	UNIVERSIDAD DON BOSCO **FACULTAD DE INGENIERÍA** **ESCUELA DE COMPUTACIÓN**	
CICLO: 02	**GUIA DE LABORATORIO #11**	
	Nombre de la Práctica:	**Firebase**
	MATERIA:	**Diseño y Programación de Software Multiplataforma**

I. Resultados de aprendizaje

Que el estudiante utilice firebase real time database como base de datos relacional

- Que conozca los métodos de acceso a nodos de firebase
- Que utilice librerías adicionales

II. Introducción

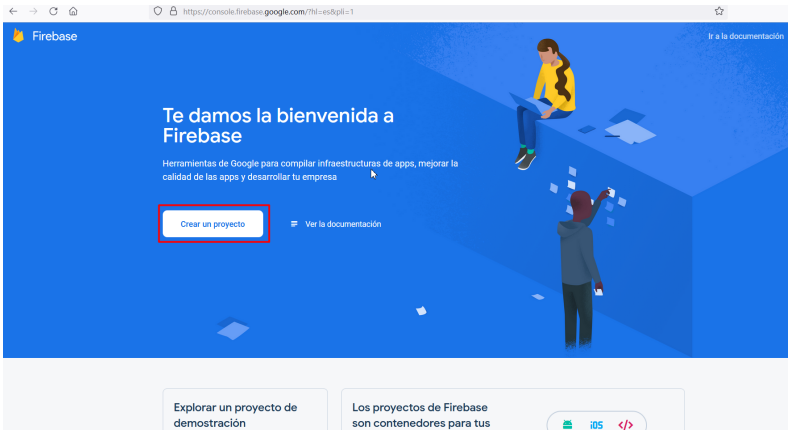
Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización.

Firebase proporciona una base de datos en tiempo real, back-end y organizada en forma de árbol JSON. El servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase.

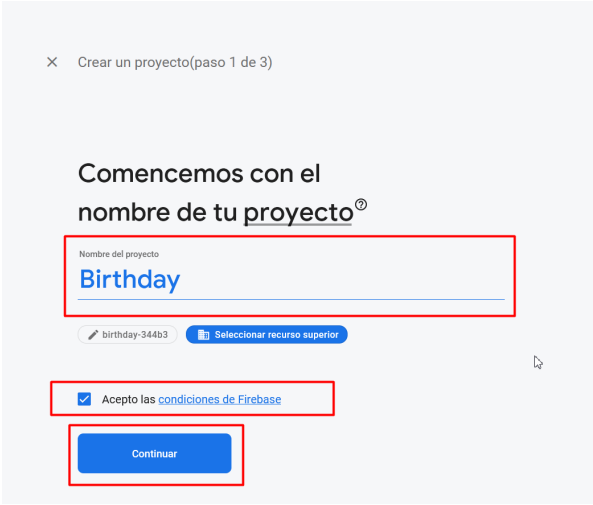
III. DESARROLLO

Parte I: Firebase

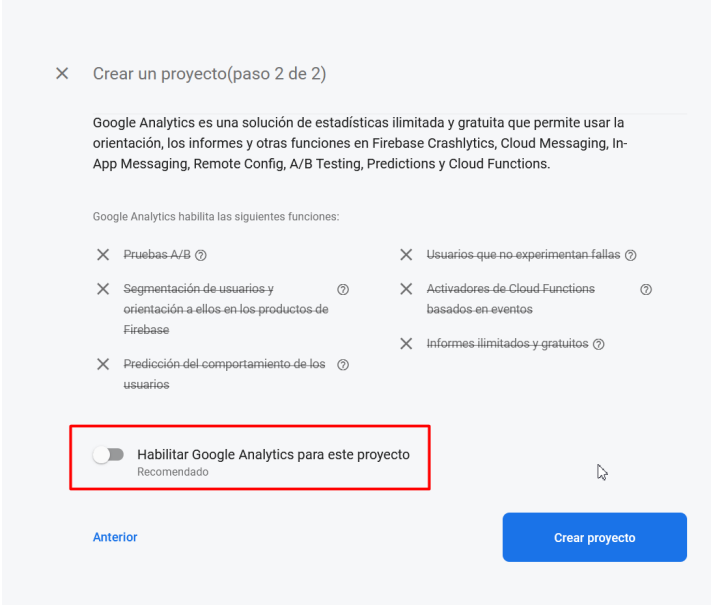
1. Ir a la página de Firebase: https://firebase.google.com/?hl=es
2. Iniciar sesión con nuestra cuenta de Gmail.
3. Ir a la consola, y dar clic en añadir proyecto



