



ASIGNATURA:

DISEÑO Y PROGRAMACIÓN DE SOFTWARE MULTIPLATAFORMA.

ALUMNO:

MARLENE ELIZABETH FUENTES RAMIREZ.

CARNE:

FJ090135.

CATEDRATICO:

INGENIERO ALEXANDER ALBERTO SIGUENZA CAMPOS.

FORO DE DISCUSIÓN 2.

AUTENTICACIÓN EN REACT NATIVE UTILIZANDO FIREBASE

SAN SALVADOR 24 DE NOVIEMBRE 2024.

INVESTIGACION

Firebase Authentication es una solución integral que ofrece múltiples métodos para autenticar a los usuarios en aplicaciones web y móviles. A continuación, se detallan las opciones disponibles, junto con sus características, ventajas, desventajas y consideraciones clave:

1. Autenticación con correo electrónico y contraseña

- **Características:** Permite a los usuarios registrarse e iniciar sesión utilizando su dirección de correo electrónico y una contraseña. Firebase gestiona el almacenamiento seguro de las credenciales y proporciona funciones para restablecer contraseñas.
- **Ventajas:**
 - Fácil de implementar y comprender para los usuarios.
 - No depende de servicios externos.
- **Desventajas:**
 - Requiere que los usuarios recuerden una contraseña adicional.
 - Puede ser más vulnerable a ataques si no se implementan medidas de seguridad adecuadas.
- **Consideraciones:** Es esencial implementar políticas de contraseñas seguras y ofrecer opciones de recuperación en caso de olvido.

2. Autenticación con proveedores de identidad federada

- **Características:** Permite a los usuarios iniciar sesión utilizando cuentas existentes de proveedores como Google, Facebook, Twitter o GitHub.
- **Ventajas:**
 - Simplifica el proceso de inicio de sesión al utilizar credenciales existentes.
 - Reduce la fricción para el usuario al no requerir la creación de nuevas cuentas.
- **Desventajas:**
 - Depende de la disponibilidad y políticas de los proveedores externos.
 - Puede requerir configuraciones adicionales y cumplimiento de las políticas de cada proveedor.
- **Consideraciones:** Es importante mantenerse actualizado sobre los cambios en las API y políticas de los proveedores para garantizar una integración continua.

3. Autenticación con número de teléfono

- **Características:** Los usuarios se autentican proporcionando su número de teléfono, al cual se envía un código de verificación por SMS.
- **Ventajas:**
 - No requiere que los usuarios recuerden contraseñas.
 - Amplia accesibilidad, ya que la mayoría de las personas poseen un teléfono móvil.
- **Desventajas:**
 - Puede generar costos adicionales debido al envío de SMS.
 - La entrega de SMS puede ser poco fiable en algunas regiones o redes.
- **Consideraciones:** Es recomendable implementar medidas para prevenir abusos, como limitar el número de solicitudes de verificación por usuario.

4. Autenticación anónima

- **Características:** Permite a los usuarios interactuar con la aplicación sin registrarse, creando una cuenta temporal que puede convertirse en permanente posteriormente.
- **Ventajas:**
 - Facilita la adopción inicial de la aplicación sin barreras de registro.
 - Permite a los usuarios probar la aplicación antes de comprometerse con un registro completo.
- **Desventajas:**
 - Las cuentas anónimas pueden perderse si el usuario desinstala la aplicación o cambia de dispositivo sin vincular la cuenta.
 - Limitaciones en la personalización y persistencia de datos a largo plazo.
- **Consideraciones:** Es aconsejable ofrecer opciones claras para que los usuarios conviertan su cuenta anónima en una registrada para preservar sus datos.

5. Autenticación personalizada

- **Características:** Permite integrar sistemas de autenticación existentes o personalizados mediante la generación de tokens personalizados que Firebase puede validar.
- **Ventajas:**
 - Flexibilidad para adaptarse a sistemas de autenticación específicos o requisitos empresariales.
 - Posibilidad de integrar con sistemas heredados o bases de datos de usuarios existentes.
- **Desventajas:**
 - Mayor complejidad en la implementación y mantenimiento.
 - Requiere una gestión cuidadosa de la seguridad y la generación de tokens.
- **Consideraciones:** Es crucial asegurar que los tokens personalizados se generen de manera segura y que se implementen prácticas sólidas de gestión de claves.

