PROCESUAL HITO 3

React Native

RAQUEL TERRAZAS

CONTENIDOS

RESUMEN

ANÁLISIS DE LOS PROBLEMAS

DESARROLLO

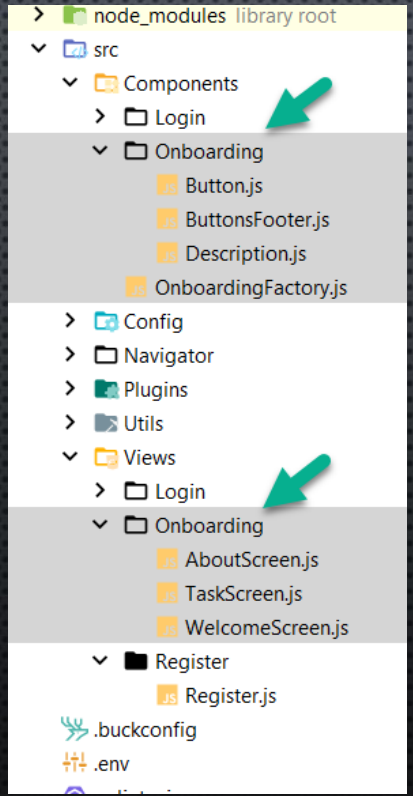
RESUMEN

La presentación actual, mostrará el proyecto propuesto para el hito 3 de la materia de programación de dispositivos móviles, con parte de la explicación para comprender el funcionamiento del mismo.

Análisis del problema

El objetivo es realizar una integración entre las tecnologías React Native y Firebase para poder gestionar autenticación de usuarios y gestión de bases de datos en tiempo real.
Adicionalmente la aplicacion movil multiplataforma debera de tener un Onboarding antes de llegar a la pagina de LOGIN.

ANÁLISIS DEL PROBLEMA (EXPLICACIÓN TÉCNICA)



ANÁLISIS DEL PROBLEMA (EXPLICACIÓN TÉCNICA)

React

Defensa Hito 3 - Caso de uso: Onboarding

- Crear el componente Button.
- Crear el componente Description.
- Crear el componente ButtonsFooter.
- Crear el componente OnboardingFactory.
- Crear los SCREENS para el navigator y verificar el login a FIREBASE.
  - Crear la vista AboutScreen.
  - Crear la vista TaskScreen.
  - Crear la vista WelcomeScreen.
  - Verificar el login Firebase.

DESARROLLO (BUTTON)

```
import React from 'react';
import {
  TouchableOpacity,
  StyleSheet,
  Text,
  View,
  ActivityIndicator,
} from 'react-native';
import Colors from '../../Config/Colors';
const Button = ({titleButton, onPress, isLoading}) => {
  const loader = () => {
    return <ActivityIndicator animating={isLoading} />;
  };
  const button = () => {
    return (
      <TouchableOpacity onPress={onPress}>
        <Text style={styles.text}>{titleButton}</Text>
      </TouchableOpacity>
    );
  };
  return <View style={styles.button}>{isLoading ? loader() : b
utton()}</View>;
};
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Para que se pueda notar la animación al hacer click, se configura el button

Se hace un return al componente para que se lo pueda exportar

DESARROLLO (BUTTON)

```
const styles = StyleSheet.create({
  button: {
    display: 'flex',
    height: 30,
    borderRadius: 20,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: 'transparent',
    shadowColor: Colors.blue,
    shadowOpacity: 0.4,
    shadowOffset: {height: 10, width: 10},
    shadowRadius: 20,
    borderColor: Colors.white,
    borderWidth: 1,
  },
  text: {
    width: 100,
    color: Colors.white,
    textAlign: 'center',
    fontWeight: 'bold',
    height: 20,
  },
});

export default Button;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

DESARROLLO DE BUTTON LOGRADO

DESARROLLO (DESCRIPTION)

