image: {

    width: responsiveWidth(7),

    height: responsiveWidth(7),
```

23 Josias Lopo

```
      resizeMode: 'contain',  
  
      position: 'absolute',  
  
      zIndex: 1,  
  
    },  
  
    profilePic: {  
  
      width: '100%',  
  
      height: '100%',  
  
      resizeMode: 'cover',  
  
      position: 'absolute',  
  
      zIndex: 2,  
  
    },  
  
  });  
  
export default ProfileScreen;
```

3.1.25 ProfilePic2.js

```
import React, { useEffect, useState } from 'react';  
  
import { View, Image, StyleSheet } from 'react-native';  
  
import { firebase } from '../config/firebase';  
  
import { getAuth } from 'firebase/auth';  
  
import { getStorage, ref, getDownloadURL } from 'firebase/storage';  
  
import { responsiveHeight, responsiveWidth } from 'react-native-responsive-dimensions';
```

Josias Lopo 235

```
import account from '../src/assets/images/account.png';

const ProfileScreen = () => {

  const [profilePic, setProfilePic] = useState(null);

  const auth = getAuth();

  useEffect(() => {

    const currentUser = auth.currentUser;

    if (currentUser) {

      const uid = currentUser.uid;

      const storage = getStorage(firebase);

      const storageRef = ref(storage, `profilePic/${uid}/profilePic`);

      const fetchProfilePic = () => {

        getDownloadURL(storageRef)

          .then(url => {

            setProfilePic(url);

          })

          .catch(error => {

            console.error('Error fetching image:', error);
```

```
});  
  
};  
  
fetchProfilePic();  
  
const interval = setInterval(fetchProfilePic, 10000);  
  
return () => clearInterval(interval);  
}  
}, []);  
  
return (  
  <View style={styles.container}>  
    <View style={styles.imageContainer}>  
      <Image source={account} style={styles.image} />  
      {profilePic && <Image source={{ uri: profilePic }} style={styles.profilePic} />}  
    </View>  
  </View>  
);
```

};

```
const styles = StyleSheet.create({  
  
  container: {  
  
    flex: 1,  
  
    alignItems: 'center',  
  
    justifyContent: 'center',  
  
  },  
  
  imageContainer: {  
  
    width: responsiveWidth(62),  
  
    height: responsiveWidth(62),  
  
    position: 'relative',  
  
    alignItems: 'center',  
  
    justifyContent: 'center'  
  
  },  
  
  image: {  
  
    width: responsiveWidth(50),  
  
    height: responsiveWidth(50),  
  
    resizeMode: 'contain',  
  
    position: 'absolute',  
  
    zIndex: 1,  
  
  },  
});
```

23 Josias Lopo

```
profilePic: {  
  
  width: '100%',  
  
  height: '100%',  
  
  resizeMode: 'cover',  
  
  position: 'absolute',  
  
  zIndex: 2,  
  
},  
  
});  
  
export default ProfileScreen;
```

3.1.26 .gitignore

Learn more <https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files>

dependencies

node_modules/

Expo

.expo/

dist/

web-build/

Native

.orig.

*.jks

*.p8

*.p12

*.key

*.mobileprovision

Metro

.metro-health-check*

debug

npm-debug.*

yarn-debug.*

yarn-error.*

macOS

24 Josias Lopo

.DS_Store

*.pem

local env files

.env*.local

typescript

*.tsbuildinfo

3.1.27 App.json

```
{  
  "expo": {  
    "plugins": [  
      [  
        "expo-image-picker",  
        {  
          "photosPermission": "The app allows you to upload photos from your library",  
          "cameraPermission": "The app needs permission to access your camera"  
        }  
      ]  
    ],  
    "name": "eventer",
```

```
"slug": "eventer",  
  
"version": "1.0.0",  
  
"orientation": "portrait",  
  
"icon": "./assets/icon.png",  
  
"userInterfaceStyle": "light",  
  
"splash": {  
  
  "image": "./assets/splash.png",  
  
  "resizeMode": "contain",  
  
  "backgroundColor": "#101014"  
},  
  
"assetBundlePatterns": ["**/*"],  
  
"packagerOpts": {  
  
  "config": "metro.config.js",  
  
  "sourceExts": [  
  
    "expo.ts",  
  
    "expo.tsx",  
  
    "expo.js",  
  
    "expo.jsx",  
  
    "ts",  
  
    "tsx",  
  
    "js",  
  
    "jsx",
```

```
"json",  
  
"wasm",  
  
"svg"  
]  
  
