

# Procesual Hito III

Docente: Ing William Barra

Estudiante: Jose Luis Rodriguez Peredo

Materia: Programación de dispositivos  
móviles

Gestión: 2020.

# Parte Teórica

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

# 1.-Defina que es un componente en React Native y muestre un ejemplo.

R: Un componente en react es un elemento reutilizable e implementable tanto en cosas pequeñas como en grandes secciones de código implementado. Los componentes permiten separar la interfaz de usuario en piezas independientes, reutilizables y pensar en cada pieza de forma aislada.

arbitrarias (llamadas "props") y devuelven a React elementos que describen lo que debe aparecer en la pantalla.

```
1 import Exponent from 'exponent';
2 import React from 'react';
3 import {Text} from 'react-native';
4
5 class App extends React.Component {
6   render() {
7     return (
8       <Text>
9         Hola React Native + Exponent
10      </Text>
11    );
12  }
13 }
14
15 Exponent.registerRootComponent(App);
16
```

## 2.-Explique cómo se realiza la navegación entre screens en React Native.

R: En react native hace falta la integración de los componente navigator, native y stack, para lograr una correcta navegación, con ello podemos organizar los Screens disponibles en nuestro proyecto a un ruta definida por nosotros en un Main navigator al que podremos consultar mediante el comando navigate y posteriormente la ruta deseada previamente definida.

### 3.-Qué significa IaaS, PaaS y SaaS

R: IaaS: Es un método para ofrecer funcionalidades de computación, almacenamiento, redes y de otros tipos a través de Internet. La IaaS permite a las empresas utilizar sistemas de funcionamiento, aplicaciones y almacenamiento basados en la web sin tener que comprar, administrar y brindar soporte a la infraestructura de nube subyacente.

PaaS: Es una oferta de computación en nube que proporciona a los usuarios un entorno de nube en el que pueden desarrollar, gestionar y entregar aplicaciones. Los usuarios, además del almacenamiento y de otros recursos de computación, pueden utilizar un conjunto de herramientas preconstruidas para desarrollar, personalizar y probar sus propias aplicaciones.

SaaS: Es un modelo de distribución de software en el que tanto el software como los datos manejados son centralizados y alojados en un único servidor externo a la empresa.

## 4.-Que es Firebase, Firestore y explique a que se refiere cuando se habla de Baas.

R:Firebase: 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.

Firestore: Se podría decir que es la base de datos de firebase, donde son almacenados los datos para ser posteriormente usados en aplicaciones y programas mediante el empleo de consultas en tiempo real .

Backend as Service: BaaS es todo un conjunto de utilerías ya implementadas en la nube, en la cual solo nos preocupamos por desarrollar las funciones de negocio o servicios que necesitará nuestra aplicación para funcionar, de tal forma que, en lugar de desplegar aplicaciones, desplegamos funciones que posteriormente será expuestas como servicios para ser consumidas por la red.

5.-Defina o explique si React es lo mismo que React Native. Si son distintos liste cuales son las diferencias.

