

# **Documentación de trabajo React Native**

## **Autora de documentación**

Anyi Johana Rugelis Vásquez

## **Trabajo desarrollado por**

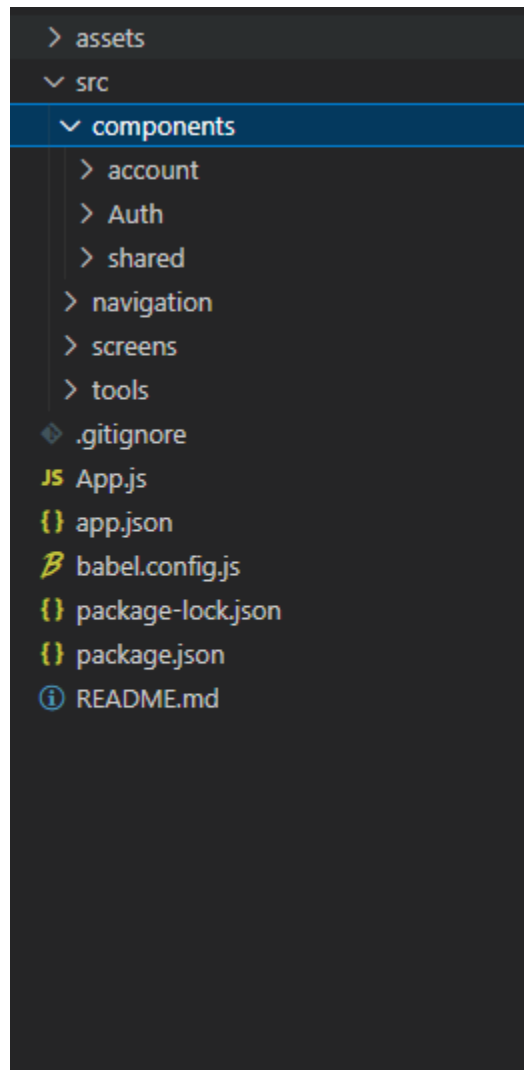
Santiago Ospina Rojas

Anyi Johana Rugelis Vásquez

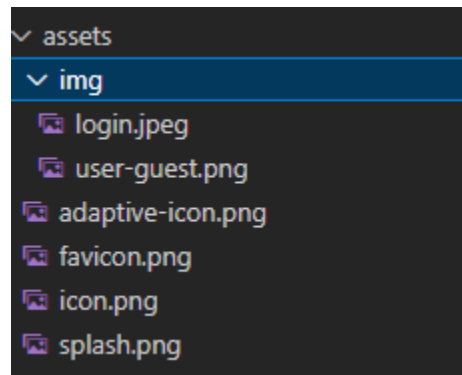
## **FECHA**

14/05/2023

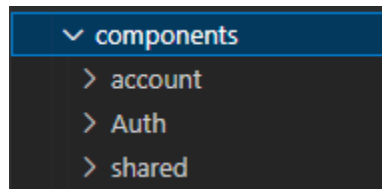
## **Estructura del sistema**



Assets and img

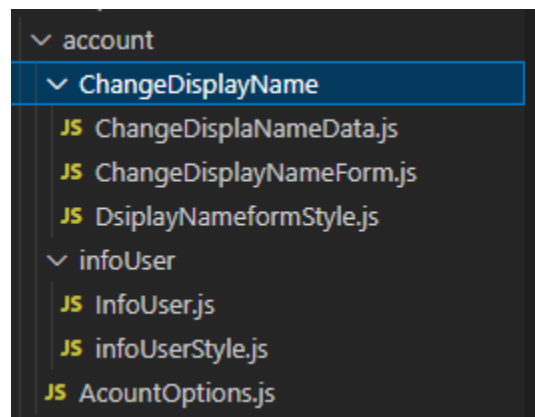


### Components



### Account

Contiene multiples carpetas las cuales mostraremos a continuación



JS ChangeDisplaNameData.js X

src > components > account > ChangeDisplayName > JS ChangeDisplaNameData.js > ...

```
1 import * as Yup from "yup";
2
3 export function initialValues(){
4   return{
5     displayName:""
6   }
7 }
8
9 export function validationSchema(){
10   return Yup.object({
11     displayName:Yup.string().required("el nombre y apellido son requeridos")
12   })
13 }
```