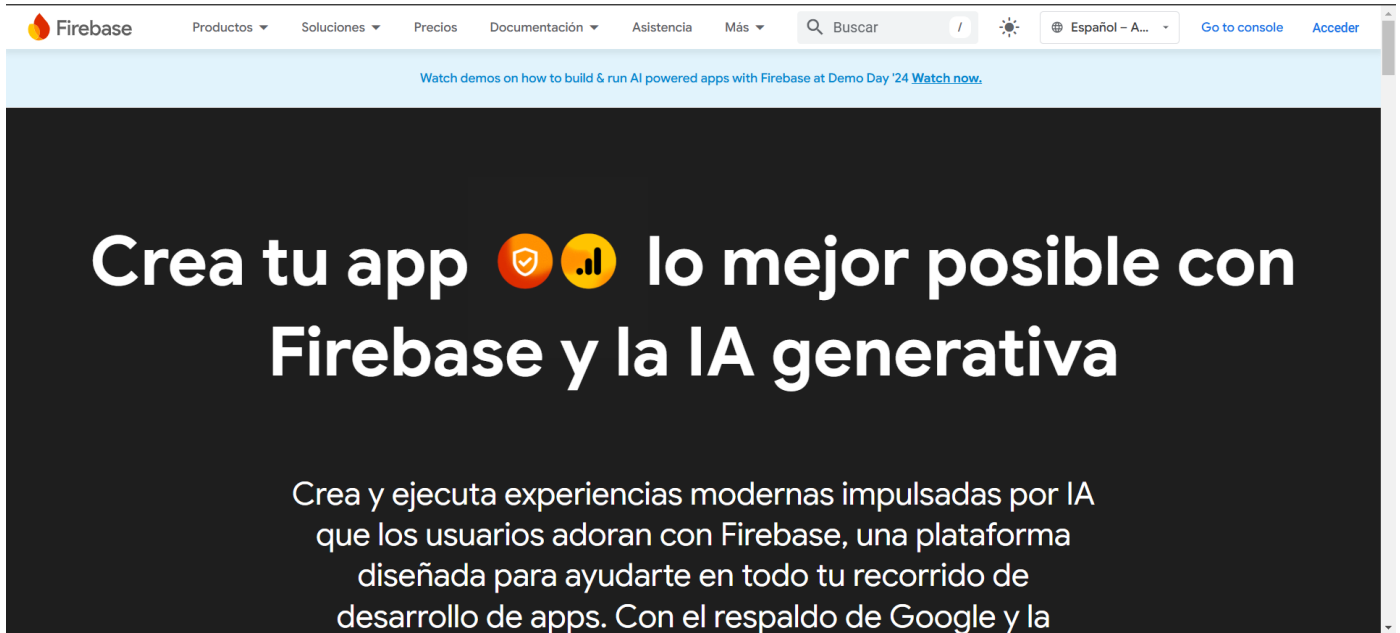
Consideraciones generales al elegir un método de autenticación en Firebase:

- **Seguridad:** Independientemente del método elegido, es fundamental implementar medidas de seguridad robustas, como la verificación en dos pasos y la detección de actividades sospechosas.
- **Experiencia del usuario:** Seleccionar el método que ofrezca la mejor experiencia para el público objetivo, considerando factores como la facilidad de uso y la familiaridad con el método de autenticación.
- **Costos:** Algunos métodos, como la autenticación por SMS, pueden implicar costos adicionales. Es importante evaluar el impacto financiero de cada opción.
- **Regulaciones y cumplimiento:** Asegurarse de que el método de autenticación cumpla con las regulaciones locales e internacionales pertinentes, como GDPR o CCPA.