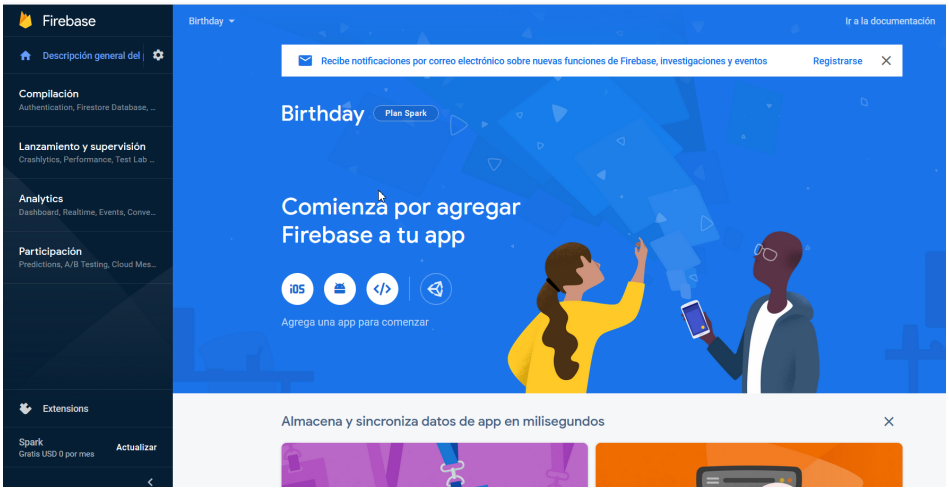
4. Poner el nombre de proyecto: Birthday y clic en continuar



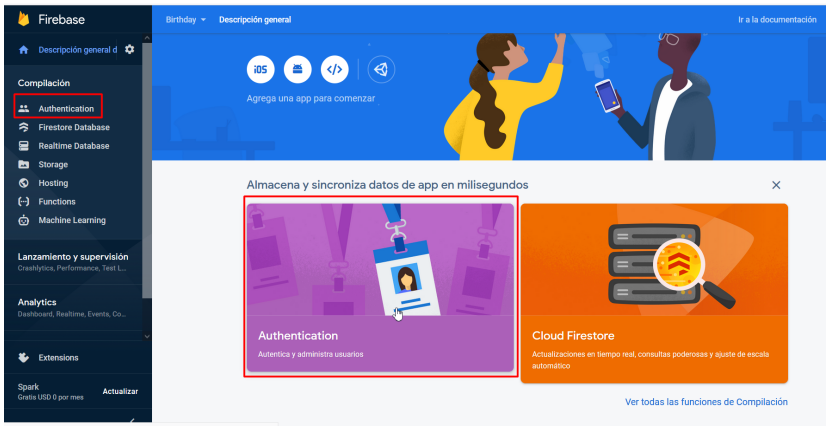
5. No habilitar analytics porque no se utilizará en este proyecto, clic en crear proyecto



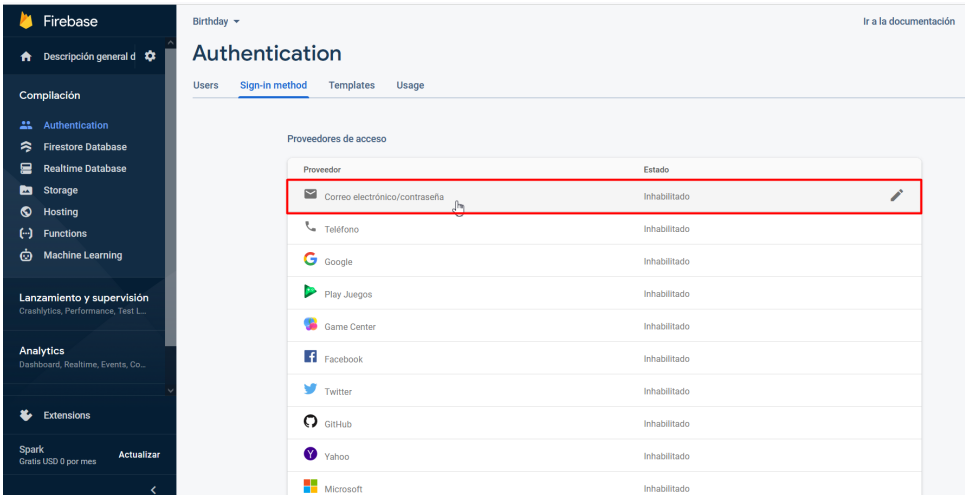
6. Clic en continuar y deberás ver una pantalla similar a la siguiente, que es ya la página de nuestro proyecto:



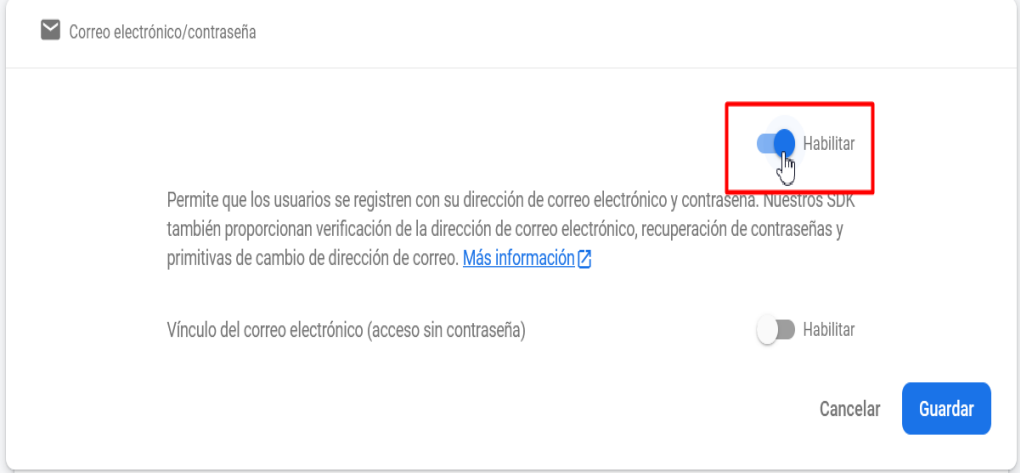
7. Clic en autenticación para configurar el inicio de sesión