JS ChangeDisplayNameForm.js X

src > components > account > ChangeDisplayName > JS ChangeDisplayNameForm.js > ...

```
1 import React, { useState } from "react";
2 import { View } from "react-native";
3 import { Button, Icon, Input, Text } from "react-native-elements";
4 import { styles } from "../DisplaynameformStyle";
5 import { useFormik } from "formik";
6 import { initialValues, validationSchema } from "../ChangeDisplaNameData";
7 import { getAuth, updateProfile } from "firebase/auth";
8 import { Toast } from "react-native-toast-message/lib/src/Toast";
9 export const ChangeDisplayNameForm = ({close,onReload}) => {
10   const formik = useFormik({
11     initialValues: initialValues(),
12     validationSchema: validationSchema(),
13     validateOnChange: false,
14     onSubmit:async (formValue) => {
15       try {
16         const { displayName } = formValue;
17         const currentUser = getAuth().currentUser;
18         await updateProfile(
19           currentUser,{displayName}
20         );
21         onReload();
22         close(false)
23       } catch (error) {
24         Toast.show({
25           type: "error",
26           position: "bottom",
27           text1: "Error al cambiar el nombre y apellido",
28         });
29       }
30     },
31   });
32   return /
```

tos para un mejor rendimiento de TypeScript IntelliSense tabnine starter Lín. 1, col. 1 Espacios: 2 UTF-8 CRLF JavaScript Go Live Prettier

5 DsiplayNameformStyle.js X

rc > components > account > ChangeDisplayName > JS DsiplayNameformStyle.js > ...

```
1  import { StyleSheet } from "react-native";
2
3  export const styles= StyleSheet.create({
4    content:{
5      alignItems:"center",
6      justifyContent:"center",
7      paddingVertical:10
8    },
9    input:{
10     width:"100%",
11     marginTop:20
12   },
13   icon:{
14     color:"#c1c1c1"
15   },
16   btnButton:{
17     backgroundColor:"#00a680"
18   },
19   btn:{
20     width:"95%",
21     marginTop:10
22   },
23
24  });
```

```
> components > account > infoUser > JS InfoUser.js > ...
1 import React, { useState } from "react";
2 import { View } from "react-native";
3 import { Avatar, Text } from "react-native-elements";
4 import { getAuth, updateProfile } from "firebase/auth";
5 import { getDownloadURL, getStorage, ref, uploadBytes } from "firebase/storage";
6 import { styles } from "../infoUserStyle";
7 import * as ImagePicker from "expo-image-picker";
8 export const InfoUser = () => {
9   const { uid, photoURL, displayName, email } = getAuth().currentUser;
10   const [avatar, setAvatar] = useState(photoURL);

11
12   const handleChangeAvatar = async () => {
13     const result = await ImagePicker.launchImageLibraryAsync({
14       mediaTypes: ImagePicker.MediaTypeOptions.All,
15       allowsEditing: true,
16       aspect: [4, 3]
17     });
18     if (!result.canceled) uploadImage(result.assets[0].uri)
19   };
20
21   const uploadImage = async (uri) => {
22     const response = await fetch(uri);
23     const blob = await response.blob();
24     const storage = getStorage();
25     const storageRef = ref(storage, `avatar/${uid}`);
26     uploadBytes(storageRef, blob).then((snapshot) => {
27       updatePhotoUrl(snapshot.metadata.fullPath)
28     });
29   }
30 }
```