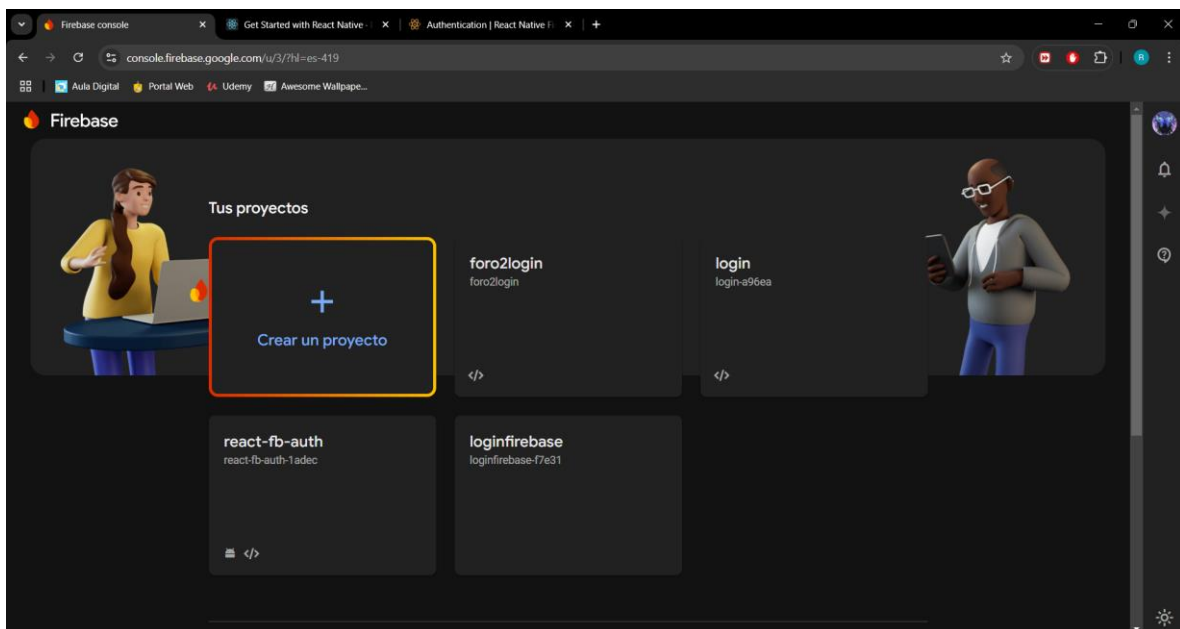
Al evaluar las opciones de autenticación que ofrece Firebase, es esencial considerar las necesidades específicas de la aplicación y de sus usuarios, equilibrando seguridad, facilidad de uso y costos para seleccionar la solución más adecuada.