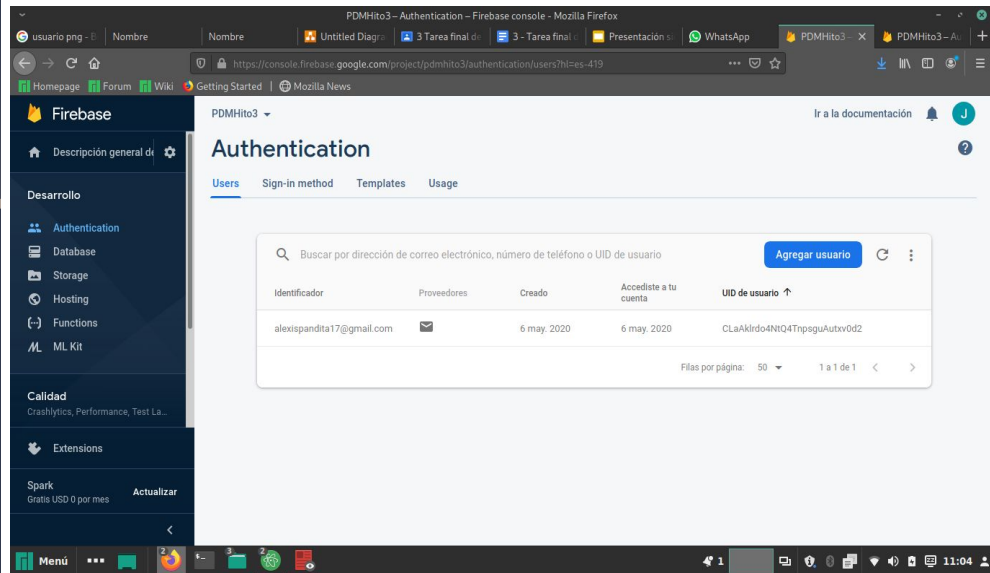
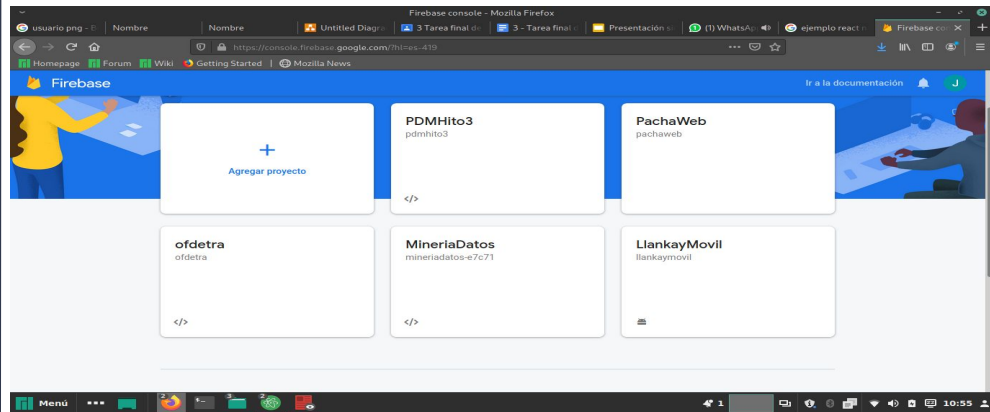
R: No son lo mismo, a pesar que react forme la base para lo que es react native, React es una librería Javascript focalizada en el desarrollo de interfaces de usuario, así se define la propia librería y evidentemente, esa es su principal área de trabajo. Mientras que react Native es un marco móvil que se compila en componente de aplicaciones nativas, lo que le permite crear aplicaciones móviles nativas para diferentes plataformas en JavaScript que le permite usar ReactJS para crear sus componentes.

# Parte Práctica

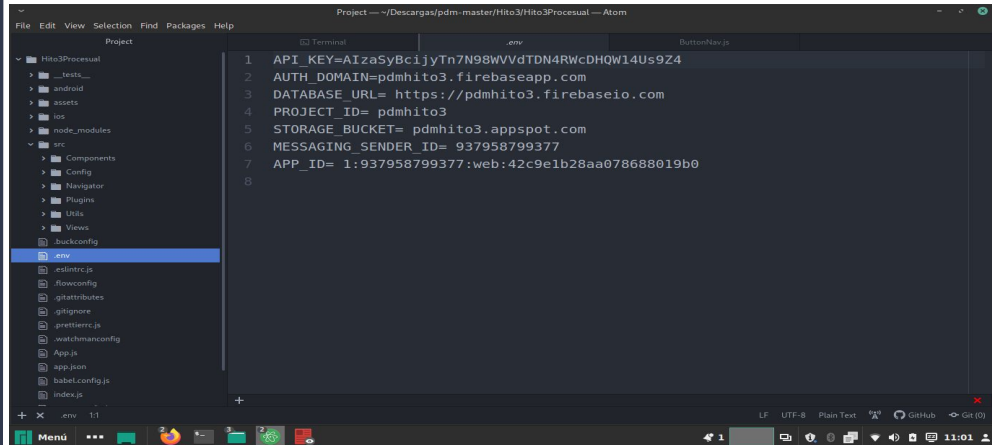
A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.



