UNIVERSIDAD DON BOSCO

ESCUELA DE COMPUTACION

DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA

TITULO:

Guía #11

DOCENTE:

Ing. Emerson Francisco Cartagena Candelario

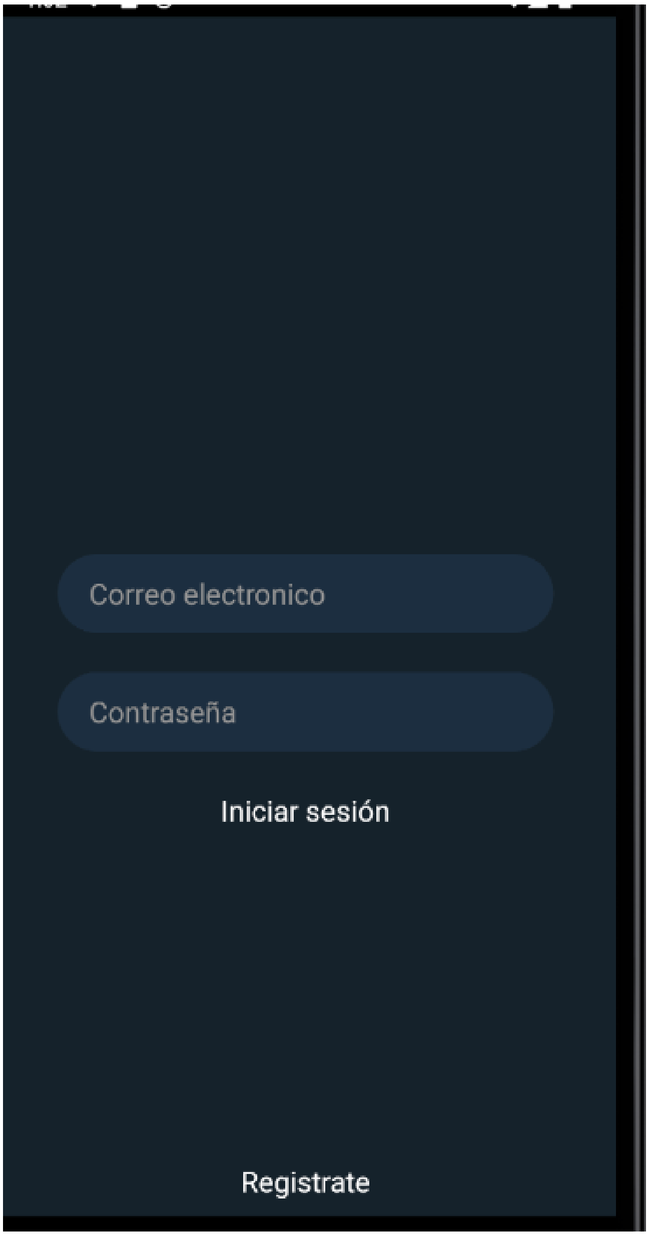
PRESENTADO POR:

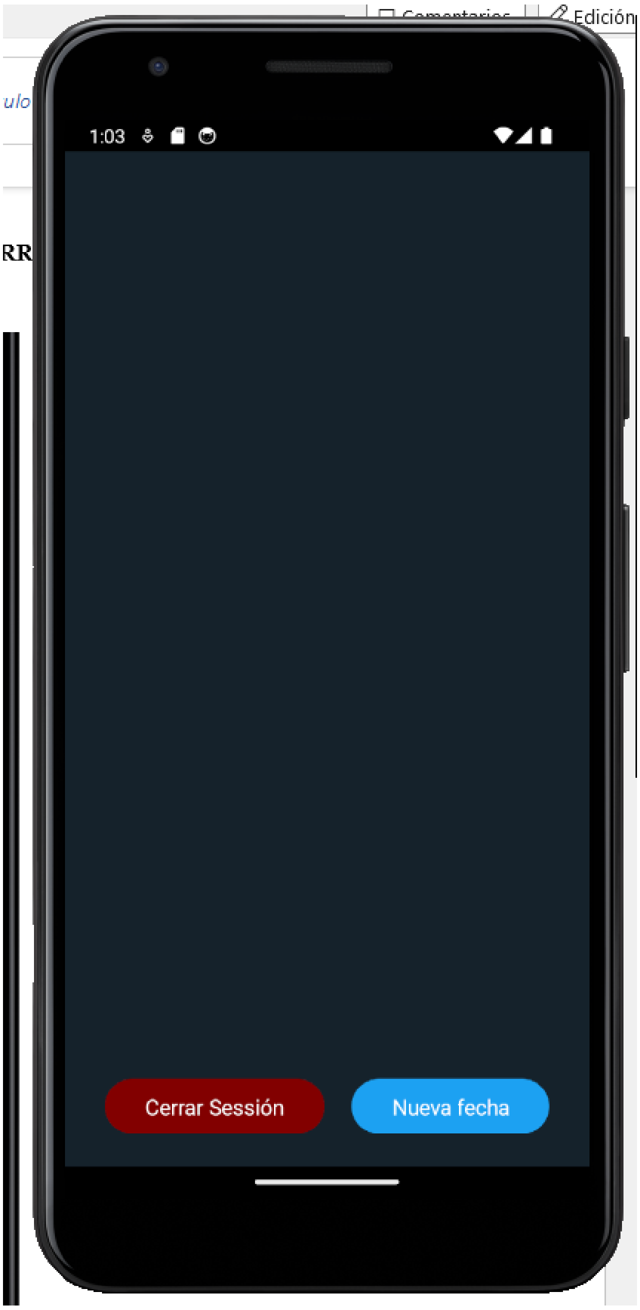
Alvarenga Romero, Josías Isaac #AR200778,