Se hace un return al componente para que se lo pueda exportar

```
import React, { Component } from 'react';
import { StyleSheet, View, TextInput, Text, Image, TextBase } from 'react-native';
import PropTypes from 'prop-types';
import Colors from '../../Config/Colors';
import Button from '../../Components/login/ButtonPyN'
import Constants from '../../Config/Constants';
import Images from '../../Config/Images';
const Descripcion = ({ source, textoDesc, textoTit,navigation }) => {
    return (
        <View style={styleDescripcion.container}>
            <Image  style={styleDescripcion.image} source={source} />

            <View >
                <Text style={styleDescripcion.text1}>
                    {textoTit}
                </Text>
                <Text style={styleDescripcion.text}
                multiline={true}>
                    {textoDesc}
                </Text>
            </View>
        </View>
    );
}
```

Se importan los componentes necesarios de react-native

Se importa los componentes que se usarán, colores, constantes e imágenes

Se declara el constructor junto con las propiedades a exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (DESCRIPTION)

```
const styleDescripcion = StyleSheet.create({
    container: {
        flex: 3,
        top:-30,
        marginBottom: 40,
        alignItems: 'center',
        justifyContent: 'center',
        backgroundColor: 'transparent',
    },
    image: {
        width: 400,
        height: 500,
        top: -140,
        right: -100,
    },
    text1: {
        alignSelf: 'center',
        color: 'white',
        fontWeight: 'bold',
        backgroundColor: 'transparent',
        marginTop: 20,
        fontSize:30,
    },
```

Se crean constantes para asignar estilos

```
text: {
        width: Constants.CONFIG.SCREEN_WIDTH
 * 0.75,
        alignSelf: 'center',
        textAlign:'center',
        color: 'white',
        fontSize:18,
        fontFamily:'times',
        backgroundColor: 'transparent',
        margin: 20,
    }
export default Descripcion;
```

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

COMPONENTE DESCRIPTION LOGRADO

DESARROLLO (BUTTONSFOOTER)

```
import React, { Component } from 'react';
import { StyleSheet, View, TextInput, Text, Image } from
 'react-native';
import PropTypes from 'prop-types';
import Colors from '../../Config/Colors';
import Button from './ButtonPyN'
import Constants from '../../Config/Constants';

const ButtonsFooter = ({  onPressP,onPressN, isLoading, n
avigation }) => {
    return (
        <View style={styleButtonsFooter.container}>
            <Button
                isLoading={isLoading}
                onPress={onPressP}
                titleButton={Constants.STRINGS.PREV}>
            </Button>
            <Button
                isLoading={isLoading}
                onPress={onPressN}
                titleButton={Constants.STRINGS.NEXT}>
            </Button>
        </View>
    );
}
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Se hace un return al componente para que se lo pueda exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (BUTTON)

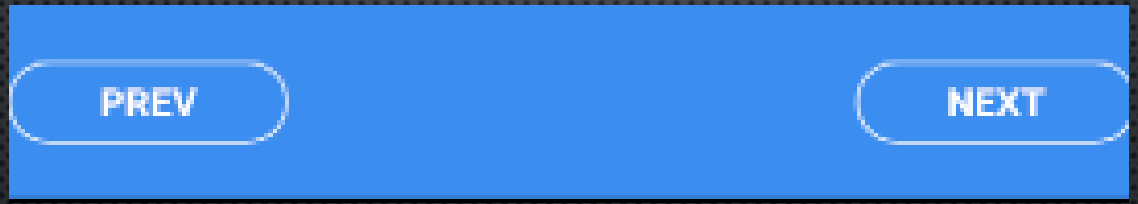
```
const styleButtonsFooter = StyleSheet.create({
    container: {
        flexDirection: 'row',
        backgroundColor: 'transparent',
        alignItems: 'baseline',
        justifyContent: 'space-between',
    },

});

export default ButtonsFooter;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

COMPONENTE BUTTONSFOOTER LOGRADO



DESARROLLO (ONBOARDINGFACTORY)

```
import React ,{ useState } from 'react';
import { StyleSheet, View } from 'react-native';
import Colors from '../Config/Colors';
import ButtonFooter from '../Components/Onboarding/ButtonsFooter';
import Descripcion from '../Components/Onboarding/Descripcion';
const OnboardingFactory = ({ navigation,textoDesc,textoTit,source,o
nPressN,onPressP,isLoading }) => {
  return (
    <View style={styles.container}>
      <Descripcion
      source={source}
      style={styles.desc}
      textoTit={textoTit}
      textoDesc={textoDesc}>

      </Descripcion>
      <ButtonFooter
        isLoading={isLoading}
        onPressP={onPressP}
        onPressN={onPressN}>
      </ButtonFooter>
    </View>
  );
};
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Se hace un return al componente para que se lo pueda exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (ONBOARDINGFACTORY)