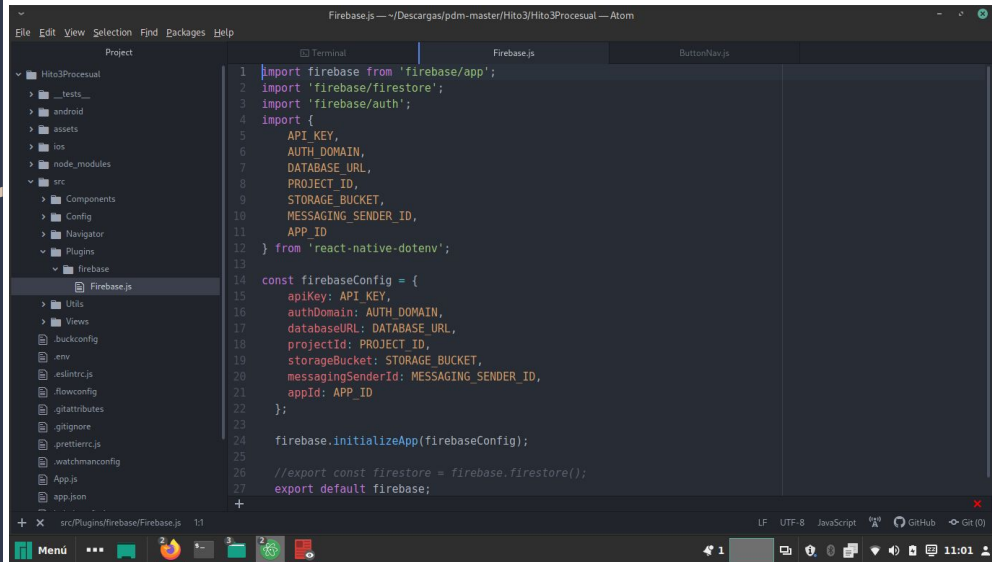
# 1. Creación de un nuevo proyecto en FIREBASE.



## 2.-Configuración de React native con Firebase para la integración.



```
1 API_KEY=AIzaSyBciJyTn7N98WVvDn4RwCDHQW14Us9Z4
2 AUTH_DOMAIN=pdmhito3.firebaseio.com
3 DATABASE_URL= https://pdmhito3.firebaseio.com
4 PROJECT_ID= pdmhito3
5 STORAGE_BUCKET= pdmhito3.appspot.com
6 MESSAGING_SENDER_ID= 937958799377
7 APP_ID= 1:937958799377:web:42c9e1b28aa078688019b0
8
```



```
1 import firebase from 'firebase/app';
2 import 'firebase/firestore';
3 import 'firebase/auth';
4 import {
5   API_KEY,
6   AUTH_DOMAIN,
7   DATABASE_URL,
8   PROJECT_ID,
9   STORAGE_BUCKET,
10  MESSAGING_SENDER_ID,
11  APP_ID
12 } from 'react-native-dotenv';
13
14 const firebaseConfig = {
15   apiKey: API_KEY,
16   authDomain: AUTH_DOMAIN,
17   databaseURL: DATABASE_URL,
18   projectId: PROJECT_ID,
19   storageBucket: STORAGE_BUCKET,
20   messagingSenderId: MESSAGING_SENDER_ID,
21   appId: APP_ID
22 };
23
24 firebase.initializeApp(firebaseConfig);
25
26 //export const firestore = firebase.firestore();
27 export default firebase;
```

# Para realizar todas las vistas se usaron:

**ButtonNav:** Es un TouchableOpacity que recibirá una variable onPress y una variable que definirá el texto del botón, de color transparente

**ButtonFoter:** un par de botones importados desde Button, ordenados en forma de fila con la propiedad row, a los cuales también se le encadenara las propiedades de título y onPress

**Description:** contendrá la imagen y el texto de cada Screen junto con un logo que cambia según el Screen

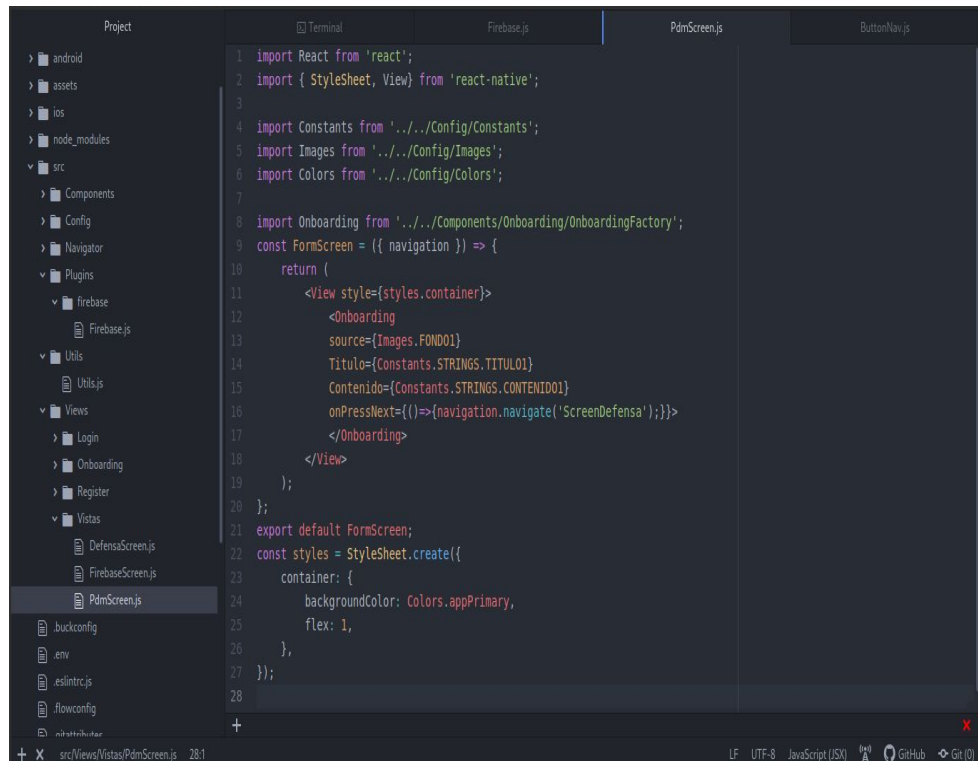
**onboardingFactory:** Es el componente donde se importan y usan los demás componentes antes mencionados y se les asigna valores a los componentes como el onPress de los botones, las imágenes de logo, los textos de la descripción, los títulos de los botones prev y next.