},  
  
"ios": {  
  
  "supportsTablet": true  
  
},  
  
"android": {  
  
  "adaptiveIcon": {  
  
    "foregroundImage": "./assets/adaptive-icon.png",  
  
    "backgroundColor": "#101014"  
  
  }  
  
},  
  
"web": {  
  
  "favicon": "./assets/favicon.png"  
  
}  
  
}  
  
}
```

3.1.28 babel.config.js

```
module.exports = function(api) {
```

```
api.cache(true);  
  
return {  
  
  presets: ['babel-preset-expo'],  
  
};  
  
};
```

3.1.29 metro.config.js

```
const { getDefaultConfig } = require('@expo/metro-config');  
  
const defaultConfig = getDefaultConfig(__dirname);  
defaultConfig.resolver.assetExts.push('cjs');  
  
module.exports = defaultConfig;
```

3.1.30 package.json

```
{  
  
  "name": "eventer",
```

24 Josias Lopo

```
"version": "1.0.0",

"main": "node_modules/expo/AppEntry.js",

"rnpm": {

  "assets": [

    "./assets/fonts"

  ]

},

"scripts": {

  "start": "expo start",

  "android": "expo start --android",

  "ios": "expo start --ios",

  "web": "expo start --web"

},

"dependencies": {

  "@expo/webpack-config": "~19.0.1",

  "@fontsource/nanum-myeongjo": "^5.0.16",

  "@react-native-async-storage/async-storage": "1.23.1",

  "@react-native-firebase/app": "^19.2.2",

  "@react-native-firebase/storage": "^19.2.2",

  "@react-navigation/bottom-tabs": "^6.5.11",

  "@react-navigation/drawer": "^6.6.15",

  "@react-navigation/material-top-tabs": "^6.6.5",
```

```
"@react-navigation/native": "^6.1.10",  
"@react-navigation/native-stack": "^6.9.17",  
"@react-navigation/stack": "^6.3.20",  
"@types/react": "~18.2.45",  
"async-storage": "^0.1.0",  
"axios": "^1.6.8",  
"expo": "^51.0.0",  
"expo-av": "~14.0.3",  
"expo-font": "~12.0.4",  
"expo-image-picker": "~15.0.4",  
"expo-permissions": "^14.4.0",  
"expo-speech": "~12.0.1",  
"expo-status-bar": "~1.12.1",  
"firebase": "^10.6.0",  
"openai-api": "^1.3.1",  
"react": "18.2.0",  
"react-dom": "18.2.0",  
"react-native": "0.74.1",  
"react-native-collapsible": "^1.6.1",  
"react-native-dotenv": "^3.4.11",  
"react-native-drawer-layout": "^3.3.0",  
"react-native-elements": "^3.4.3",
```

```
"react-native-gesture-handler": "~2.16.1",  
"react-native-image-picker": "^7.1.2",  
"react-native-modal": "^13.0.1",  
"react-native-paper": "^5.12.3",  
"react-native-reanimated": "3.10.0",  
"react-native-responsive-dimensions": "^3.1.1",  
"react-native-safe-area-context": "4.10.1",  
"react-native-screens": "3.31.1",  
"react-native-side-menu": "^1.1.3",  
"react-native-tab-view": "^3.5.2",  
"react-native-vector-icons": "^10.0.3",  
"react-native-video": "^5.2.1",  
"react-native-web": "~0.19.6",  
"typescript": "~5.3.3",  
"video.js": "^8.12.0"  
},  
"devDependencies": {  
  "@babel/core": "^7.24.0",  
  "@types/react-native-vector-icons": "^6.4.18",  
  "react-native-svg-transformer": "^1.1.0"  
},  
"private": true
```