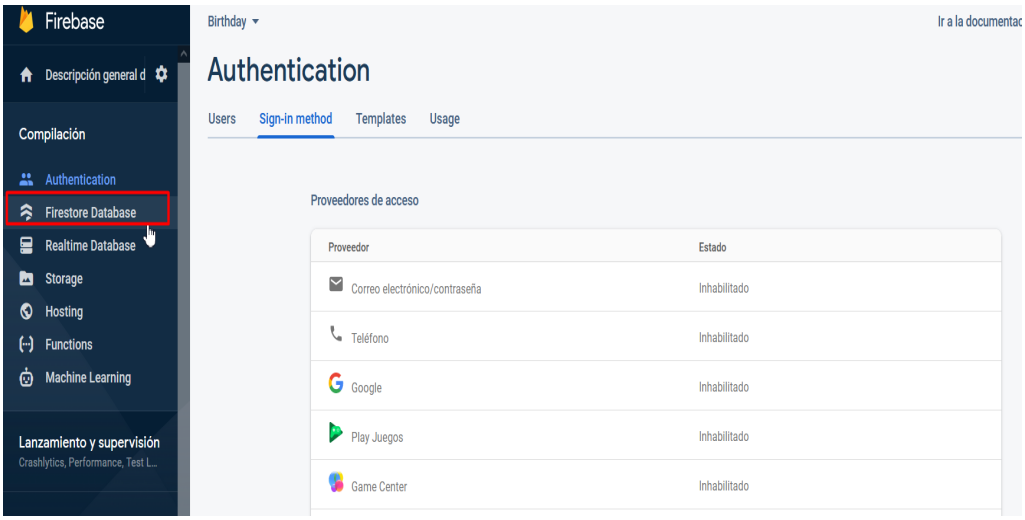
8. Clic en **comenzar** y deberá ver la siguiente página, en donde daremos clic la opción **correo electrónico/contraseña**



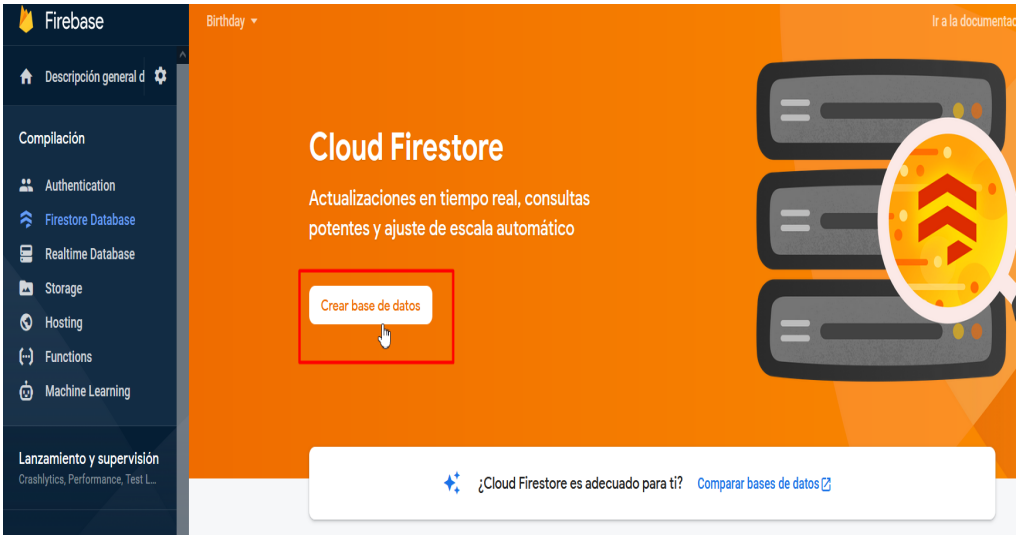
9. Para habilitar un sistema de login por correo electrónico debes habilitar la siguiente opción