```
const styles = StyleSheet.create({
  container: {
    flex: 3,
    paddingBottom:10,
    backgroundColor: 'transparent',
    justifyContent: 'space-between',
},
  text: {
    color: Colors.white,
    textAlign: 'center',
    fontWeight: 'bold',
    height: 20,
  },
  desc:{
    textAlign:'center',
  }
});

export default OnboardingFactory;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

DESARROLLO ONBOARDINGFACTORY LOGRADO

React

DESARROLLO (SCREENS ABOUTSCREEN)

```
import React, { useState } from 'react';
import { StyleSheet, View, TextInput, Text, Image } from 'react-native';
import Colors from '../../Config/Colors';
import Images from '../../Config/Images';
import OnboardingFactory from '../../Components/OnboardingFactory';

const AboutScreen = ({ navigation }) => {
  const [isLoading, setIsLoading] = useState(false);
  const _onPressP = () => {
    navigation.navigate('Pdm');
  };
  const _onPressN = () => {
    navigation.navigate('Firebase');
  };
  return (
    <View style={styles.container}>
      <OnboardingFactory
        source={Images.IMG2}
        style={styles.desc}
        textoTit={'DEFENSA HITO3'}
        textoDesc={'Univ. William Barra Gestión 2020'}
        isLoading={isLoading}
        onPressP={_onPressP}
        onPressN={_onPressN}
      >
      </OnboardingFactory>
    </View>
  );
};
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Se crean los eventos

Se hace un return al componente para que se lo pueda exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (SCREENS ABOUTSCREEN)

```
const styles = StyleSheet.create({
  container: {
    flex: 3,
    paddingBottom: 10,
    backgroundColor: Colors.orange,
    justifyContent: 'space-between',
  },
  text: {
    color: Colors.white,
    textAlign: 'center',
    fontWeight: 'bold',
    height: 20,
  },
  desc: {
    textAlign: 'center',
  }
});

export default AboutScreen;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

DESARROLLO SCREENS ABOUTSCREEN LOGRADO

React

DESARROLLO (SCREENS TASKSCREEN)

```
import React ,{ useState } from 'react';
import { StyleSheet, View, TextInput, Text, Image } from 'react-native';
import Colors from '../../Config/Colors';
import Images from '../../Config/Images';
import OnboardingFactory from '../../Components/OnboardingFactory';

const TaskScreen = ({ navigation }) => {
  const [isLoading, setIsLoading] = useState(false);
  const _onPressP = () => {
    navigation.navigate('DefensaHito3');
  };
  const _onPressN = () => {
    navigation.navigate('Login');
  };

  return (
    <View style={styles.container}>
      <OnboardingFactory
      source={Images.IMG3}
      style={styles.desc}
      textoTit={'FIREBASE'}
      textoDesc={'Integración de React Native con Firebase' }
        isLoading={isLoading}
        onPressP={_onPressP}
        onPressN={_onPressN}>
      </OnboardingFactory>
    </View>
  );
};
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Se crean los eventos

Se hace un return al componente para que se lo pueda exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (SCREANS TASKSCREEN)

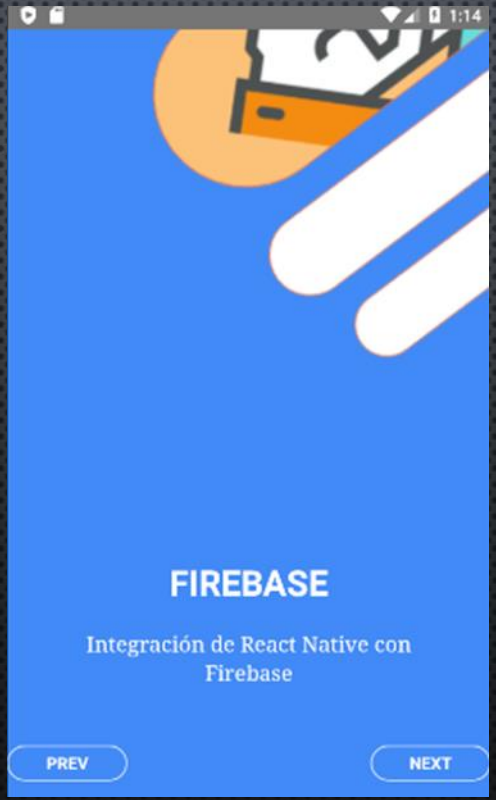
```
const styles = StyleSheet.create({
  container: {
    flex: 3,
    paddingBottom:10,
    backgroundColor: Colors.bluee,
    justifyContent: 'space-between',
},
  text: {
    color: Colors.white,
    textAlign: 'center',
    fontWeight: 'bold',
    height: 20,
  },
  desc:{
    textAlign:'center',
  }
});