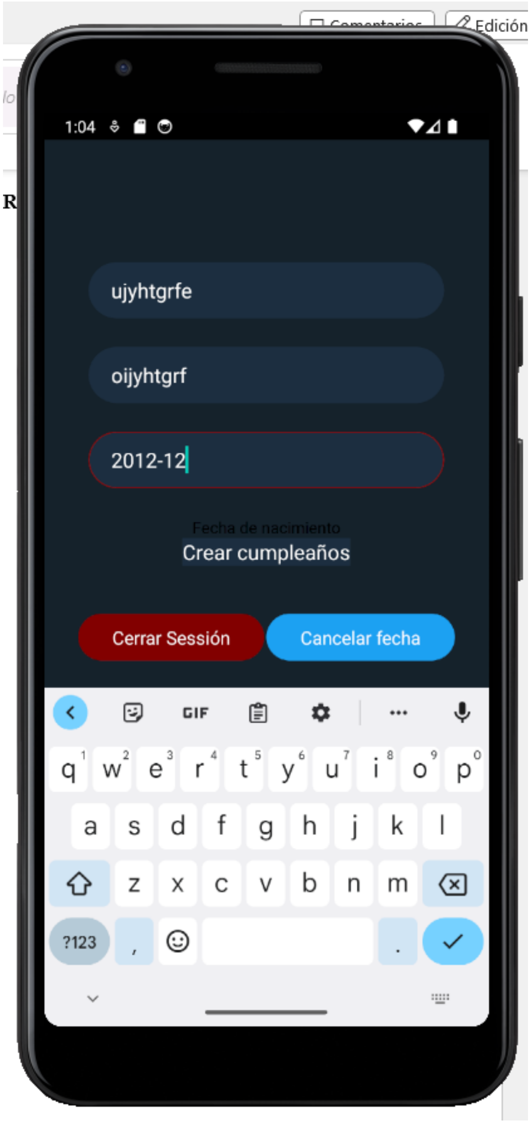
Antiguo Cuscatlán, 8 de noviembre de 2023