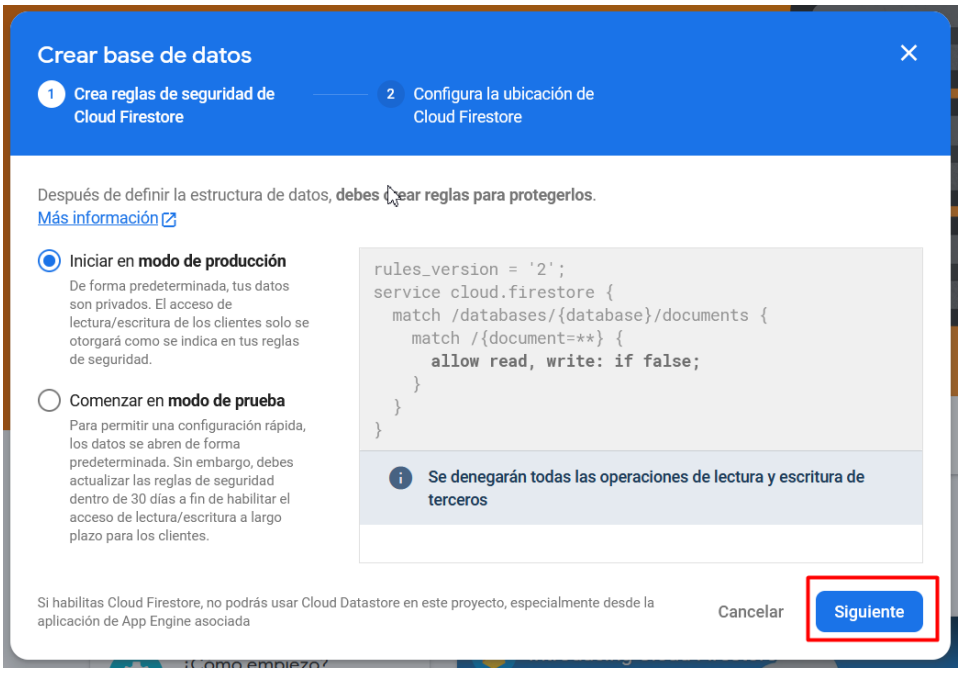
10. Ahora crearemos nuestra base de datos para la aplicación, dar clic en firestore Database