export default TaskScreen;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

DESARROLLO SCREENS TASKSCREEN LOGRADO



React

DESARROLLO (SCREENS WELCOMESCREEN)

```
import React ,{ useState } from 'react';
import { StyleSheet, View, TextInput, Text, Image } from 'react-native';
import Colors from '../../Config/Colors';
import Images from '../../Config/Images';
import OnboardingFactory from '../../Components/OnboardingFactory';

const WelcomeScreen = ({ navigation }) => {
  const [isLoading, setIsLoading] = useState(false);
  const _onPressP = () => {
  };
  const _onPressN = () => {
    navigation.navigate('DefensaHito3');
  };

  return (
    <View style={styles.container}>
      <OnboardingFactory
      source={Images.IMG1}
      style={styles.desc}
      textoTit={'PDM'}
      textoDesc={'Programación de dispositivos Móviles - Unifranz' }
        isLoading={isLoading}
        onPressP={_onPressP}
        onPressN={_onPressN}>
      </OnboardingFactory>
    </View>
  );
};
```

Se importan los componentes necesarios de react-native

Se importa los colores para el button

Se declara el constructor junto con las propiedades a exportar

Se crean los eventos

Se hace un return al componente para que se lo pueda exportar

Se llaman y configuran todos los componentes necesarios para crear la descripción

DESARROLLO (SCREANS WELCOMESCREEN)

```
const styles = StyleSheet.create({
  container: {
    flex: 3,
    paddingBottom:10,
    backgroundColor: Colors.green,
    justifyContent: 'space-between',
},
  text: {
    color: Colors.white,
    textAlign: 'center',
    fontWeight: 'bold',
    height: 20,
  },
  desc:{
    textAlign:'center',
  }
});

export default WelcomeScreen;
```

Se crea la constante style de la colección stylesheet

Se crean constantes para asignar estilos

Se exporta para que sea utilizable por todos los componentes o vistas que lo requieran

DESARROLLO SCREENS WELCOMESCREEN LOGRADO

React

DESARROLLO (VERIFICACIÓN LOGIN FUNCIONES)

```
const LoginScreen2 = ({ navigation }) => {
    const [email, setEmail] = useState('');
    const [errorEmail, setErrorEmail] = useState('');
    const [password, setPassword] = useState('');
    const [errorPassword, setErrorPassword] = useState('');
    const [isLoading, setIsLoading] = useState(false);

    const _validateEmailAddress = () => {
        let isValidEmail = Utils.isValidEmail(email);
        isValidEmail ? setErrorEmail('') : setErrorEmail(Constants.STRINGS.EMAIL_ERROR);
        return isValidEmail;
    }
    const _validatePassword = () => {
        let isValidPassword = Utils.isValidField(password);
        isValidPassword ? setErrorPassword('') : setErrorPassword(Constants.STRINGS.PASSWORD
_ERROR);
        return isValidPassword;
    }
    const _onPress = () => {
        let emailData = _validateEmailAddress();
        let passwordData = _validatePassword();

        if (emailData && passwordData) {
            loginApp(email, password);
        } else {
            Alert.alert(Constants.STRINGS.EMPTY_TITLE, Constants.STRINGS.EMPTY_VALUES);
        }
    };
```