```
}
```

3.1.31 react-native.config.js

```
module.exports = {  
  
  assets: ['./assets/fonts'], };
```

3.2 Código Website

3.2.1 Index.html

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
  <meta charset="UTF-8">  
  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
  <link rel="preconnect" href="https://fonts.googleapis.com">  
  
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
  
  <link                href="https://fonts.googleapis.com/css2?family=M+PLUS+1p&display=swap"  
rel="stylesheet">  
  
  
  <title>Eventer</title>  
  
</head>
```

```
<script>

// Check if the screen width is smaller than a threshold (e.g., 768px)

if (screen.width < 900) {

// Redirect to the mobile version

window.location.href = "/mobile.html";

}

</script>
```

```
<style>

body {

background-color: #101014;

display: flex; /* Define o body como um container flexível */

flex: 1; /* Permite que o body cresça para preencher a tela */

flex-direction: column; /* Change to column layout */

}

.logo {

flex: 1.2; /* Permite que a div cresça para preencher o espaço disponível */

align-content: flex-end;

width: 100%; /* Garante que a div ocupe toda a largura do body */

font-family: "M PLUS 1p", sans-serif;
```

```
font-weight: 900;

font-style: normal;

}

h1 {

  color: #f0f0f0;

  text-align: center;

  font-size: 300%;

}

.contentRow {

  flex: 6; /* Permite que a div cresça para preencher o espaço disponível */

  display: flex; /* Define a div como um container flexível */

  flex-direction: row; /* Define a direção dos elementos filhos */

  padding-left: 100px;

  padding-right: 100px;

}

.container , .container2, .container3 {

  display: flex;

  width: 100%;
```

```
margin: 1px;  
  
font-size: 400%;  
  
color: white;  
  
font-family: "M PLUS 1p ", sans-serif;  
  
font-style: normal;  
  
}
```

```
.text1 {  
  
font-size: 40%;  
  
color: #525468;  
  
font-family: "M PLUS 1p ", sans-serif;  
  
font-style: normal;  
  
font-weight: 700;  
  
}
```

```
.container a {  
  
display: block;  
  
text-align: left;  
  
margin-bottom: 3%;  
  
}
```

```
.container div {  
  
    display: flex;  
  
    flex-direction: column;  
  
    justify-content: center;  
  
    height: 100%;  
  
    gap: 30px;  
  
    margin-left: 10%;  
  
}  
  
.container img {  
  
    display: block; /* Garante que as imagens fiquem uma abaixo da outra */  
  
}  
  
.container3 img {  
  
    display: block; /* Garante que a imagem do mockup seja um bloco */  
  
    margin: auto; /* Centraliza horizontalmente */  
  
    align-self: center; /* Centraliza verticalmente */  
  
}  
  
.container3 {  
  
    display: flex; /* Garante que a imagem do mockup seja um bloco */
```

```
margin: auto; /* Centraliza horizontalmente */

justify-content: center;

align-items: center;

}

.container2 {

  /* ... other styles ... */

  display: flex;          /* Enable flexbox for positioning */

  flex-direction: column; /* Arrange items vertically */

  justify-content: flex-end; /* Align items to the bottom */

  align-items: center;    /* Center items horizontally */

}

.container2 div {

  display: flex;

  justify-content: center;

  gap: 10px; /* Espaço entre os elementos internos */

  margin-top: 20px;

}

</style>

<body>
```

```
<div class="logo">

  <h1>Eventer<a style="color: #1563FF;">.</a></h1>

</div>

<div class="contentRow">

  <div class="container">

    <div>

      <text>

        <a class="text1">Powered by <text style="color: #1563FF">AI</text></a>

        <a>Download <br />now</a>

      </text>

    </div>

  </div>

</div>

<div class="container2">

  <div>

  </div>

</div>
```

```


</div>

</div>

<div class="container3">

  <div>

  </div>

</div>

</div>

</body>

</html>
```

3.2.2 Mobile.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link                href="https://fonts.googleapis.com/css2?family=M+PLUS+1p&display=swap"
  rel="stylesheet">

  <title>Eventer</title>

</head>

<script>

  // Check if the screen width is smaller than a threshold (e.g., 768px)

  if (screen.width > 900) {

    // Redirect to the mobile version

    window.location.href = "/index.html";

  }

</script>

<style>

  body {

    background-color: #101014;
```

```
display: flex;  
  
flex-direction: column;  
  
padding: 20px;  
  
height: auto; /* Allow body to grow with content */  
  
}
```

```
.logo {  
  
  flex: 1;  
  
  width: 100%;  
  
  font-family: "M PLUS 1p", sans-serif;  
  
  text-align: center;  
  
  font-weight: 900;  
  
}
```

```
h1 {  
  
  color: #f0f0f0;  
  
  font-size: 2em;  
  
}
```

```
.contentRow {  
  
  display: flex;  
  
  flex-direction: column;
```

```
gap: 20px;

}

.container, .container2, .container3 {

width: 100%;

text-align: center;

color: white;

font-family: "M PLUS 1p", sans-serif;

font-weight: 500;

}

.container3 {

margin-bottom: 10%;

margin-top: 10%;

}

.text {

margin-bottom: 15%;

}

.text1 {

font-size: 1.5em;
```

25 Josias Lopo

```
color: #525468;