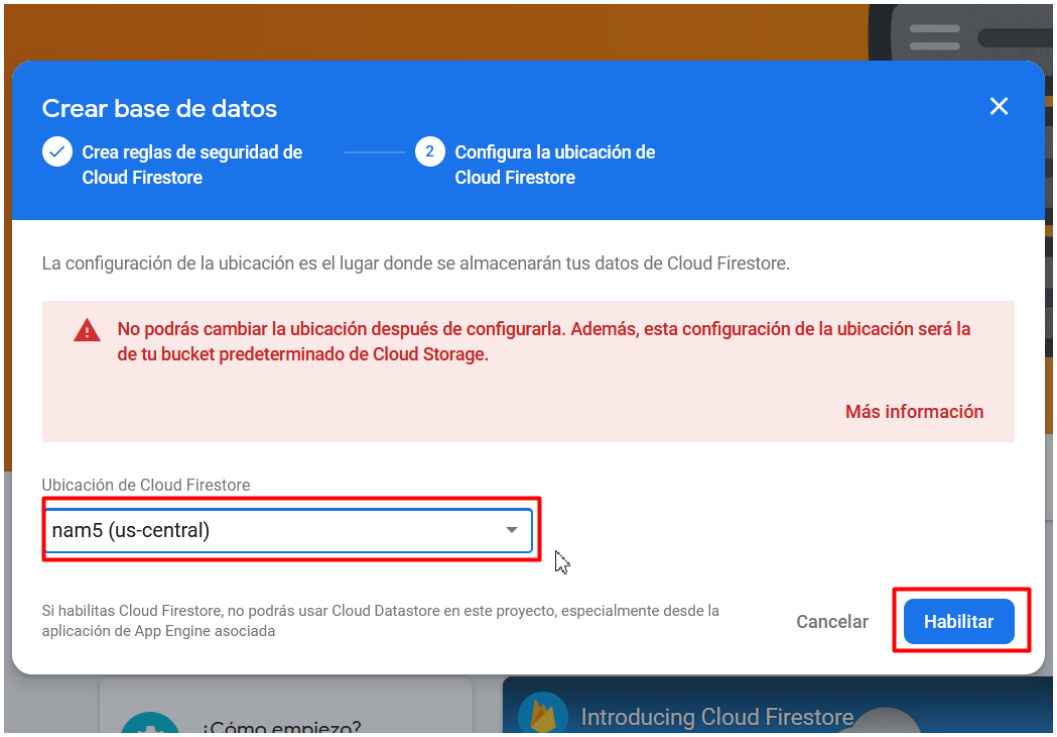
11. Clic en crear base de datos



12. Nos mostrara la siguiente pantalla con reglas, en donde le daremos clic en siguiente

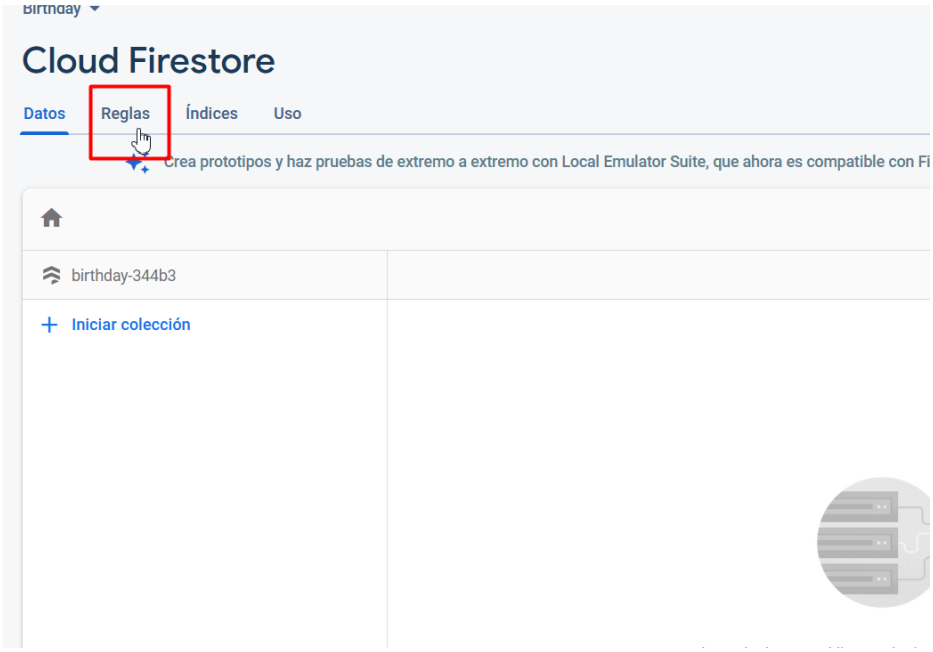


13. Seleccionar una ubicación multiregional, dar clic en habilitar.



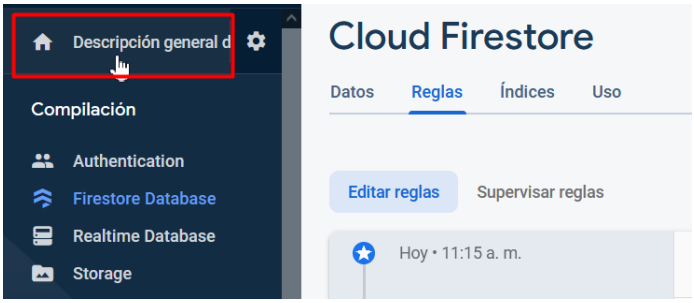
Nota: Para maximizar la disponibilidad y la durabilidad de la base de datos, debes seleccionar una ubicación multirregional. Las ubicaciones multirregionales pueden soportar la pérdida de regiones completas y mantener la disponibilidad sin perder datos

14. Con la base de datos creada ahora debemos de configurar las reglas de nuestra base de datos:



15. Deberemos de modificar las reglas para que solo usuarios autenticados puedan leer y escribir en nuestra base de datos, si no esta autenticado entonces el usuario solo podrá registrarse y loguearse.

16. Ya con la configuración de seguridad para nuestra base de datos, procedemos a configurar nuestra base para poder conectarla a nuestra aplicación react. Damos clic en descripción general.



17. Nuestra aplicación no es Android, ni Ios si no que hibrida creado en un entorno web así que la configuración deberemos hacerlo para web

NOTA: Nuestra aplicación no es Android, ni iOS, es una aplicación creada en REACT y por lo tanto es hibrida, y por eso debemos configurar como si fuera un entorno web

18. Le ponemos nombre a nuestra aplicación, y clic en Registrar app

19. Copie los datos de su código de configuración que es lo que utilizaremos en nuestra aplicación react.

PARTE II: Aplicación react

1. Crear la aplicación con nombre birthday

2. Crear una carpeta src y dentro de ella crear otra con nombre utils y dentro de ella crear un fichero con nombre firebase.js. En este archivo copiamos el código que nos genera firebase que obtuvimos en el paso 19 de la Parte I.

```
// Import the functions you need from the SDKs you need
import { firebase } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
// Your web app's Firebase configuration
// Initialize Firebase
const firebaseConfig = {
  apiKey: 'api-key',
  authDomain: 'project-id.firebaseapp.com',
  databaseURL: 'https://project-id.firebaseio.com',
  projectId: 'project-id',
  storageBucket: 'project-id.appspot.com',
  messagingSenderId: 'sender-id',
  appId: 'app-id',
  measurementId: 'G-measurement-id',
};

// Initialize Firebase
export default firebase.initializeApp(firebaseConfig);
```

sus credenciales del proyecto firebase

3. Procedemos a instalar los modulo que utilizaremos en la aplicación:
   npm install --save firebase ó
   npm install --save moment
   npm install --save react-native-base64
   npm i react-native-modal-datetime-picker @react-native-community/datetimepicker

4. Dentro de la carpeta utils agregamos el fichero validations.js

```
export function validateEmail(email) {
    const re = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".
+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9
]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(email);
  }
```

5. Crear una carpeta de nombre components, adentro de la carpeta src ya creada.
6. Dentro de la carpeta components agregar el fichero ActionBar.js

```
import React from 'react';
import {StyleSheet, Text, View} from 'react-native';
import firebase from '../utils/firebase';

export default function ActionBar(props) {
  const {showList, setShowList} = props;

  return (
```

```
    <View style={styles.viewFooter}>
      <View style={styles.viewClose}>
        <Text style={styles.text} onPress={() => firebase.auth().signOut(
)}>
          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',
  },
});
```