IMPLEMENTACIÓN

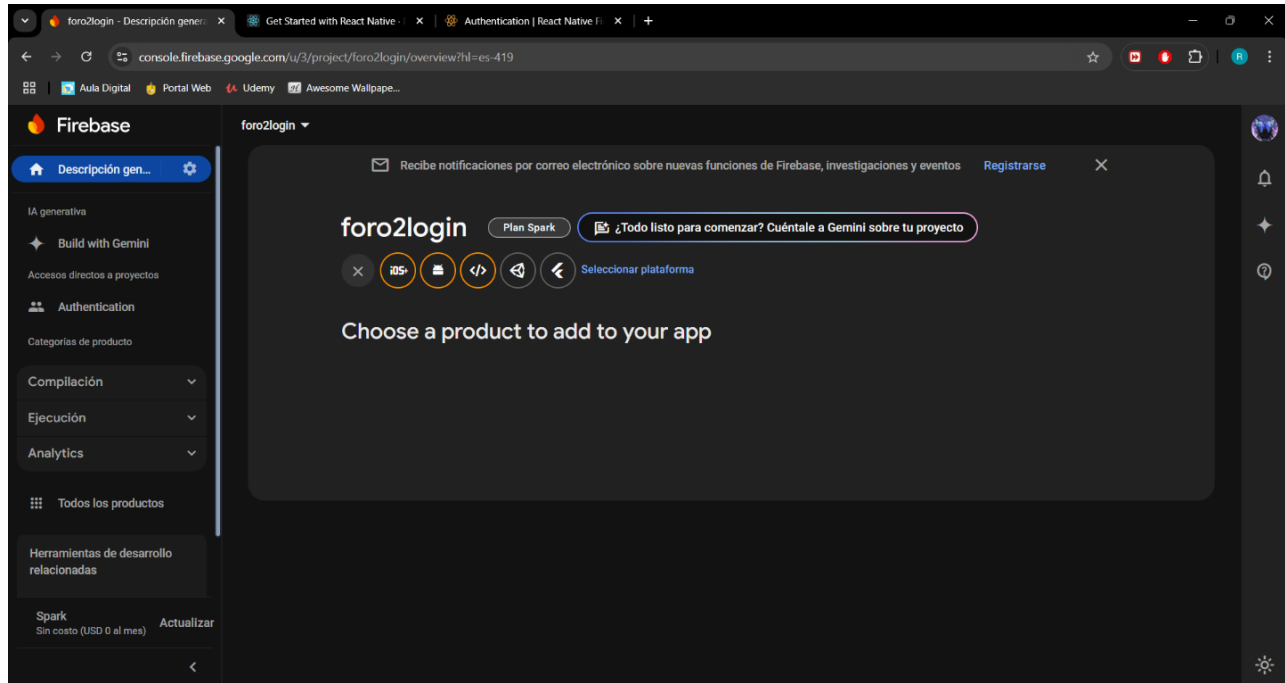
1. Crear un proyecto en la consola de Firebase (<https://console.firebase.google.com/>):



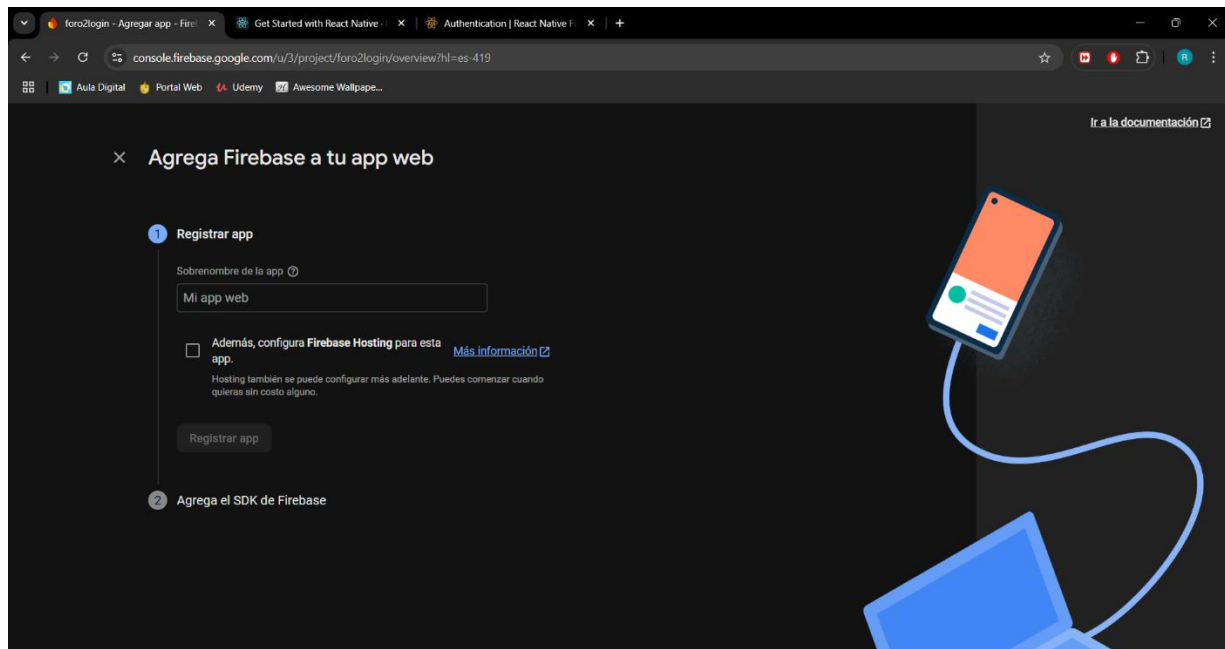
2. En la siguiente sección creamos el proyecto el cual nos servirá para vincular la aplicación.