https://github.com/isaac-rom/birthday-guia11

DESARROLLO


Correo electronico
Contraseña
Iniciar sesión
Registrate


Comentarios
Edición
1:03
Cerrar Sessión
Nueva fecha



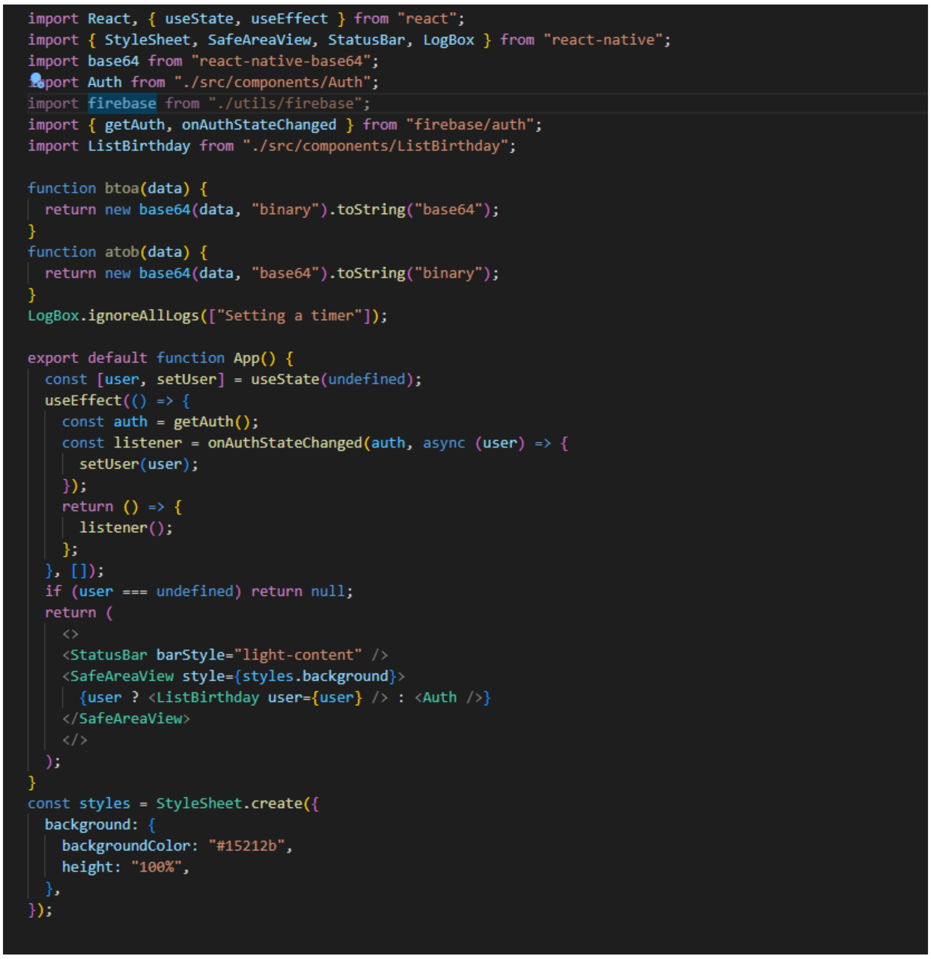
ujyhtgrfe

oijyhtgrf

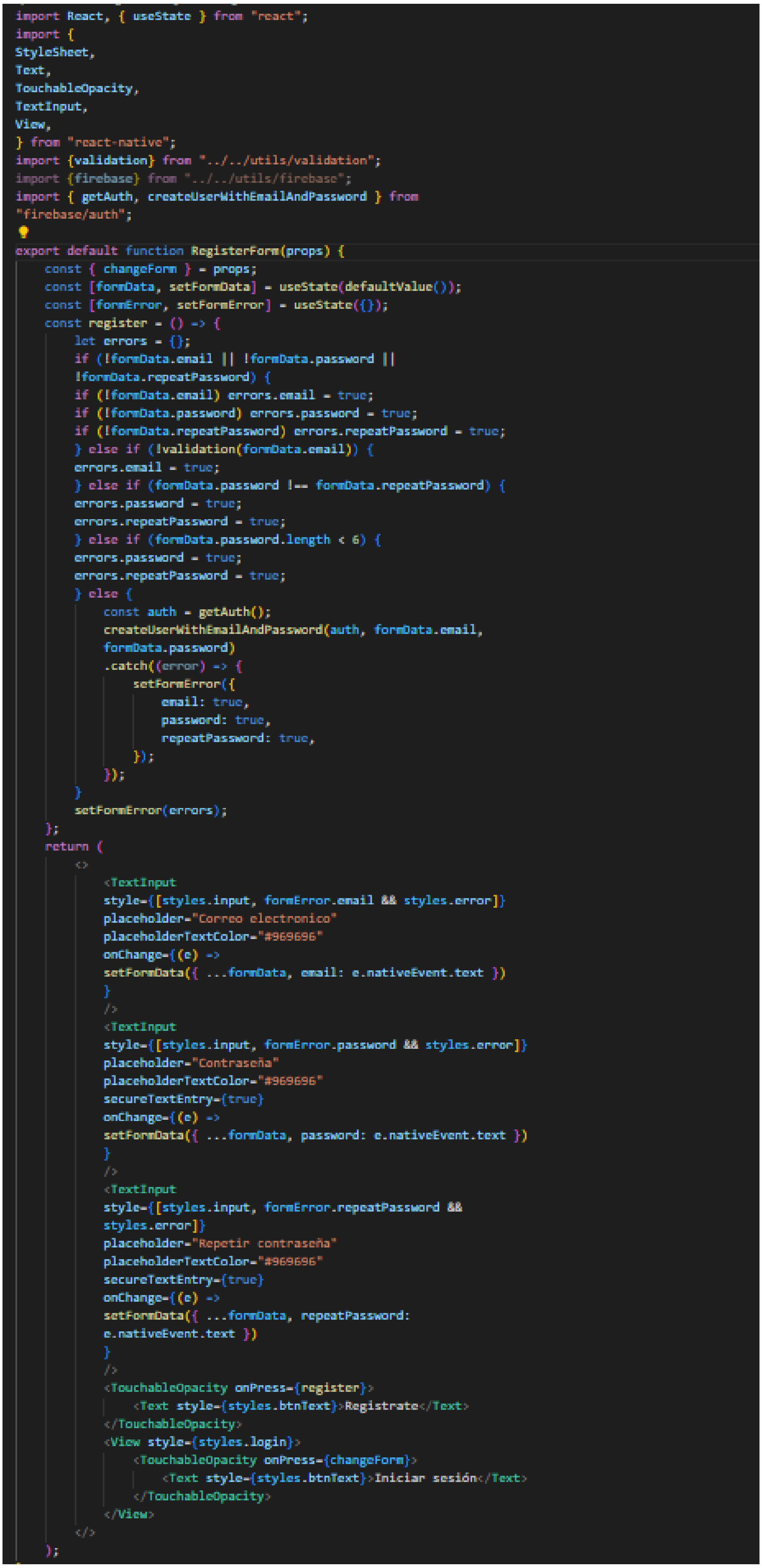
2012-12

Fecha de nacimiento

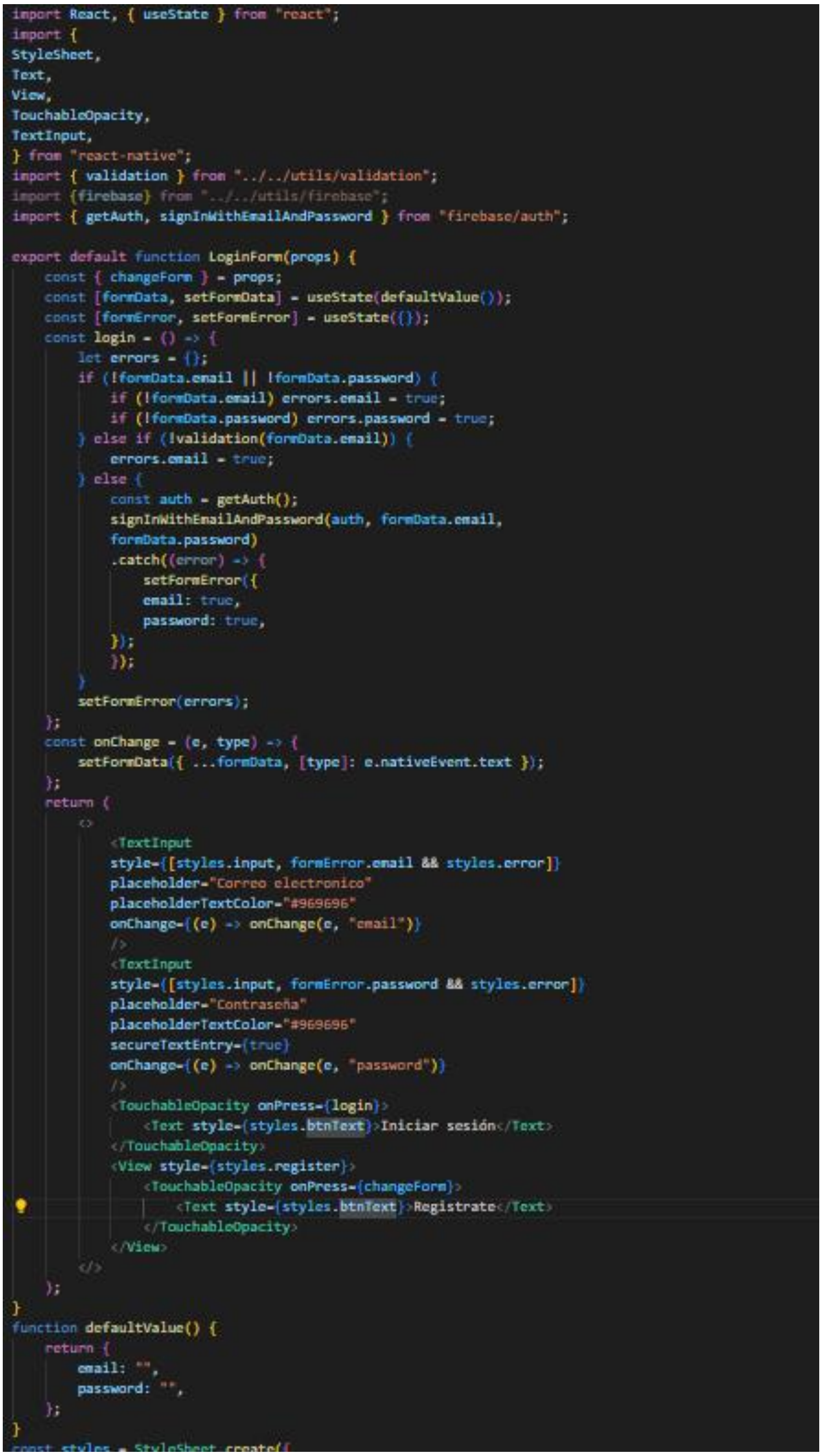
Crear cumpleaños



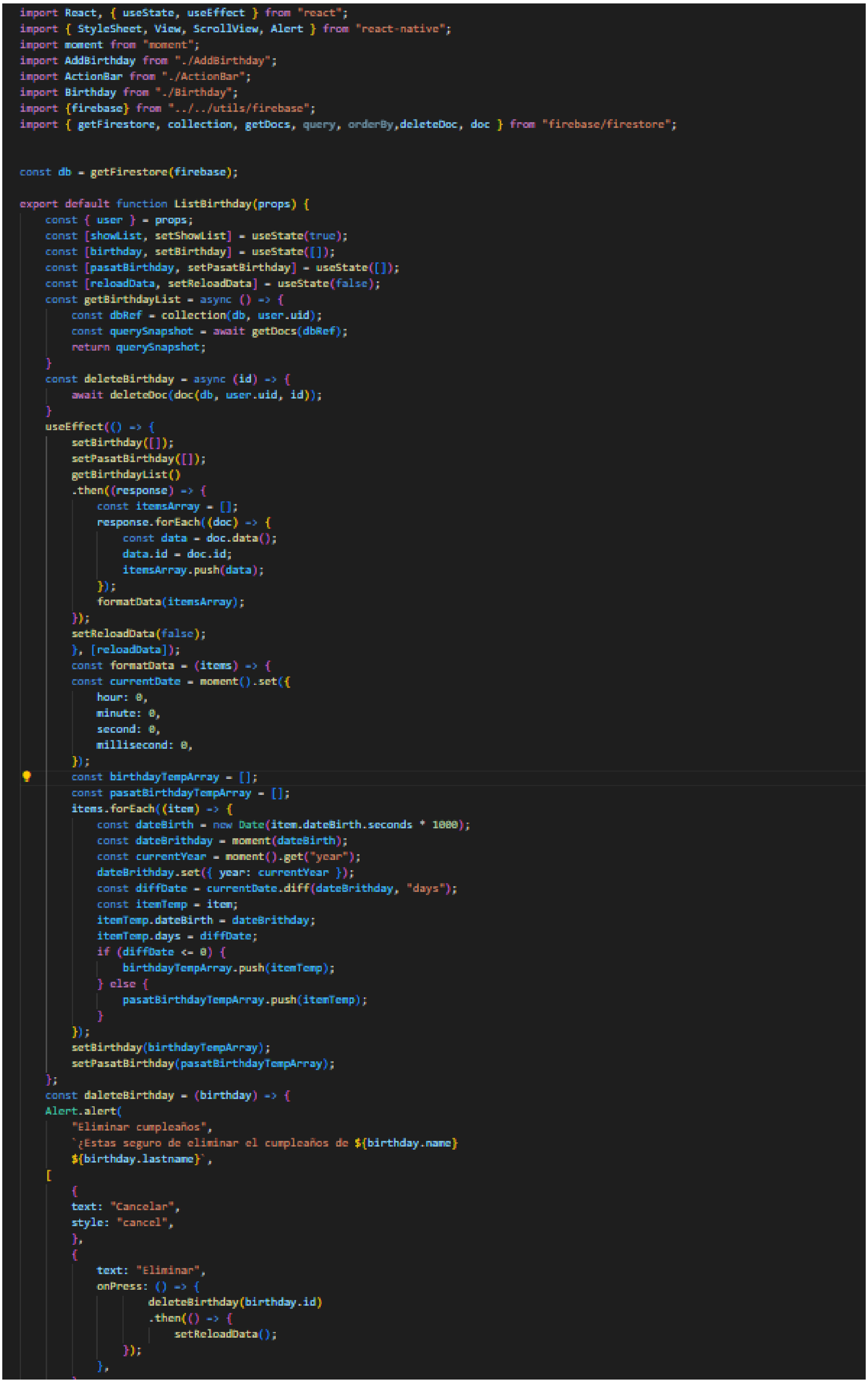
```
import React, { useState, useEffect } from "react";
import { StyleSheet, SafeAreaView, StatusBar, LogBox } from "react-native";
import base64 from "react-native-base64";
import Auth from "./src/components/Auth";
import firebase from "./utils/firebase";
import { getAuth, onAuthStateChanged } from "firebase/auth";
import ListBirthday from "./src/components/ListBirthday";

function btoa(data) {
  return new base64(data, "binary").toString("base64");
}
function atob(data) {
  return new base64(data, "base64").toString("binary");
}
LogBox.ignoreAllLogs(["Setting a timer"]);

export default function App() {
  const [user, setUser] = useState(undefined);
  useEffect(() => {
    const auth = getAuth();
    const listener = onAuthStateChanged(auth, async (user) => {
      setUser(user);
    });
    return () => {
      listener();
    };
  }, []);
  if (user === undefined) return null;
  return (
    <>
    <StatusBar barStyle="light-content" />
    <SafeAreaView style={styles.background}>
      {user ? <ListBirthday user={user} /> : <Auth />}
    </SafeAreaView>
    </>
  );
}
const styles = StyleSheet.create({
  background: {
    backgroundColor: "#15212b",
    height: "100%",
  },
});
```