7. Dentro de la carpeta components agregar el fichero AddBirthday.js

```
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 'firebase/firestore';

firebase.firestore().settings({experimentalForceLongPolling: true});
const db = firebase.firestore(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 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);
      db.collection(user.uid)
        .add(data)
        .then(() => {
          setReloadData(true);
          setShowList(true);
        })
        .catch(() => {
          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: '#940
c0c'}]}
          placeholder="Apellidos"
          placeholderTextColor="#969696"
          onChange={(e) => onChange(e, 'lastname')}
        />
        <View
          style={[
            styles.input,
            styles.datepicker,
            formError.dateBirth && {borderColor: '#940c0c'},
          ]}>
          <Text
            style={{
              color: formData.dateBirth ? '#fff' : '#969696',
              fontSize: 18,
            }}
            onPress={showDatePicker}>
```

```
            {formData.dateBirth
              ? moment(formData.dateBirth).format('LL')
              : 'Fecha de nacimiento'}
          </Text>
        </View>
        <TouchableOpacity onPress={onSubmit}>
          <Text style={styles.addButton}>Crear cumpleaños</Text>
        </TouchableOpacity>
      </View>

      <DateTimePickerModal
        isVisible={isDatePicketVisible}
        mode="date"
        onConfirm={handlerConfirm}
        onCancel={hideDatePicker}
      />
    </>
  );
}

const styles = StyleSheet.create({
  container: {
    height: '100%',
    width: '100%',
    justifyContent: 'center',
    alignItems: 'center',
  },
  input: {
    height: 50,
    color: '#fff',
    width: '80%',
    marginBottom: 25,
    backgroundColor: '#1e3040',
    paddingHorizontal: 20,
    borderRadius: 50,
    fontSize: 18,
    borderWidth: 1,
    borderColor: '#1e3040',
  },
  datepicker: {
    justifyContent: 'center',
  },
  addButton: {
    fontSize: 18,
    color: '#fff',
    backgroundColor:'#1e3040',
  },
});
```

8. Dentro de la carpeta components agregar el fichero Auth.js

```
import React, {useState} from 'react';
import {StyleSheet, View, Text, Image} from 'react-native';
import LoginForm from './LoginForm';
import RegisterForm from './RegisterForm';

export default function Auth() {
  const [isLogin, setIsLogin] = useState(true);

  const changeForm = () => {
    setIsLogin(!isLogin);
  };

  return (
    <View style={styles.view}>
      <Image style={styles.logo} source={require('../assets/logo.png')} /
>
      {isLogin ? (
        <LoginForm changeForm={changeForm} />
      ) : (
        <RegisterForm changeForm={changeForm} />
      )}
    </View>
  );
}

const styles = StyleSheet.create({
  view: {
    flex: 1,
    alignItems: 'center',
  },
  logo: {
    width: '80%',
    height: 240,
    marginTop: 50,
    marginBottom: 50,
  },
});
```

9. Dentro de la carpeta components agregar el fichero Birthday.js

```
import React from 'react';
import {StyleSheet, Text, View, TouchableOpacity} from 'react-native';

export default function Birthday(props) {
```

```
  const {birthday, daleteBirthday} = props;
  const pasat = birthday.days > 0 ? true : false;

  const infoDay = () => {
    if (birthday.days === 0) {
      return <Text style={{color: '#fff'}}>Es suy cumpleaños</Text>;
    } else {
      const days = -birthday.days;
      return (
        <View style={styles.textCurrent}>
          <Text>{days}</Text>
          <Text>{days === 1 ? 'Dia' : 'Dias'}</Text>
        </View>
      );
    }
  };

  return (
    <TouchableOpacity
      style={[
        styles.card,
        pasat
          ? styles.pasat
          : birthday.days === 0
          ? styles.actual
          : styles.current,
      ]}
      onPress={() => daleteBirthday(birthday)}>
      <Text style={styles.userName}>
        {birthday.name} {birthday.lastname}
      </Text>
      {pasat ? <Text style={{color: '#fff'}}>Pasado</Text> : infoDay()}
    </TouchableOpacity>
  );
}

const styles = StyleSheet.create({
  card: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    height: 60,
    alignItems: 'center',
    paddingHorizontal: 10,
    margin: 10,
    borderRadius: 15,
  },
  actual: {
    backgroundColor: '#559204',
```

```
  },
  current: {
    backgroundColor: '#1ae1f2',
  },
  pasat: {
    backgroundColor: '#820000',
  },
  userName: {
    color: '#fff',
    fontSize: 16,
  },
  textCurrent: {
    backgroundColor: '#fff',
    borderRadius: 20,
    width: 50,
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

10. Dentro de la carpeta components agregar el fichero ListBirthday.js

```
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 'firebase/firestore';

firebase.firestore().settings({experimentalForceLongPolling: true});
const db = firebase.firestore(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);

  useEffect(() => {
    setBirthday([]);
    setPasatBirthday([]);
    db.collection(user.uid)
      .orderBy('dateBirth', 'asc')
      .get()
```

```
    .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} ${birt
hday.lastname}`,
      [
        {
          text: 'Cancelar',
          style: 'cancel',
        },
        {
          text: 'Eliminar',
          onPress: () => {
            db.collection(user.uid)
              .doc(birthday.id)
              .delete()
              .then(() => {
                setReloadData();
              });
          },
        },
      ],
      {cancelable: false},
    );
  };

  return (
    <View style={styles.container}>
      {showList ? (
        <ScrollView style={styles.scrollView}>
          {birthday.map((item, index) => (
            <Birthday
              key={index}
              birthday={item}
              daleteBirthday={daleteBirthday}
            />
          ))}
          {pasatBirthday.map((item, index) => (
            <Birthday
              key={index}
              birthday={item}
              daleteBirthday={daleteBirthday}
            />
          ))}
        </ScrollView>
      ) : (
        <AddBirthday
          user={user}
          setShowList={setShowList}
          setReloadData={setReloadData}
        />
```

