**Passo a Passo para Desenvolver o Aplicativo de Chat**

**1. Iniciar o Projeto com Expo**

Primeiro, você inicia o projeto utilizando o Expo. Abra seu terminal e execute os seguintes comandos:

npx create-expo-app chat --template blank

cd chat

npx expo install firebase

Esses comandos criam uma nova aplicação Expo chamada "chat" e instalam a biblioteca Firebase.

**2. Configurar o Firebase**

Crie um arquivo chamado firebaseConfig.js na raiz do seu projeto para configurar o Firebase. Insira o seguinte código, substituindo os dados de configuração pelo que você obteve no console do Firebase:

import { initializeApp } from 'firebase/app';

import { getFirestore } from 'firebase/firestore';

import { getAuth } from 'firebase/auth';

import { getStorage } from 'firebase/storage';

const firebaseConfig = {

apiKey: "Sua\_API\_Key",

authDomain: "Seu\_Domain",

projectId: "Seu\_Project\_ID",

storageBucket: "Seu\_Storage\_Bucket",

messagingSenderId: "Seu\_Sender\_ID",

appId: "Seu\_App\_ID"

};

// Inicializando Firebase

const app = initializeApp(firebaseConfig);

const db = getFirestore(app);

const auth = getAuth(app);

export const storage = getStorage(app);

export { db, auth };

**3. Criar as Telas de Login e Chat**

Crie uma pasta chamada screens e dentro dela crie dois arquivos: LoginScreen.js e ChatScreen.js.

**LoginScreen.js**

import React, { useState } from 'react';

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

export default function LoginScreen({ navigation }) {

const [name, setName] = useState('');

const handleLogin = () => {

navigation.navigate('Chat', { name });

};

return (

<View style={styles.container}>

<Text style={styles.title}>Bem-vindo ao Chat!</Text>

<TextInput

style={styles.input}

placeholder="Digite seu nome"

value={name}

onChangeText={setName}

/>

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

<Text style={styles.buttonText}>Entrar</Text>

</TouchableOpacity>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

justifyContent: 'center',

padding: 20,

backgroundColor: '#F5F5F5'

},

title: {

fontSize: 24,

fontWeight: 'bold',

textAlign: 'center',

marginBottom: 20,

color: '#35797d',

},

input: {

borderBottomWidth: 1,

marginBottom: 20,

padding: 10,

borderColor: '#35797d',

fontSize: 18,

},

button: {

backgroundColor: '#35797d',

padding: 10,

borderRadius: 5,

alignItems: 'center',

},

buttonText: {

color: '#fff',

fontSize: 18,

fontWeight: 'bold',

}

});

**ChatScreen.js**

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

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

import { db } from '../firebaseConfig';

import { collection, addDoc, onSnapshot, query, orderBy, serverTimestamp } from 'firebase/firestore';

export default function ChatScreen({ route }) {

const { name } = route.params;

const [message, setMessage] = useState('');

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

useEffect(() => {

const messagesQuery = query(

collection(db, 'messages'),

orderBy('createdAt', 'desc')

);

const unsubscribe = onSnapshot(messagesQuery, snapshot => {

setMessages(snapshot.docs.map(doc => ({

id: doc.id,

...doc.data()

})));

});

return unsubscribe;

}, []);

const handleSend = async () => {

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

text: message,

name,

createdAt: serverTimestamp()

});

setMessage('');

};

return (

<View style={styles.container}>

<FlatList

data={messages}

renderItem={({ item }) => (

<View style={styles.messageContainer}>

<Text style={styles.sender}>{item.name}:</Text>

<Text style={styles.message}>{item.text}</Text>

</View>

)}

keyExtractor={item => item.id}

inverted

/>

<View style={styles.inputContainer}>

<TextInput

style={styles.input}

value={message}

onChangeText={setMessage}

placeholder="Escreva uma mensagem"

/>

<Button title="Enviar" onPress={handleSend} color="#35797d" />

</View>

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

padding: 10,

backgroundColor: '#F5F5F5'

},

inputContainer: {

flexDirection: 'row',

alignItems: 'center',

marginTop: 10,

},

input: {

flex: 1,

borderBottomWidth: 1,

padding: 10,

borderColor: '#35797d',

marginRight: 10,

fontSize: 16,

},

messageContainer: {

paddingVertical: 5,

backgroundColor: '#E0E0E0',

padding: 10,

borderRadius: 10,

marginBottom: 5,

maxWidth: '80%',

alignSelf: 'flex-start',

},

sender: {

fontWeight: 'bold',

color: '#35797d',

marginBottom: 3,

},

message: {

fontSize: 16,

color: '#333',

},

});

**4. Configurar a Navegação**

No arquivo principal do seu aplicativo, que pode ser chamado App.js, você configurará a navegação entre a tela de login e a tela de chat:

import React from 'react';

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

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

import ChatScreen from './screens/ChatScreen';

import LoginScreen from './screens/LoginScreen';

const Stack = createStackNavigator();

export default function App() {

return (

<NavigationContainer>

<Stack.Navigator>

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

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

</Stack.Navigator>

</NavigationContainer>

);

}

**5. Instalar Dependências Necessárias**

Certifique-se de que todas as dependências necessárias estejam instaladas. O seu arquivo package.json já possui as bibliotecas do Firebase e navegação que você precisará.

**6. Executar o Aplicativo**

Finalmente, execute o aplicativo para ver seu chat em ação:

npx expo start

Você pode abrir o aplicativo no emulador, no seu dispositivo físico ou na web.