}

.text2 {

font-size: 3em;

}

.container2 {

display: flex;

align-items: center;

justify-content: space-evenly;

}

.buttons {

margin-left: 0%;

}

.container2 div {

align-items: center;

display: flex;

gap: 15%;
```

}

/* Make images responsive using percentages */

```
.container img {  
  
  max-width: 44%;  
  
  margin: 5px;  
  
  margin-bottom: 10%;  
  
}
```

```
.container3 img {  
  
  max-width: 50%; /* Allow image to take up full width if needed */  
  
  height: auto;  
  
}
```

/* Media Query for larger screens (adjust as needed) */

```
@media (min-width: 768px) {  
  
  h1 { font-size: 3em; }  
  
  .text1 { font-size: 2em; }  
  
  .text2 { font-size: 3.3em; }  
  
  .container3 img { width: 300px; }
```

}

</style>

<body>

<div class="logo">

<h1>Eventer.</h1>

</div>

<div>

<div class="contentRow">

<div class="container3">

</div>

<div class="container">

<div style="margin-bottom: 10%;">

Powered by <text style="color: #1563FF">AI
</text>

Download Now

</div>

<div style="display: flex; justify-content: center;">


```


</div>

</div>

<div class="container2">

  <div>

  </div>

  <div class="buttons">

  </div>

</div>

</div>

</div>

</body>

</html>
```

3.2.3 Redirect.js

```
function checkAndRedirect() {

    // Check if the screen width is smaller than a threshold (e.g., 900px)

    if (window.innerWidth < 900) {

        // Redirect to the mobile version if not already there

        if (!window.location.href.includes("/mobile.html")) {

            window.location.href = "/mobile.html";

        }

    } else {

        // Redirect to the desktop version if not already there

        if (!window.location.href.includes("/index.html")) {

            window.location.href = "/index.html";

        }

    }

}

// Executar a função quando a página é carregada

checkAndRedirect();

// Executar a função apenas após um pequeno atraso para evitar chamar muitas vezes

let resizeTimer;
```

```
window.addEventListener('resize', function() {  
  
  clearTimeout(resizeTimer);  
  
  resizeTimer = setTimeout(checkAndRedirect, 250);  
  