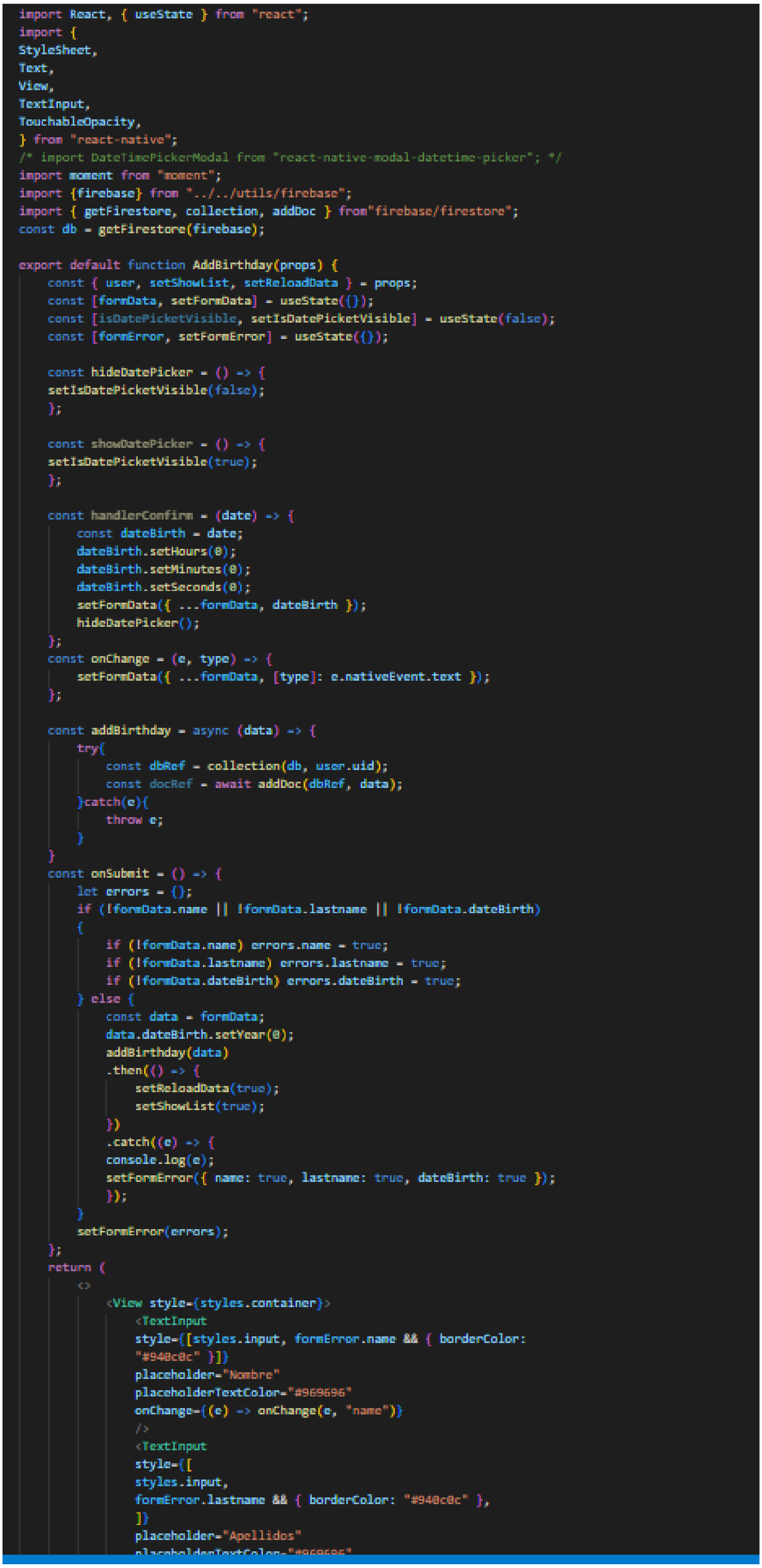
```
import React, { useState } from "react";
import {
StyleSheet,
Text,
TouchableOpacity,
TextInput,
View,
} from "react-native";
import {validation} from "../../utils/validation";
import {firebase} from "../../utils/firebase";
import { getAuth, createUserWithEmailAndPassword } from
"firebase/auth";

export default function RegisterForm(props) {
    const { changeForm } = props;
    const [formData, setFormData] = useState(defaultValue());
    const [formError, setFormError] = useState({});
    const register = () => {
        let errors = {};
        if (!formData.email || !formData.password ||
        !formData.repeatPassword) {
        if (!formData.email) errors.email = true;
        if (!formData.password) errors.password = true;
        if (!formData.repeatPassword) errors.repeatPassword = true;
        } else if (!validation(formData.email)) {
        errors.email = true;
        } else if (formData.password !== formData.repeatPassword) {
        errors.password = true;
        errors.repeatPassword = true;
        } else if (formData.password.length < 6) {
        errors.password = true;
        errors.repeatPassword = true;
        } else {
            const auth = getAuth();
            createUserWithEmailAndPassword(auth, formData.email,
            formData.password)
            .catch((error) => {
                setFormError({
                    email: true,
                    password: true,
                    repeatPassword: true,
                });
            });
        }
        setFormError(errors);
    };
    return (
        <>
            <TextInput
            style={[styles.input, formError.email && styles.error]}
            placeholder="Correo electronico"
            placeholderTextColor="#969696"
            onChange={(e) =>
            setFormData({ ...formData, email: e.nativeEvent.text })
            }
            />
            <TextInput
            style={[styles.input, formError.password && styles.error]}
            placeholder="Contraseña"
            placeholderTextColor="#969696"
            secureTextEntry={true}
            onChange={(e) =>
            setFormData({ ...formData, password: e.nativeEvent.text })
            }
            />
            <TextInput
            style={[styles.input, formError.repeatPassword &&
            styles.error]}
            placeholder="Repetir contraseña"
            placeholderTextColor="#969696"
            secureTextEntry={true}
            onChange={(e) =>
            setFormData({ ...formData, repeatPassword:
            e.nativeEvent.text })
            }
            />
            <TouchableOpacity onPress={register}>
                <Text style={styles.btnText}>Registrate</Text>
            </TouchableOpacity>
            <View style={styles.login}>
                <TouchableOpacity onPress={changeForm}>
                    <Text style={styles.btnText}>Iniciar sesión</Text>
                </TouchableOpacity>
            </View>
        </>
    );
```