components > account > infoUser > JS infoUserStyle.js > ...

```
import { StyleSheet } from "react-native";
```

```
export const styles= StyleSheet.create({  
  content:{  
    alignItems:"center",  
    justifyContent:"center",  
    flexDirection:"row",  
    backgroundColor:"#f2f2f2",  
    paddingVertical:30  
  },  
  avatar:{  
    marginRight:20,  
    backgroundColor:"green",  
  },  
  displayname:{  
    fontWeight:"bold",  
    paddingBottom:5  
  }  
});
```



```
terminal Ayuda AccountOptions.js - area-react - visual studio code
JS AccountOptions.js X
src > components > account > JS AccountOptions.js > ...
11 const selectedComponent = (key) => {
12   if (key === "displayName") {
13     setRenderComponent(<ChangeDisplayNameForm onReload={onReload} close={setShow} />)
14   }
15   if (key === "Email") {
16     setRenderComponent(<Text>cambiando email</Text>)
17   }
18   }
19   if (key === "password") {
20     setRenderComponent(<Text>cambiando password</Text>)
21   }
22   }
23   }
24   }
25   setShow(true)
26 };
27 const menuOptions = getMenuOptions(selectedComponent);
28
29 return (
30   <View>
31     {menuOptions.map((menu, index) => (
32       <ListItem key={index} bottomDivider onPress={menu.onPress}>
33         <Icon
34           type={menu.iconType}
35           name={menu.iconNameLeft}
36           color={menu.iconColorLeft}
37         />
38         <ListItem.Content>
39           <ListItem.Title>{menu.title}</ListItem.Title>
40         </ListItem.Content>
41         <Icon
42           type={menu.iconType}
43           name={menu.iconNameRight}

```

- ✓ LoginForm
- JS LoginForm.js
- JS LoginFormStyle.js
- ✓ RegisterForm
- JS RegisterFormData.js
- JS RegisterForm.js
- JS RegisterFormStyle.js

```
JS RegisterForm.js X
src > components > Auth > RegisterForm > JS RegisterForm.js > ...
1 import { View } from "react-native";
2 import React, { useState } from "react";
3 import { styles } from "../RegisterFormStyle";
4 import { Button, Icon, Input } from "react-native-elements";
5 import { getAuth, createUserWithEmailAndPassword } from "firebase/auth";
6 import { useNavigation } from "@react-navigation/native";
7 import Toast from "react-native-toast-message";
8 export const RegisterForm = () => {
9   const [email, setEmail] = useState("");
10   const [password, setPassword] = useState("");
11   const [repetpassword, setRepetpassword] = useState("");
12   const [show, setShow] = useState(true);
13   const [showrepet, setShowrepet] = useState(true);
14   const [loading, setLoading] = useState(false);
15   const navigation = useNavigation();
16
17   const showChangePassword = () => {
18     setTimeout(() => setShow(true), 5000);
19     setShow(false);
20   };
21   const showChangeRepetPassword = () => {
22     setTimeout(() => setShowrepet(true), 5000);
23     setShowrepet(false);
24   };
25
26   const handleRegister = async () => {
27     setLoading(true);
28     try {
29       const auth = getAuth();
30       await createUserWithEmailAndPassword(auth, email, password);
31       navigation.navigate("Accounts");
32     } catch (error) {
```

```
1  import { StyleSheet } from "react-native";
2
3  export const styles= StyleSheet.create({
4    content:{
5      flex:1,
6      alignContent:"center",
7      justifyContent:"center",
8      marginTop:30
9    },
10   input:{
11     width:"100%",
12     marginTop:20
13   },
14   icon:{
15     color:"#c1c1c1"
16   },
17   btnButton:{
18     marginTop:20,
19     width:"95%",
20     marginHorizontal:10
21   },
22   btn:{
23     backgroundColor:"#00A680"
24   }
25 });
```

JS LoadingModal.js X

src > components > shared > LoadingModal > JS LoadingModal.js > ...

```
1  import { View, ActivityIndicator } from "react-native";
2  import React from "react";
3  import { Overlay, Text } from "react-native-elements";
4  import { styles } from "../LoadingModalStyle";
5
6  export const LoadingModal = ({ show, text }) => {
7    return (
8      <Overlay
9        isVisible={show}
10        // windowBackgroundColor="rba(0,0,0,0.5)"
11        // overlayBackgroundColor="transparent"
12        overlayStyle={styles.overlay}
13      >
14        <View style={styles.view}>
15          <ActivityIndicator size={"large"} color="#00A680" />
16          {text} && <Text style={styles.text}>{text}</Text>
17        </View>
18      </Overlay>
19    );
20  };
21
22  LoadingModal.defaultProps = {
23    show: false,
24  };
25
```

JS LoadingModalStyle.js X

src > components > shared > LoadingModal > JS LoadingModalStyle.js > ...