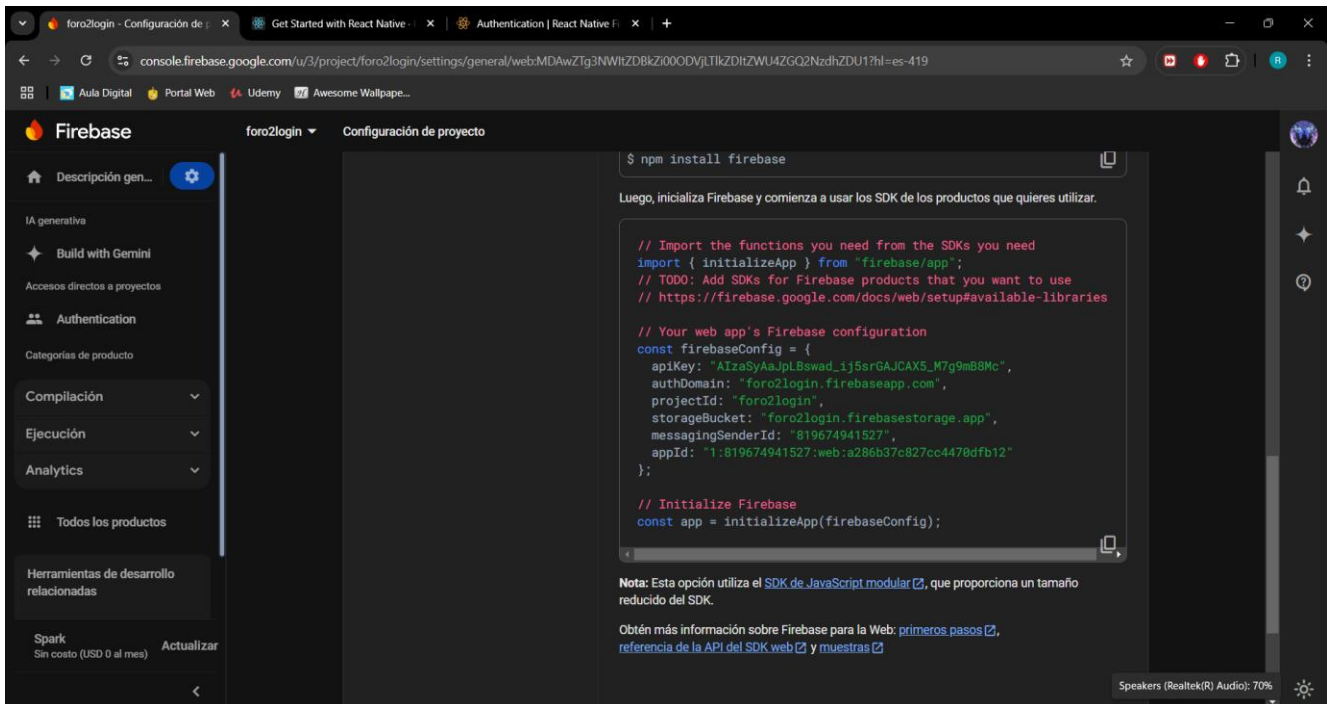
3. Para este trabajo se creó el proyecto denominado foro2login.
4. Tenemos que registrar la app que estará vinculada a firebase estas pueden ser Android, IOS o Web en el caso de utilizar React Native podemos utilizar una Web.