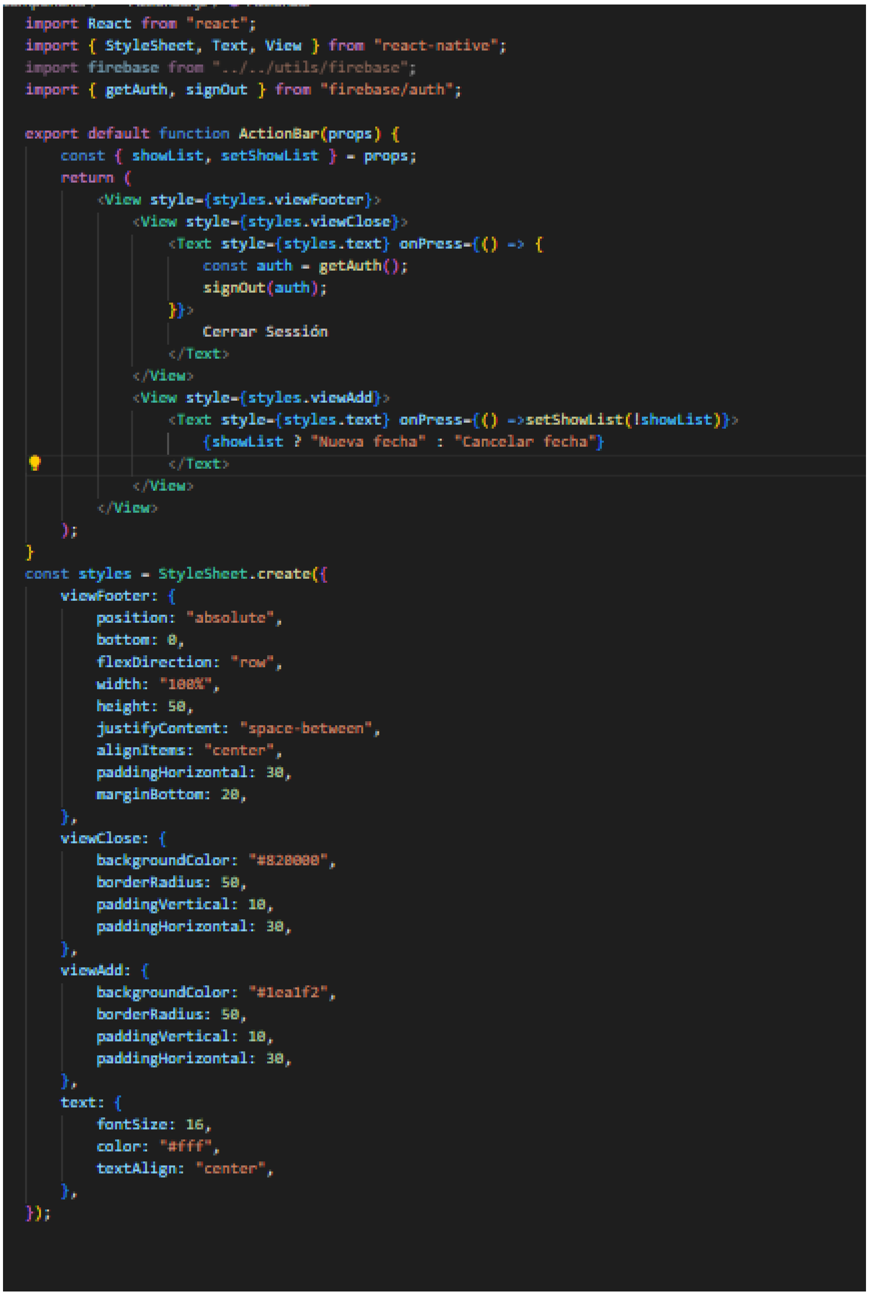
```
import React, { useState } from "react";
import {
StyleSheet,
Text,
View,
TouchableOpacity,
TextInput,
} from "react-native";
import { validation } from "../../utils/validation";
import {firebase} from "../../utils/firebase";
import { getAuth, signInWithEmailAndPassword } from "firebase/auth";

export default function LoginForm(props) {
    const { changeForm } = props;
    const [formData, setFormData] = useState(defaultValue());
    const [formError, setFormError] = useState({});
    const login = () => {
        let errors = {};
        if (!formData.email || !formData.password) {
            if (!formData.email) errors.email = true;
            if (!formData.password) errors.password = true;
        } else if (!validation(formData.email)) {
            errors.email = true;
        } else {
            const auth = getAuth();
            signInWithEmailAndPassword(auth, formData.email,
            formData.password)
            .catch((error) => {
                setFormError({
                email: true,
                password: true,
            });
            });
        }
        setFormError(errors);
    };
    const onChange = (e, type) => {
        setFormData({ ...formData, [type]: e.nativeEvent.text });
    };
    return (
        <>
            <TextInput
            style={[styles.input, formError.email && styles.error]}
            placeholder="Correo electronico"
            placeholderTextColor="#969696"
            onChange={(e) => onChange(e, "email")}
            />
            <TextInput
            style={[styles.input, formError.password && styles.error]}
            placeholder="Contraseña"
            placeholderTextColor="#969696"
            secureTextEntry={true}
            onChange={(e) => onChange(e, "password")}
            />
            <TouchableOpacity onPress={login}>
                <Text style={styles.btnText}>Iniciar sesión</Text>
            </TouchableOpacity>
            <View style={styles.register}>
                <TouchableOpacity onPress={changeForm}>
                    <Text style={styles.btnText}>Registrate</Text>
                </TouchableOpacity>
            </View>
        </>
    );
}
function defaultValue() {
    return {
        email: "",
        password: "",
    };
}
const styles = StyleSheet.create({
```



```
import React, { useState, useEffect } from "react";
import { StyleSheet, View, ScrollView, Alert } from "react-native";
import moment from "moment";
import AddBirthday from "./AddBirthday";
import ActionBar from "./ActionBar";
import Birthday from "./Birthday";
import {firebase} from "../../utils/firebase";
import { getFirestore, collection, getDocs, query, orderBy,deleteDoc, doc } from "firebase/firestore";


const db = getFirestore(firebase);

export default function ListBirthday(props) {
    const { user } = props;
    const [showList, setShowList] = useState(true);
    const [birthday, setBirthday] = useState([]);
    const [pasatBirthday, setPasatBirthday] = useState([]);
    const [reloadData, setReloadData] = useState(false);
    const getBirthdayList = async () => {
        const dbRef = collection(db, user.uid);
        const querySnapshot = await getDocs(dbRef);
        return querySnapshot;
    }
    const deleteBirthday = async (id) => {
        await deleteDoc(doc(db, user.uid, id));
    }
    useEffect(() => {
        setBirthday([]);
        setPasatBirthday([]);
        getBirthdayList()
        .then((response) => {
            const itemsArray = [];
            response.forEach((doc) => {
                const data = doc.data();
                data.id = doc.id;
                itemsArray.push(data);
            });
            formatData(itemsArray);
        });
        setReloadData(false);
        }, [reloadData]);
        const formatData = (items) => {
        const currentDate = moment().set({
            hour: 0,
            minute: 0,
            second: 0,
            millisecond: 0,
        });
        const birthdayTempArray = [];
        const pasatBirthdayTempArray = [];
        items.forEach((item) => {
            const dateBirth = new Date(item.dateBirth.seconds * 1000);
            const dateBrithday = moment(dateBirth);
            const currentYear = moment().get("year");
            dateBrithday.set({ year: currentYear });
            const diffDate = currentDate.diff(dateBrithday, "days");
            const itemTemp = item;
            itemTemp.dateBirth = dateBrithday;
            itemTemp.days = diffDate;
            if (diffDate <= 0) {
                birthdayTempArray.push(itemTemp);
            } else {
                pasatBirthdayTempArray.push(itemTemp);
            }
        });
        setBirthday(birthdayTempArray);
        setPasatBirthday(pasatBirthdayTempArray);