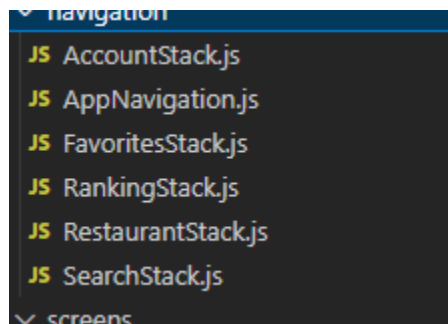
```
1  import { StyleSheet } from "react-native";
2
3  export const styles= StyleSheet.create({
4    overlay:{
5      height:100,
6      width:200,
7      backgroundColor:"#FFF",
8      borderColor:"#00A680",
9      borderWidth:2,
10     borderRadius:10,
11   },
12   view:{
13     flex:1,
14     alignItems:"center",
15     justifyContent:"center"
16   },
17   text:{
18     color:"#00A680",
19     textTransform:"uppercase",
20     marginTop:10
21   }
22 });
```

Ln 1, col 1 Espacios: 4 LITE-8 CRLE JavaScript @Go Live 89

```
JS Modal.js x
src > components > shared > Modal > JS Modal.js > ...
1  import React from "react";
2  import { Overlay } from "react-native-elements";
3  import { styles } from "../ModalStyles";
4
5  export const Modal = ({ show, close, children }) => {
6    return (
7      <Overlay
8        isVisible={show}
9        overlayStyle={styles.overlay}
10       onBackdropPress={close}
11      >
12        {children}
13      </Overlay>
14    );
15  };
16
```

## Navigation

Los navigation están agrupando las screen en las cuales se debe tener comunicación



AccountStack.js X

src > navigation > JS AccountStack.js > ...

```
1 import { createNativeStackNavigator } from "@react-navigation/native-stack";
2 import React from "react";
3 import { Account } from "../screens/Account/AccountScreen";
4 import { LoginScreen } from "../screens/Account/LoginScreen/LoginScreen";
5 import { RegisterScreen } from "../screens/Account/RegisterScreen/RegisterScreen";
6
7
8
9 export const AccountsStack = () =>{
10   const Stack = createNativeStackNavigator();
11   return (
12     <Stack.Navigator>
13       <Stack.Screen
14         name="Accounts"
15         component={Account}
16         options={{ title: "Cuenta" }}
17       />
18       <Stack.Screen
19         name="Login"
20         component={LoginScreen}
21         options={{ title: "Iniciar sesion " }}
22       />
23       <Stack.Screen
24         name="Register"
25         component={RegisterScreen}
26         options={{ title: "Crear tu cuenta " }}
27       />
28     </Stack.Navigator>
29   )
30 };
31
```

JS FavoritesStack.js X

src > navigation > JS FavoritesStack.js > ...

```
1  import { createNativeStackNavigator } from "@react-navigation/native";
2  import React from "react";
3  import { FavoritesScreen } from "../screens/FavoritesScreen";
4
5  export const FavoritesStack = () =>{
6    const Stack = createNativeStackNavigator();
7    return (
8      <Stack.Navigator>
9        <Stack.Screen
10          name="Favoritos"
11          component={FavoritesScreen}
12          options={{ title: "Favoritos" }}
13        />
14      </Stack.Navigator>
15    )
16  };
17
```