### 3.-Creación de el componente PDM (1ra vista)



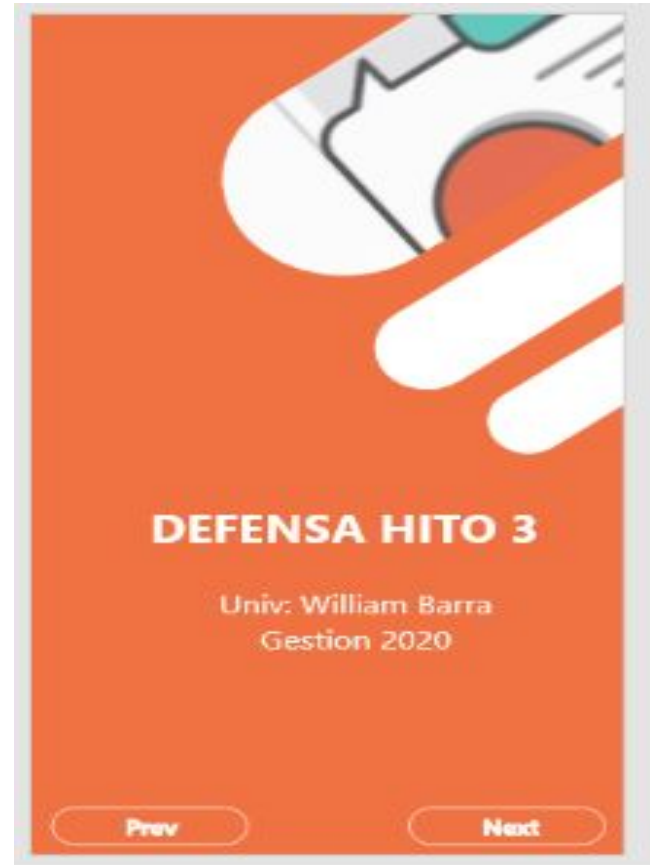
# Solución:



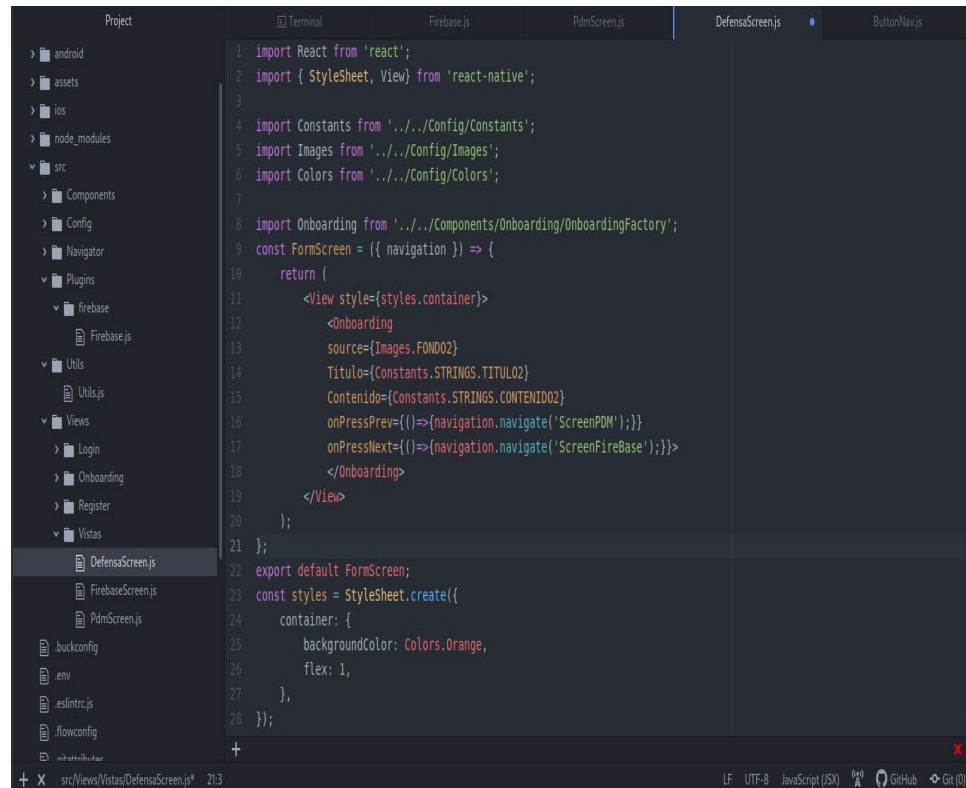
The screenshot shows a code editor with a project explorer on the left and a code editor on the right. The project explorer shows a directory structure for a React Native application. The code editor shows the implementation of the `PdmScreen.js` file. The code imports `React`, `StyleSheet`, `View`, `Constants`, `Images`, and `Colors`. It then defines a `FormScreen` component that uses `Onboarding` from a factory. The component renders an `Onboarding` view with a source image, a title, and content. It also has an `onPressNext` handler that navigates to the `ScreenDefensa` screen. The component is styled with a primary background color and a flex layout.

```
1 import React from 'react';
2 import { StyleSheet, View } from 'react-native';
3
4 import Constants from '../../Config/Constants';
5 import Images from '../../Config/Images';
6 import Colors from '../../Config/Colors';
7
8 import Onboarding from '../../Components/Onboarding/OnboardingFactory';
9 const FormScreen = ({ navigation }) => {
10   return (
11     <View style={styles.container}>
12       <Onboarding
13         source={Images.FOND01}
14         Titulo={Constants.STRING.TITULO1}
15         Contenido={Constants.STRING.CONTENIDO1}
16         onPressNext={()=>navigation.navigate('ScreenDefensa')}>
17       </Onboarding>
18     </View>
19   );
20 };
21 export default FormScreen;
22 const styles = StyleSheet.create({
23   container: {
24     backgroundColor: Colors.appPrimary,
25     flex: 1,
26   },
27 });
28
```

## 4.-Creación del componente DEFENSAHITO3 (2da Vista)



# Solución:



The image shows a screenshot of an IDE with a project structure on the left and a code editor on the right. The project structure includes folders for android, assets, ios, node\_modules, src, Components, Config, Navigator, Plugins, firebase, Utils, Views, Login, Onboarding, Register, Vistas, and files like DefensaScreen.js, FirebaseScreen.js, PdmScreen.js, buckconfig, env, eslintrc.js, flowconfig, and .metro.config. The code editor displays the content of DefensaScreen.js, which is a React Native component for an onboarding screen. It imports React, StyleSheet, View, Constants, Images, Colors, and Onboarding. It defines a FormScreen component that takes navigation as a prop and returns a View containing an Onboarding component. The Onboarding component has a title, content, and two onPress handlers for navigation. The FormScreen is exported as the default export. The styles are defined with a container style that has an orange background color and a flex of 1.