});
```

4. Outros projetos

4.1 NOKAUTE.pt

Nokate.pt é o primeiro website realizado por mim para venda, tem como objetivo a divulgação dos ginásios *NOKAUTE* e tabelas de aulas e preços.



4.2 Innovationpro.pt

Durante o segundo ano da EPB realizei um projeto juntamente com mais dois colegas sobre a promoção de espaços mais limpos, supervisionado pelo professor António Araújo da área da programação fizemos um website com uma tecnologia nova (*webflow*) moderno e responsivo.

4.3 jogOleo

Durante o segundo ano da EPB eu e o meu colega José Rebelo realizamos um jogo para *Windows* em *c#*, com inteligência artificial muito simples e com base a promover o demasiado gasto de óleo no mundo. Foi supervisionado pelo professor Horácio Magalhães da área de Integração e realizado um dia para ser jogado na escola por algumas turmas.

4.4 Davdsm.pt

Durante o estágio do segundo ano, desenvolvi para a empresa *davdsm.pt* as antigas versões dos websites *new-vintage.pt* e *at-hometexteis.pt*, fora do estágio ajudei a desenvolver algumas zonas para o website *thehighjump.co*.



Conclusão

Com a finalização deste projeto deixo fui deixado com uma marca grande de desafios, foi além das expectativas que tinha e consegui desenvolver competências para aprender, resolver e enfrentar problemas e frustrações.

Nestes 3 anos de percurso escolar da EPB só tenho a agradecer a tudo o que aprendi, ao meio que fui submetido, aos professores que me ajudaram muito na compreensão do aprender e do como ser uma pessoa melhor, desde na realização de projetos até relacionamentos pessoais, com isto consegui desenvolver muito as minhas *soft skills*. Participei em muitas atividades escolares desde atividades da minha área técnica a atividades que não tinham nenhuma ligação isso fez-me conhecer muitas pessoas de outros cursos e pessoas do mundo do trabalho.

Infelizmente devido a escolha de um projeto muito grande juntamente com tecnologias muito recentes não consegui realizar o meu projeto da PAP como queria, mas vou manter o projeto *open-source* para quem quiser aproveitar e vou realizar atualizações para melhor experiência de utilizador. O meu objetivo do que vou fazer na finalização deste curso é continuar o meu percurso letivo seguindo a licenciatura em engenharia de sistemas informáticos e abrindo uma empresa de programação e *design*.

Bibliografia

Webgrafia

React Native - <https://radixweb.com/blog/flutter-vs-react-native>

Expo - <https://docs.expo.dev/tutorial/create-your-first-app/>

Android Studio - <https://developer.android.com/studio/run/emulator>

Firebase - <https://rockcontent.com/br/blog/firebase/>

Xcode - <https://developer.apple.com/documentation/xcode/running-your-app-in-simulator-or-on-a-device>

VS Code e Git - <https://code.visualstudio.com/docs/sourcecontrol/intro-to-git>

Firebase e expo - <https://docs.expo.dev/guides/using-firebase/>

Apps responsivas React Native: <https://www.browserstack.com/guide/how-to-make-react-native-app-responsive>

ChatGpt API: <https://platform.openai.com/docs/api-reference/making-requests>

API Gemini: <https://ai.google.dev/gemini-api/docs/api-overview>

Anexos

Anexo 1 – Curriculum Vitae


europass



Josias de Araújo Lopo

Date of birth: 16/09/2005 | **Nationality:** Portuguese | **Gender:** Male |

Phone number: (+351) 932611168 (Mobile) | **Email address:**
josiaslopo2005@gmail.com |

Address: R. S. João Batista , Nr 945, 2ºDrt, 4805-676, Brito - Guimarães,
Portugal (Home)

● **WORK EXPERIENCE**

16/09/2021 – CURRENT Portugal
STUDENT

● **EDUCATION AND TRAINING**

16/09/2021 – CURRENT Braga, Portugal
**STUDENT 12º GRADE - PROFESSIONAL COURSE FOR MANAGEMENT AND PROGRAMMING ON
COMPUTER SYSTEMS** Escola Profissional de Braga - EPB

Address R. Augusto Veloso 140, 4705-082, Braga, Portugal | **Website** epb.pt

MICROSOFT OFFICE SPECIALIST Certiport

05/06/2023 – 25/07/2023 Guimarães, Portugal
INTERNSHIP - DAVDSM - WEBSITE BUILDER DAVDSM

Website <https://davdsm.pt/>

● **LANGUAGE SKILLS**

Mother tongue(s): **PORTUGUESE**

Other language(s):

	UNDERSTANDING		SPEAKING		WRITING
	Listening	Reading	Spoken production	Spoken interaction	
ENGLISH	B2	B2	B1	B1	B1

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

● **DIGITAL SKILLS**

Wordpress | CMD | Windows Server | Linux | MacOS | GitHub | iTerm | Microsoft Office | Gmail |
Use of internet browser | google | social media | team player | easy learner on new software | MS
Windows | Computer knowledge from user point of view | skype | zoom | whatsapp | figma |
expo | Microsoft Teams | outlook | BandLab | Photoshop And Figma

● **DRIVING LICENCE**

Driving Licence: B1
Driving Licence: B

Josias Lopo 267

• PROJECTS

CURRENT

Eventer - Event App

App for Event organization, post sharing and chat.
IOS and Android

• HOBBIES AND INTERESTS

Hobbies

Musician (singer, guitar, bass and drums player).
Photographer, Designer, amateur bodybuilder, gamer and tech enthusiast (hardware and software)

• CREATIVE WORKS

CURRENT

Website and design freelancer

Start building websites and post my first in 2023 (Innovationpro.pt, Nokaute.pt)
In progress (Josias.pt, Oficinacrista.pt and eugenioandrade.pt)

• COMMUNICATION AND INTERPERSONAL SKILLS

Communication and interpersonal skills

1. Respectful;
2. Assiduous;
3. Great Communicator;
4. Leadership;
5. Empathy;
6. Creativity;
7. Teamwork;
8. Proactivity;
9. Work Ethic;
10. Critical thoughts
11. Positive Attitude;

• VOLUNTEERING

19/02/2022 – 27/02/2022 Altice Arena Fórum, Braga, Portuga

EMACI2022 - Bragaades

Write here the descI served as a translator;
Help people to reach their destiny;
Work at the athlete courses, organizing competition equipment.

04/11/2022 – 03/03/2023

Senior Formator

Senior trainer in MS Office, E-mail, Web browsing, Cybersecurity, Computer History;

• PROGRAMMING SKILLS

Programming Languages | Technologies and Frameworks

Basic React Native
Python
C#, windows forms
html and css
My Sql
Basic Php
React Native
Expo
Firebase