```
      )}
      <ActionBar showList={showList} setShowList={setShowList} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    alignItems: 'center',
    height: '100%',
  },
  scrollView: {
    marginBottom: 50,
    width: '100%',
  },
});
```

11. Dentro de la carpeta components agregar el fichero LoginForm.js

```
import React, {useState} from 'react';
import {
  StyleSheet,
  Text,
  View,
  TouchableOpacity,
  TextInput,
} from 'react-native';
import {validateEmail} from '../utils/validations';
import firebase from '../utils/firebase';

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 (!validateEmail(formData.email)) {
      errors.email = true;
    } else {
      firebase
        .auth()
        .signInWithEmailAndPassword(formData.email, formData.password)
        .catch(() => {
```

```
          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({
  input: {
    height: 50,
    color: '#fff',
    width: '80%',
    marginBottom: 25,
    backgroundColor: '#1e3040',
    paddingHorizontal: 20,
    borderRadius: 50,
    fontSize: 18,
    borderWidth: 1,
    borderColor: '#1e3040',
  },
  btnText: {
    color: '#fff',
    fontSize: 18,
  },
  register: {
    flex: 1,
    justifyContent: 'flex-end',
    marginBottom: 10,
  },
  error: {
    borderColor: '#940c0c',
  },
});
```

12. Dentro de la carpeta components agregar el fichero RegisterForm.js

```
import React, {useState} from 'react';
import {
  StyleSheet,
  Text,
  TouchableOpacity,
  TextInput,
  View,
} from 'react-native';
import {validateEmail} from '../utils/validations';
import firebase from '../utils/firebase';

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 (!validateEmail(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 {
      firebase
        .auth()
        .createUserWithEmailAndPassword(formData.email, formData.password
)
        .catch(() => {
          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.t
ext})}
      />
      <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>
    </>
  );
}

function defaultValue() {
  return {
    email: '',
    password: '',
    repeatPassword: '',
  };
}

const styles = StyleSheet.create({
  btnText: {
    color: '#fff',
    fontSize: 18,
  },
  input: {
    height: 50,
    color: '#fff',
    width: '80%',
    marginBottom: 25,
    backgroundColor: '#1e3040',
    paddingHorizontal: 20,
    borderRadius: 50,
    fontSize: 18,
    borderWidth: 1,
    borderColor: '#1e3040',
  },
  login: {
    flex: 1,
```

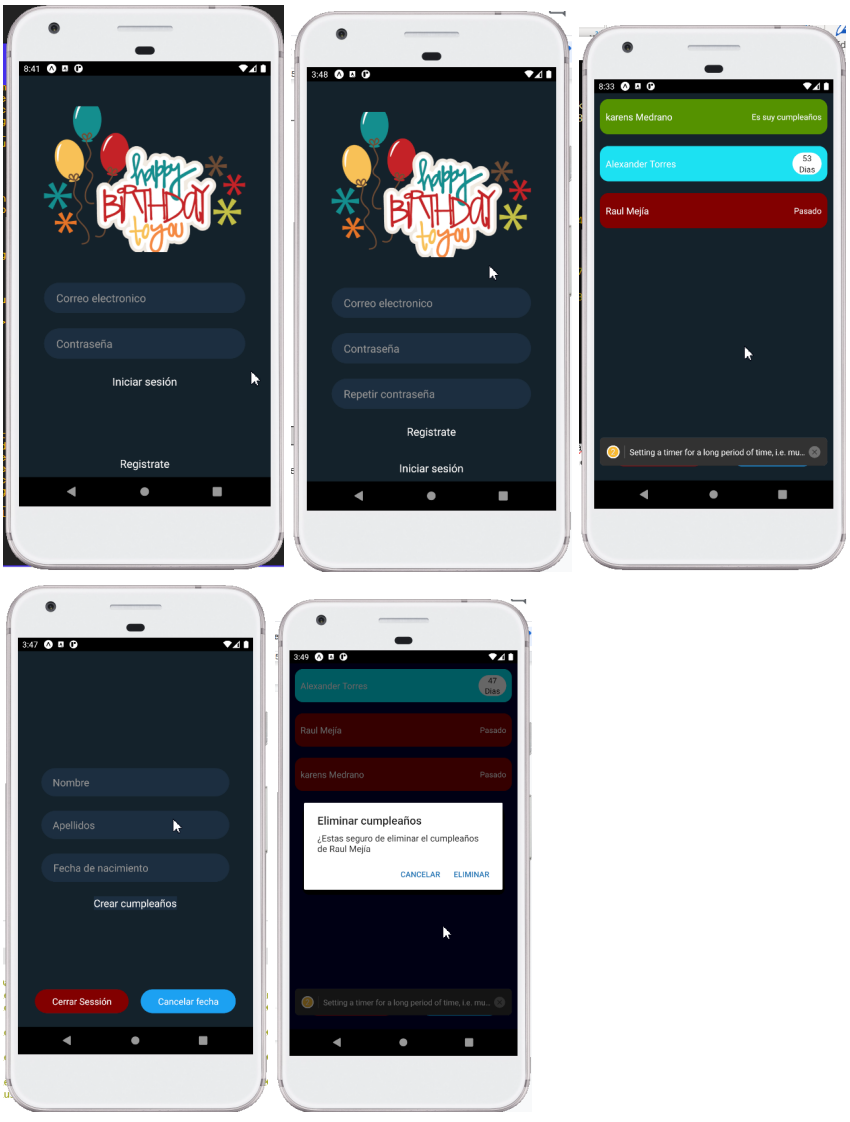
```
    justifyContent: 'flex-end',
    marginBottom: 10,
  },
  error: {
    borderColor: '#940c0c',
  },
});
```

13. Agregar la imagen “logo.png” proporcionada en los recursos en una nueva carpeta, dentro del src llamada assets.
14. Procedemos a realizar los cambios en App.js