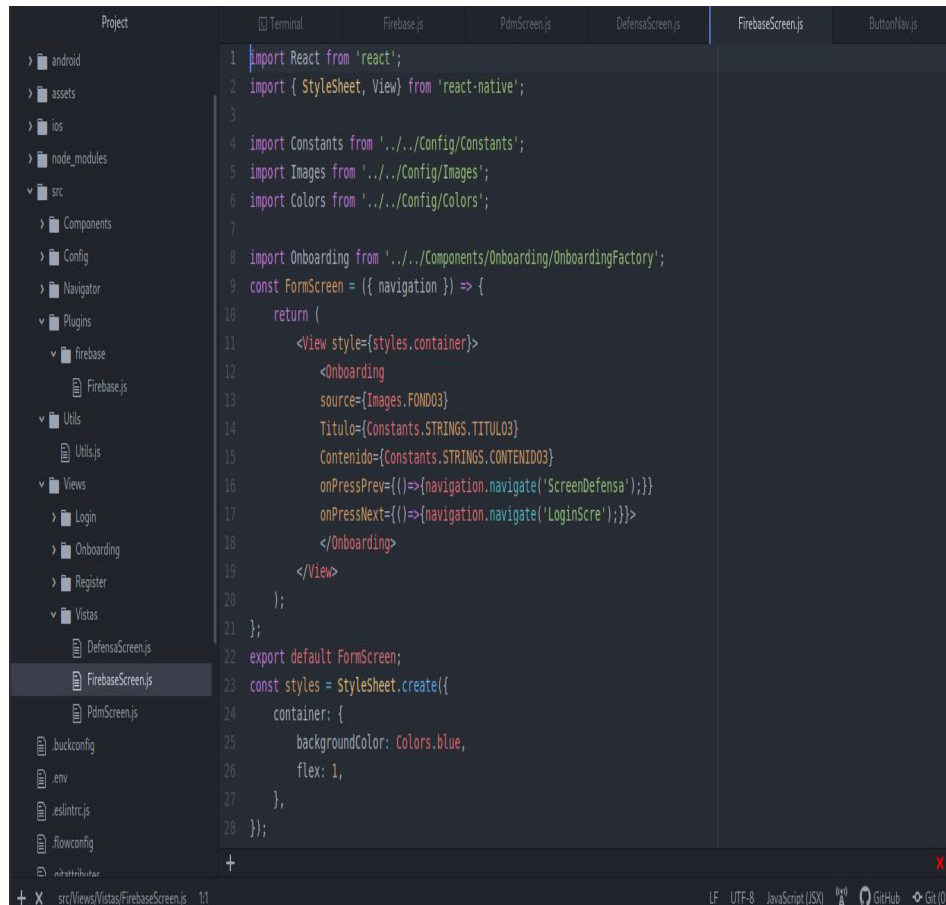
```
1 import React from 'react';
2 import { StyleSheet, View } from 'react-native';
3
4 import Constants from '../Config/Constants';
5 import Images from '../Config/Images';
6 import Colors from '../Config/Colors';
7
8 import Onboarding from '../Components/Onboarding/OnboardingFactory';
9 const FormScreen = ({ navigation }) => {
10   return (
11     <View style={styles.container}>
12       <Onboarding
13         source={Images.FOND02}
14         Titulo={Constants.STRING.TITULO2}
15         Contenido={Constants.STRING.CONTENTID02}
16         onPressPrev={()=>{navigation.navigate('ScreenPDM')}}
17         onPressNext={()=>{navigation.navigate('ScreenFireBase')}}
18       />
19     </View>
20   );
21 };
22 export default FormScreen;
23 const styles = StyleSheet.create({
24   container: {
25     backgroundColor: Colors.Orange,
26     flex: 1,
27   },
28 });
```

## 5.-Creación del componente FIREBASE (3ra Vista)





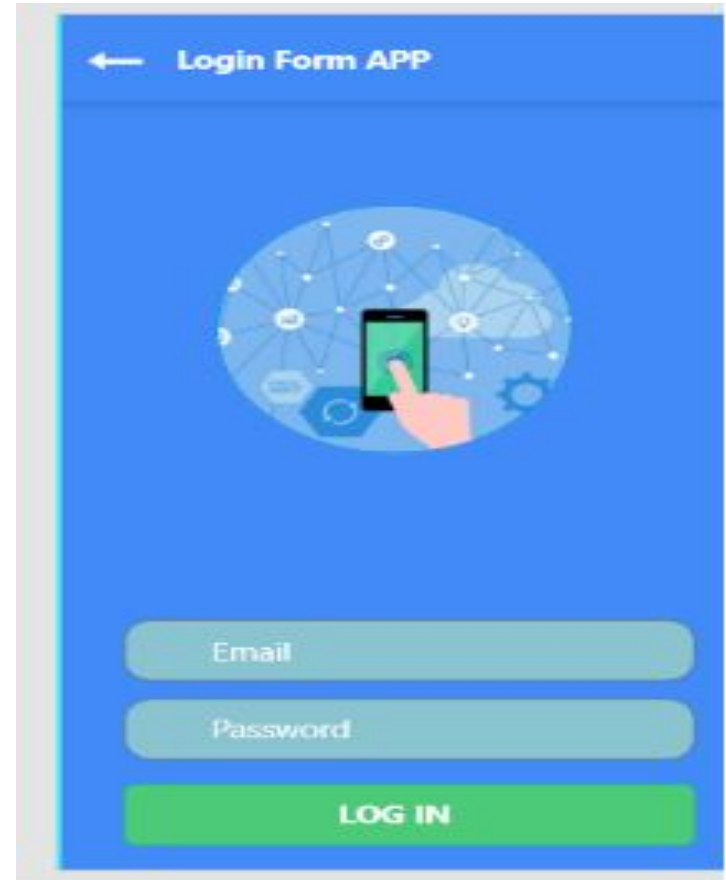
# Solución:



The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes folders for android, assets, ios, node\_modules, src, Components, Config, Navigator, Plugins, firebase, firebase.js, Utils, Utils.js, Views, Login, Onboarding, Register, Vistas, DefensaScreen.js, FirebaseScreen.js, PdmScreen.js, .buckconfig, .env, .eslintrc.js, .flowconfig, and .gitattributes. The code editor displays the content of FirebaseScreen.js, which imports React, StyleSheet, View, Constants, Images, Colors, and Onboarding. It defines a FormScreen component that returns a View containing an Onboarding screen with specific source, title, and content, and navigation handlers for 'ScreenDefensa' and 'LoginScre'. The component is styled with a blue background and a flex layout.

```
1 import React from 'react';
2 import { StyleSheet, View } from 'react-native';
3
4 import Constants from '../Config/Constants';
5 import Images from '../Config/Images';
6 import Colors from '../Config/Colors';
7
8 import Onboarding from '../Components/Onboarding/OnboardingFactory';
9 const FormScreen = ({ navigation }) => {
10   return (
11     <View style={styles.container}>
12       <Onboarding
13         source={Images.FOND03}
14         Titulo={Constants.STRING.S.TITULO3}
15         Contenido={Constants.STRING.S.CONTENTID03}
16         onPressPrev={()=>{navigation.navigate('ScreenDefensa')}}
17         onPressNext={()=>{navigation.navigate('LoginScre')}}
18       </Onboarding>
19     </View>
20   );
21 };
22 export default FormScreen;
23 const styles = StyleSheet.create({
24   container: {
25     backgroundColor: Colors.blue,
26     flex: 1,
27   },
28 });
```

## 6.-Creación del componente LOGIN (4ta vista)



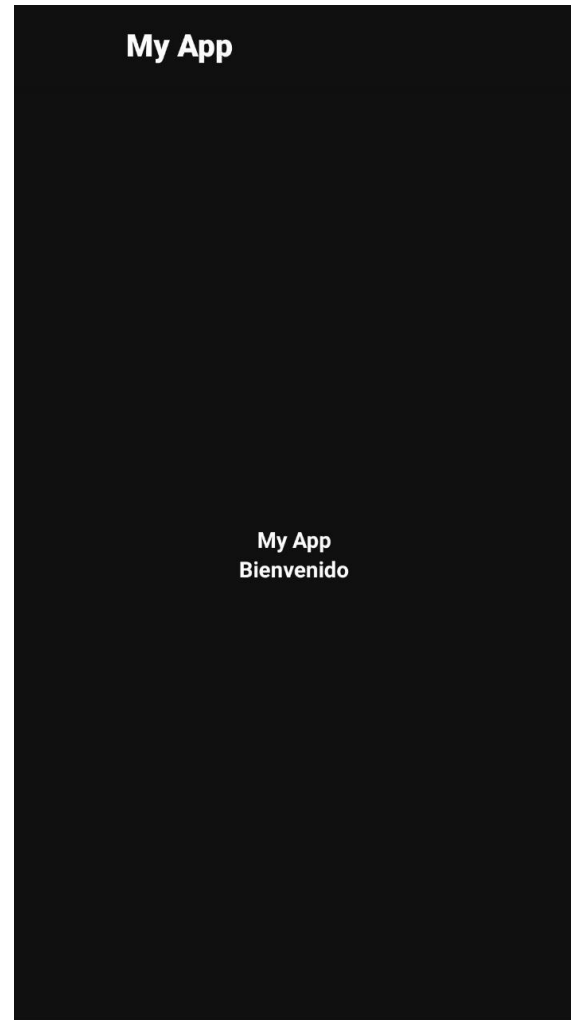
# Solución:

```
47 //DESDE AQUÍ
48 const loginApp = (email, password) => {
49   try {
50     setIsLoading(true);
51     FirebasePlugin.auth()
52       .signInWithEmailAndPassword(email, password)
53       .then(user => {
54         setIsLoading(false);
55         navigation.navigate('Register');
56       })
57       .catch(error => {
58         FirebasePlugin.auth()
59           .createUserWithEmailAndPassword(email, password)
60           .then(user => {
61             setIsLoading(false);
62             navigation.navigate('Register');
63           })
64           .catch(error => {
65             setIsLoading(false);
66             Alert.alert('Invalid Values', error.message);
67           });
68       });
69   } catch (error) {
70     setIsLoading(true);
71     Alert.alert('Firebase Error', error.message);
72   }
73 };
74 //HASTA AQUÍ
75 return (
76   <DismissKeyboard>
77     <AuthScreen>
78   </DismissKeyboard>
79 );
```