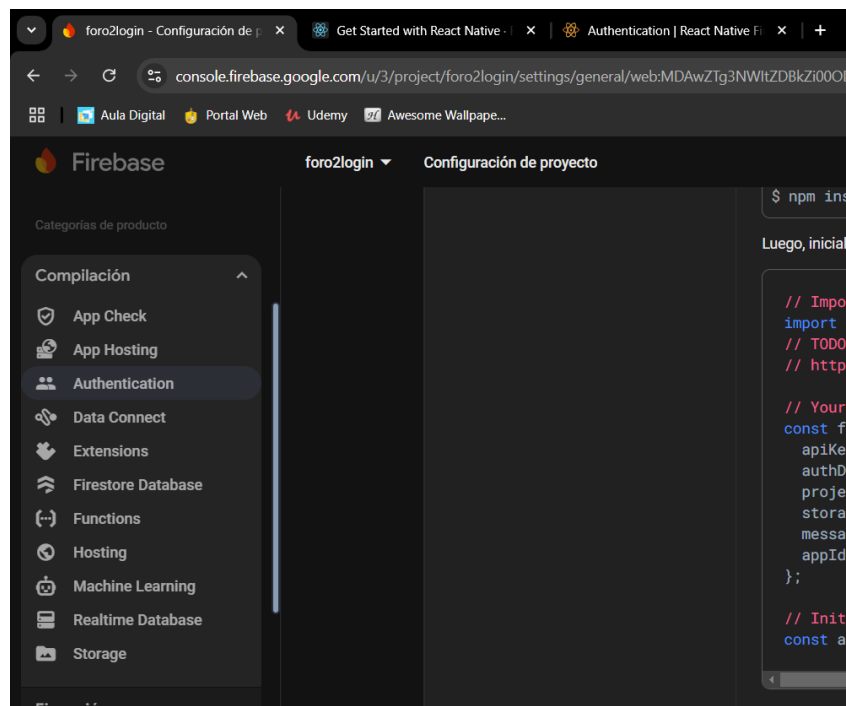
5. Al seleccionar la autenticación web tendremos que colocarle un nombre en este caso le colocamos la misma del nombre del proyecto foro2login.



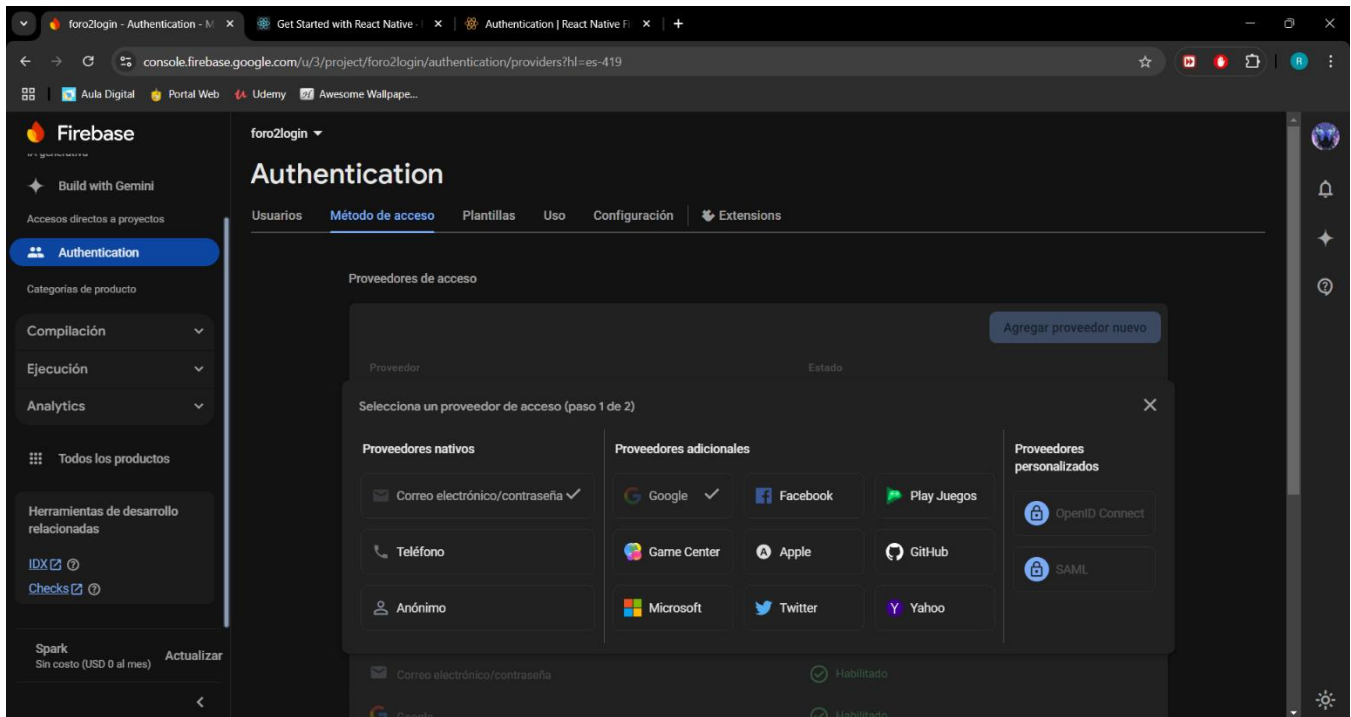
6. Una vez registrada la app nos aparecerán opciones y la configuración que debemos integrar en nuestro código React:



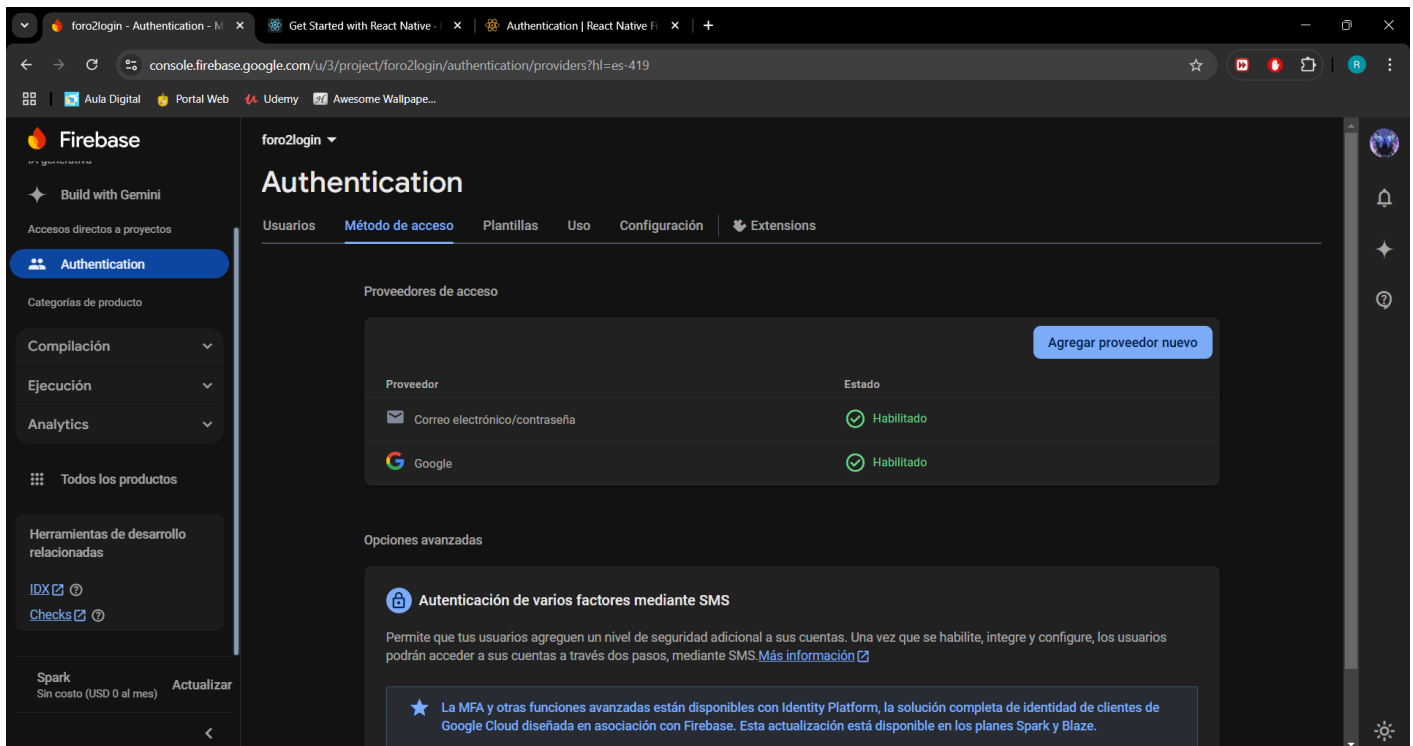
7. A un costado podemos ver una opción llamada Compilación damos click en ella para poder ver la parte de la autenticación