```
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 './src/utils/firebase';
import '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(() => {
    firebase.auth().onAuthStateChanged((response) => {
      setUser(response);
    });
  }, []);

  if (user === undefined) return null;

  return (
    <>
      <StatusBar barStyle="light-content" />
      <SafeAreaView style={styles.background}>
        {user ? <ListBirthday user={user} /> : <Auth />}
      </SafeAreaView>
    </>
  );
}
```

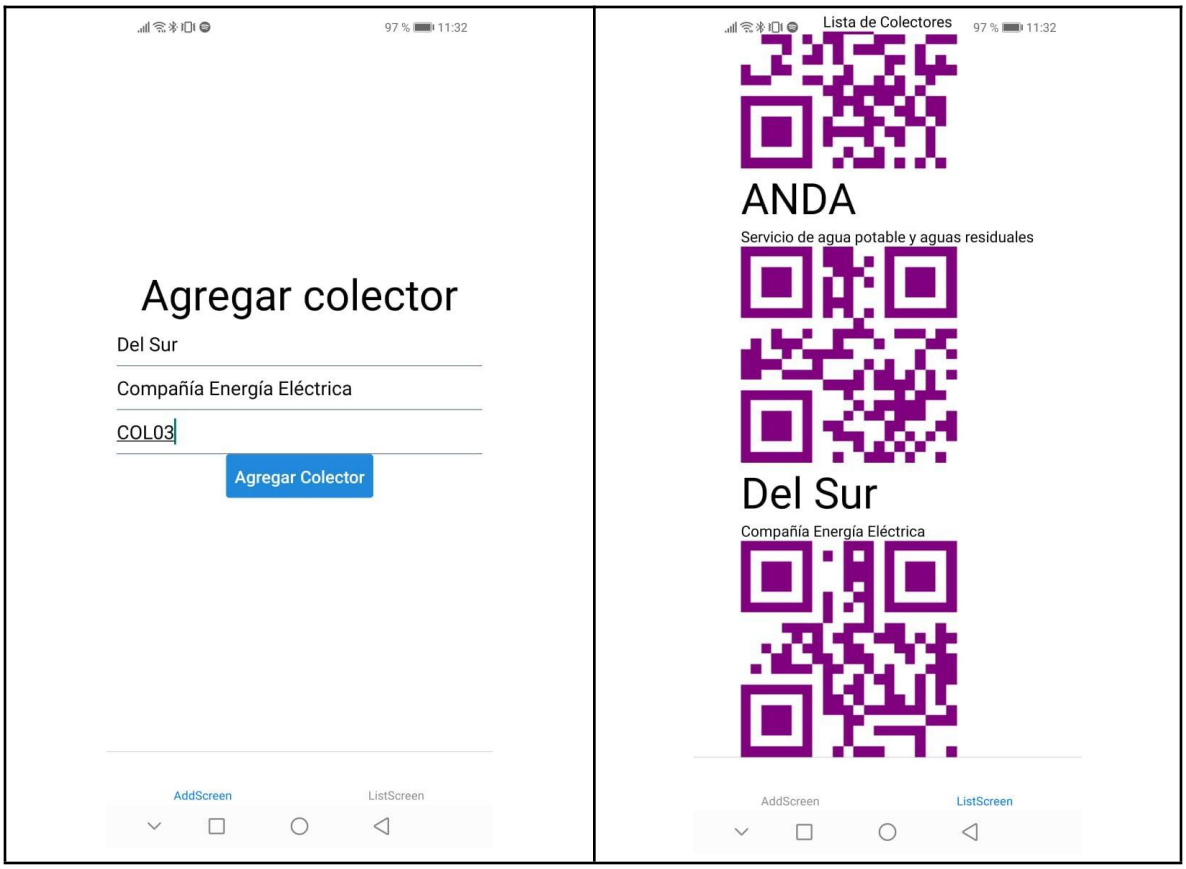
```
const styles = StyleSheet.create({
  background: {
    backgroundColor: '#15212b',
    height: '100%',
  },
});
```

15. Ejecuta la aplicación y deberás poder las siguientes pantallas



V. EJERCICIOS COMPLEMENTARIOS

- Realizar una aplicación para de colectores, usando react-native y firabase, la cual puede tener la siguiente vista. Realizar los respectivos mantenimientos para cada colector.


97 % 11:32
Agregar colector
Del Sur
Compañía Energía Eléctrica
COL03
Agregar Colector
AddScreen
ListScreen
Lista de Colectores
97 % 11:32
ANDA
Servicio de agua potable y aguas residuales
Del Sur
Compañía Energía Eléctrica
AddScreen
ListScreen