    };
    const daleteBirthday = (birthday) => {
    Alert.alert(
        "Eliminar cumpleaños",
        `¿Estas seguro de eliminar el cumpleaños de ${birthday.name}
        ${birthday.lastname}`,
    [
        {
        text: "Cancelar",
        style: "cancel",
        },
        {
            text: "Eliminar",
            onPress: () => {
                    deleteBirthday(birthday.id)
                    .then(() => {
                        setReloadData();
                });
            },
```



```
import React, { useState } from "react";
import {
StyleSheet,
Text,
View,
TextInput,
TouchableOpacity,
} from "react-native";
/* import DateTimePickerModal from "react-native-modal-datetime-picker"; */
import moment from "moment";
import {firebase} from "../../utils/firebase";
import { getFirestore, collection, addDoc } from"firebase/firestore";
const db = getFirestore(firebase);

export default function AddBirthday(props) {
    const { user, setShowList, setReloadData } = props;
    const [formData, setFormData] = useState({});
    const [isDatePicketVisible, setIsDatePicketVisible] = useState(false);
    const [formError, setFormError] = useState({});

    const hideDatePicker = () => {
    setIsDatePicketVisible(false);
    };

    const showDatePicker = () => {
    setIsDatePicketVisible(true);
    };

    const handlerConfirm = (date) => {
        const dateBirth = date;
        dateBirth.setHours(0);
        dateBirth.setMinutes(0);
        dateBirth.setSeconds(0);
        setFormData({ ...formData, dateBirth });
        hideDatePicker();
    };
    const onChange = (e, type) => {
        setFormData({ ...formData, [type]: e.nativeEvent.text });
    };

    const addBirthday = async (data) => {
        try{
            const dbRef = collection(db, user.uid);
            const docRef = await addDoc(dbRef, data);
        }catch(e){
            throw e;
        }
    }
    const onSubmit = () => {
        let errors = {};
        if (!formData.name || !formData.lastname || !formData.dateBirth)
        {
            if (!formData.name) errors.name = true;
            if (!formData.lastname) errors.lastname = true;
            if (!formData.dateBirth) errors.dateBirth = true;
        } else {
            const data = formData;
            data.dateBirth.setYear(0);
            addBirthday(data)
            .then(() => {
                setReloadData(true);
                setShowList(true);
            })
            .catch((e) => {
            console.log(e);
            setFormError({ name: true, lastname: true, dateBirth: true });
            });
        }
        setFormError(errors);
    };
    return (
        <>
            <View style={styles.container}>
                <TextInput
                style={[styles.input, formError.name && { borderColor:
                "#940c0c" }]}
                placeholder="Nombre"
                placeholderTextColor="#969696"
                onChange={(e) => onChange(e, "name")}
                />
                <TextInput
                style={[
                styles.input,
                formError.lastname && { borderColor: "#940c0c" },
                ]}
                placeholder="Apellidos"
                placeholderTextColor="#969696"
```



```
import React from "react";
import { StyleSheet, Text, View } from "react-native";
import firebase from "../../utils/firebase";
import { getAuth, signOut } from "firebase/auth";

export default function ActionBar(props) {
    const { showList, setShowList } = props;
    return (
        <View style={styles.viewFooter}>
            <View style={styles.viewClose}>
                <Text style={styles.text} onPress={() => {
                    const auth = getAuth();
                    signOut(auth);
                }}>
                    Cerrar Sesión
                </Text>
            </View>
            <View style={styles.viewAdd}>
                <Text style={styles.text} onPress={() =>setShowList(!showList)}>
                    {showList ? "Nueva fecha" : "Cancelar fecha"}
                </Text>
            </View>
        </View>
    );
}
const styles = StyleSheet.create({
    viewFooter: {
        position: "absolute",
        bottom: 0,
        flexDirection: "row",
        width: "100%",
        height: 50,
        justifyContent: "space-between",
        alignItems: "center",
        paddingHorizontal: 30,
        marginBottom: 20,
    },
    viewClose: {
        backgroundColor: "#820000",
        borderRadius: 50,
        paddingVertical: 10,
        paddingHorizontal: 30,
    },
    viewAdd: {
        backgroundColor: "#1ea1f2",
        borderRadius: 50,
        paddingVertical: 10,
        paddingHorizontal: 30,
    },
    text: {
        fontSize: 16,
        color: "#fff",
        textAlign: "center",
    },
});
```