Declarar las variables

Validar email

Validar contraseña

Funciones

DESARROLLO (VERIFICACIÓN LOGIN FUNCIONES)

```
const loginApp = (email, password) => {
        try {
            setIsLoading(true);
            FirebasePlugin.auth()
                .signInWithEmailAndPassword(email, password)
                .then(user => {
                    setIsLoading(false);
                    navigation.navigate('Register');
                })
                .catch(error => {
                    FirebasePlugin.auth()
                        .createUserWithEmailAndPassword(email, password)
                        .then(user => {
                            setIsLoading(false);
                            navigation.navigate('Register');
                        })
                        .catch(error => {
                            setIsLoading(false);
                            Alert.alert('Invalid Values', error.message);
                        });
                });
        } catch (error) {
            setIsLoading(true);
            Alert.alert('Firebase Error', error.message);
        }
    };
```

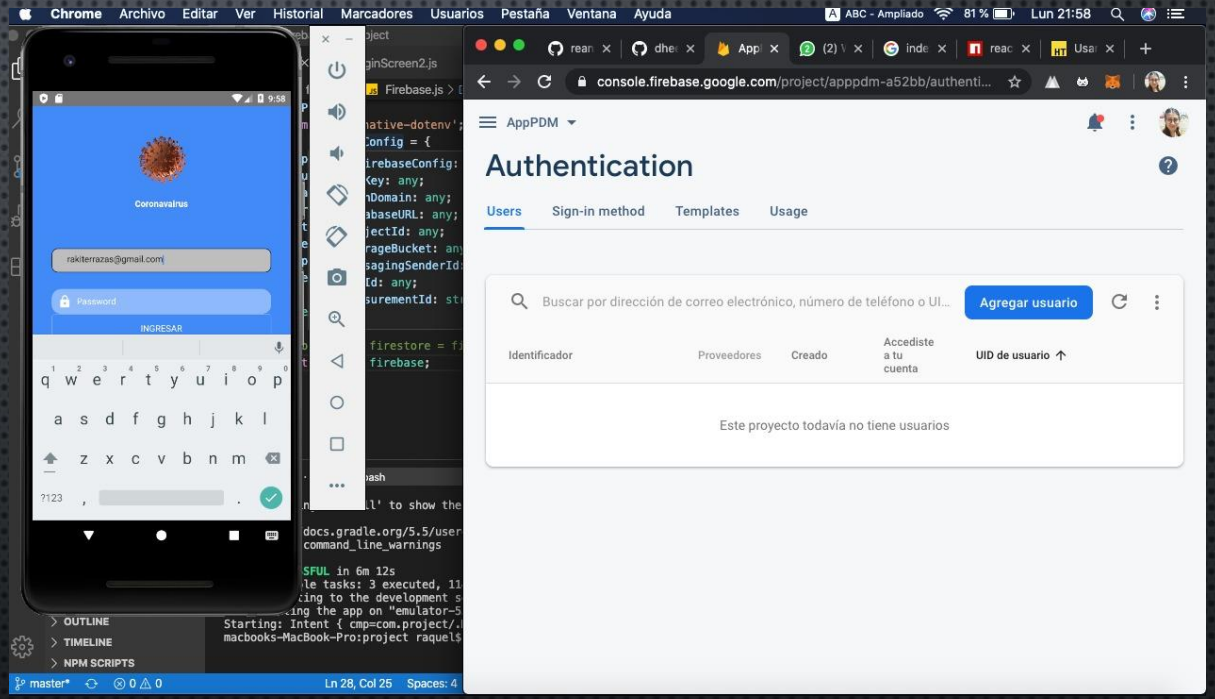
Ingresar con email

Crear usuario con email y password

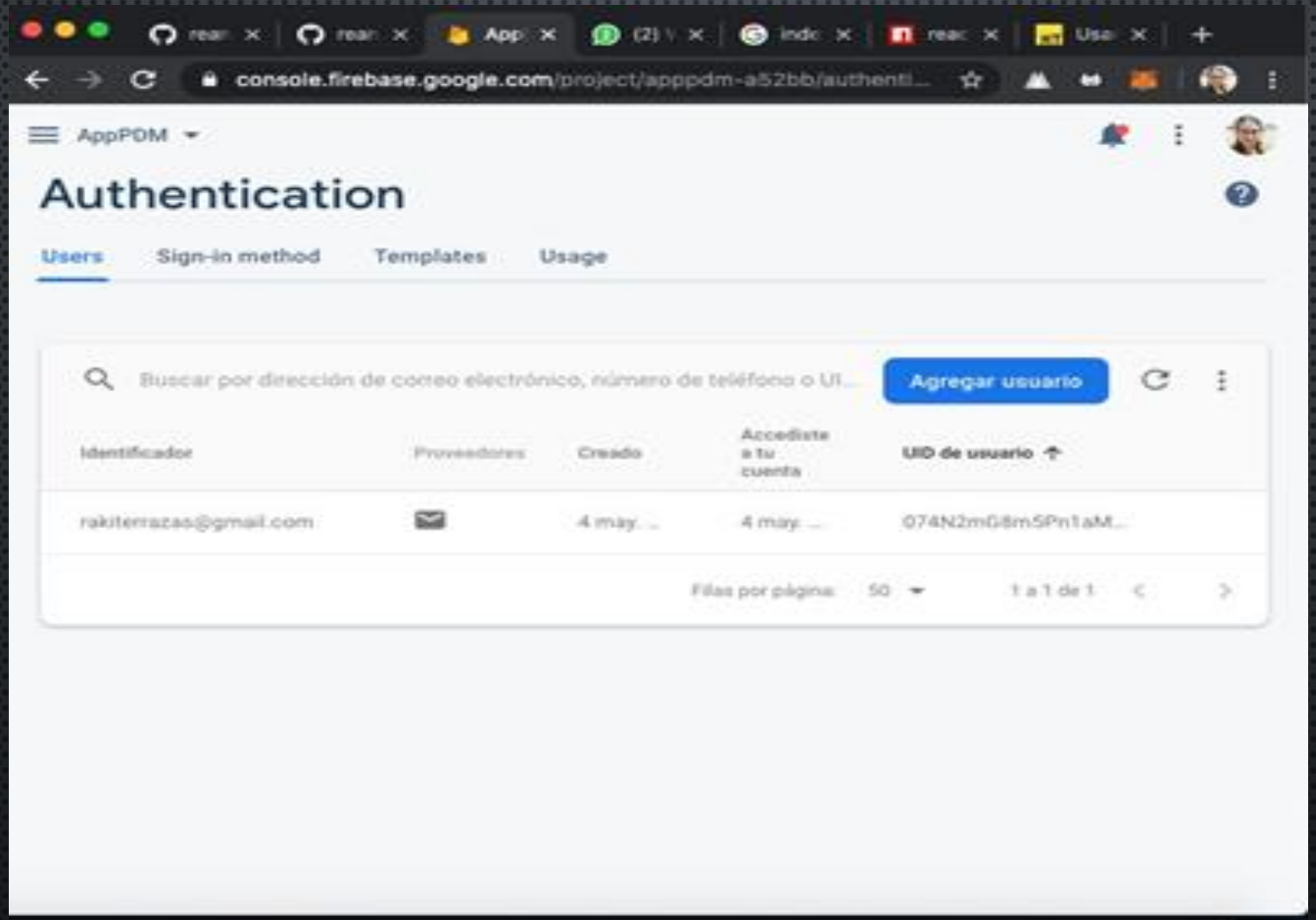
Funcion login

DESARROLLO (LOGIN FIREBASE LOGRADO)

Datos ingresados
Email
rakiterrazas@gmail.com
Contraseña
vcvvxvxvx
Resultado
guardado en firebase



React


console.firebase.google.com/project/appppdm-a52bb/authenti...
AppPDM
Authentication
Users
Sign-in method
Templates
Usage
Buscar por dirección de correo electrónico, número de teléfono o UI...
Agregar usuario
Identificador
Proveedores
Creado
Accediste a tu cuenta
UID de usuario
rakiterrazas@gmail.com
4 may. ...
4 may. ...
074N2mG8m5Pn1aM...
Filas por página: 50
1 a 1 de 1