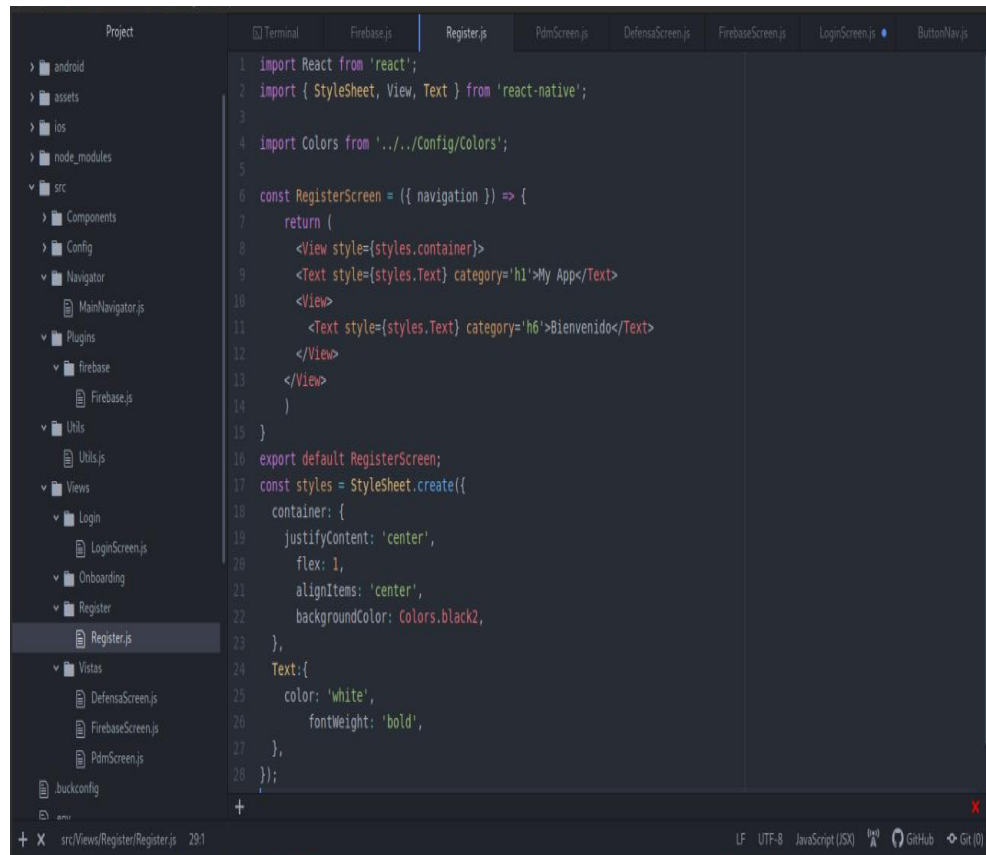
src/Views/LoginScreen.js\* 67:28

LF UTF-8 JavaScript (JSX) GitHub Git (0)

7.-reacción del  
componente MIAPP  
(Vista que se  
mostrara después  
de hacer LOGIN).



# Solución:



```
1 import React from 'react';
2 import { StyleSheet, View, Text } from 'react-native';
3
4 import Colors from '../Config/Colors';
5
6 const RegisterScreen = ({ navigation }) => {
7   return (
8     <View style={styles.container}>
9       <Text style={styles.Text} category='h1'>My App</Text>
10      <View>
11        <Text style={styles.Text} category='h6'>Bienvenido</Text>
12      </View>
13    </View>
14  )
15 }
16 export default RegisterScreen;
17 const styles = StyleSheet.create({
18   container: {
19     justifyContent: 'center',
20     flex: 1,
21     alignItems: 'center',
22     backgroundColor: Colors.black2,
23   },
24   Text: {
25     color: 'white',
26     fontWeight: 'bold',
27   },
28 });
```