JS RestaurantStack.js X

src > navigation > JS RestaurantStack.js > ...

```
1  import { createNativeStackNavigator } from "@react-navigation/native-stack";
2  import React from "react";
3  import { AddRestaurant } from "../screens/Restaurants/AddRestaurant";
4  import { RestaurantsScreen } from "../screens/Restaurants/RestaurantsScreen";
5
6  export const RestaurantStack = () => {
7    const Stack = createNativeStackNavigator();
8    return (
9      <Stack.Navigator>
10        <Stack.Screen
11          name="Restaurantes"
12          component={RestaurantsScreen}
13          options={{ title: "Restaurantes" }}
14        />
15        <Stack.Screen
16          name="AddRestaurants"
17          component={AddRestaurant}
18          options={{ title: "Agregar Restaurantes" }}
19        />
20      </Stack.Navigator>
21    );
22  };
23
24
```

```

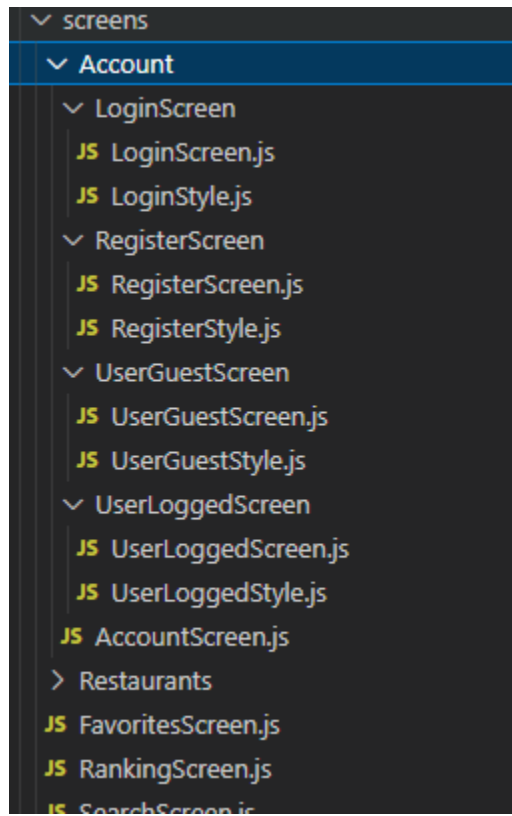
5 AppNavigation.js X
src > navigation > JS AppNavigation.js > ...
11 const Tab = createBottomTabNavigator(),
12
13 export function AppNavigation() {
14   return (
15     <Tab.Navigator
16       screenOptions={({ route }) => ({
17         tabBarActiveTintColor: "#00A680",
18         tabBarInactiveTintColor: "#646464",
19         tabBarIcon: ({ color, size }) => screenIcons(route, color, size),
20       })
21     >
22     <Tab.Screen
23       name={"Restaurants"}
24       component={RestaurantStack}
25       options={{ title: "Restaurantes", headerShown: false }}
26     />
27     <Tab.Screen
28       name="Ranking"
29       component={RankingStack}
30       options={{ title: "Ranking", headerShown: false }}
31     />
32
33     <Tab.Screen
34       name="Favorites"
35       component={FavoritesStack}
36       options={{ title: "Favoritos", headerShown: false }}
37     />
38
39     <Tab.Screen
40       name="Account"
41       component={AccountsStack}
42       options={{ title: "Cuenta", headerShown: false }}
43     />

```

Conexión con firebase

```
JS firebase.js X
src > tools > JS firebase.js > ...
1
2 import { initializeApp } from "firebase/app";
3
4 const firebaseConfig = {
5   apiKey: "AIzaSyB6yEW3Iuipp54eqQmQP_hcMkRek5vUrM4",
6   authDomain: "restaurante-v27.firebaseio.com",
7   projectId: "restaurante-v27",
8   storageBucket: "restaurante-v27.appspot.com",
9   messagingSenderId: "961191275023",
10  appId: "1:961191275023:web:25830f67e527ca0041d17a"
11 };
12
13 export const initFireb = initializeApp(firebaseConfig);
```

Screens



El resto de las screen se pueden visualizar en el github