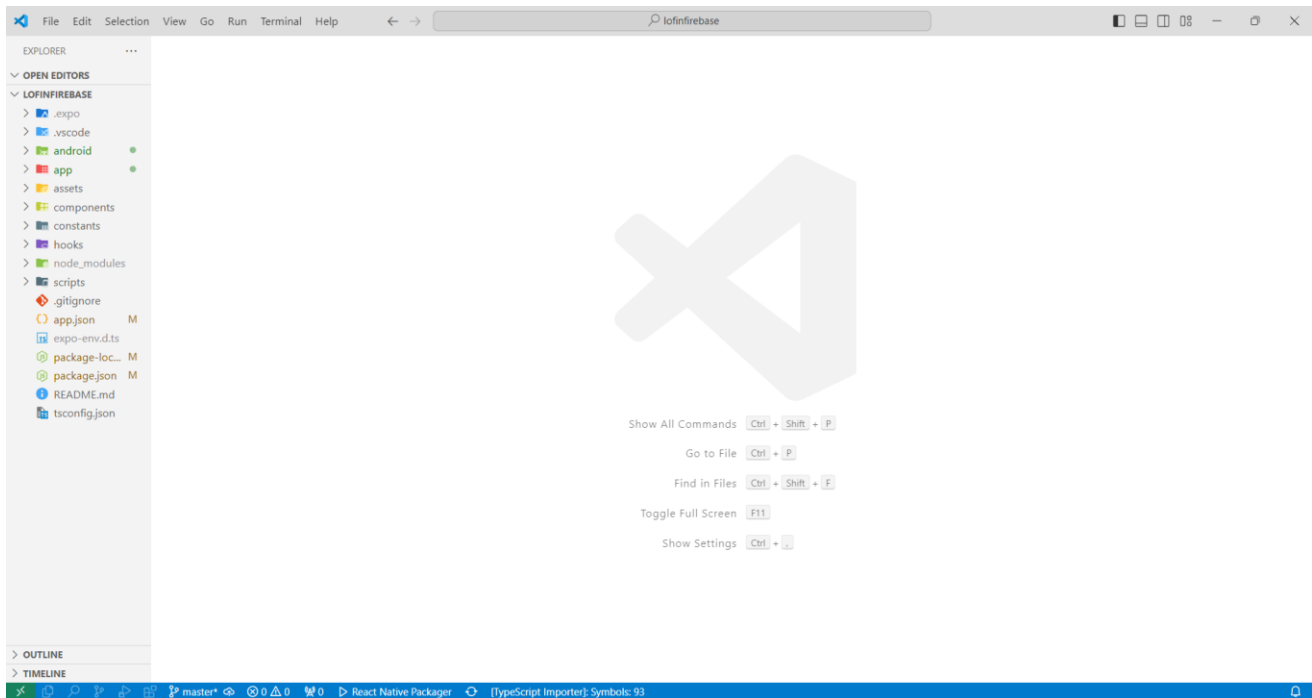
8. En la siguiente vista saldrán los proveedores que estan disponibles para la autenticación con firebase



9. En este caso seleccionaremos correo electrónico y Google para poder realizar el Login y el registro:



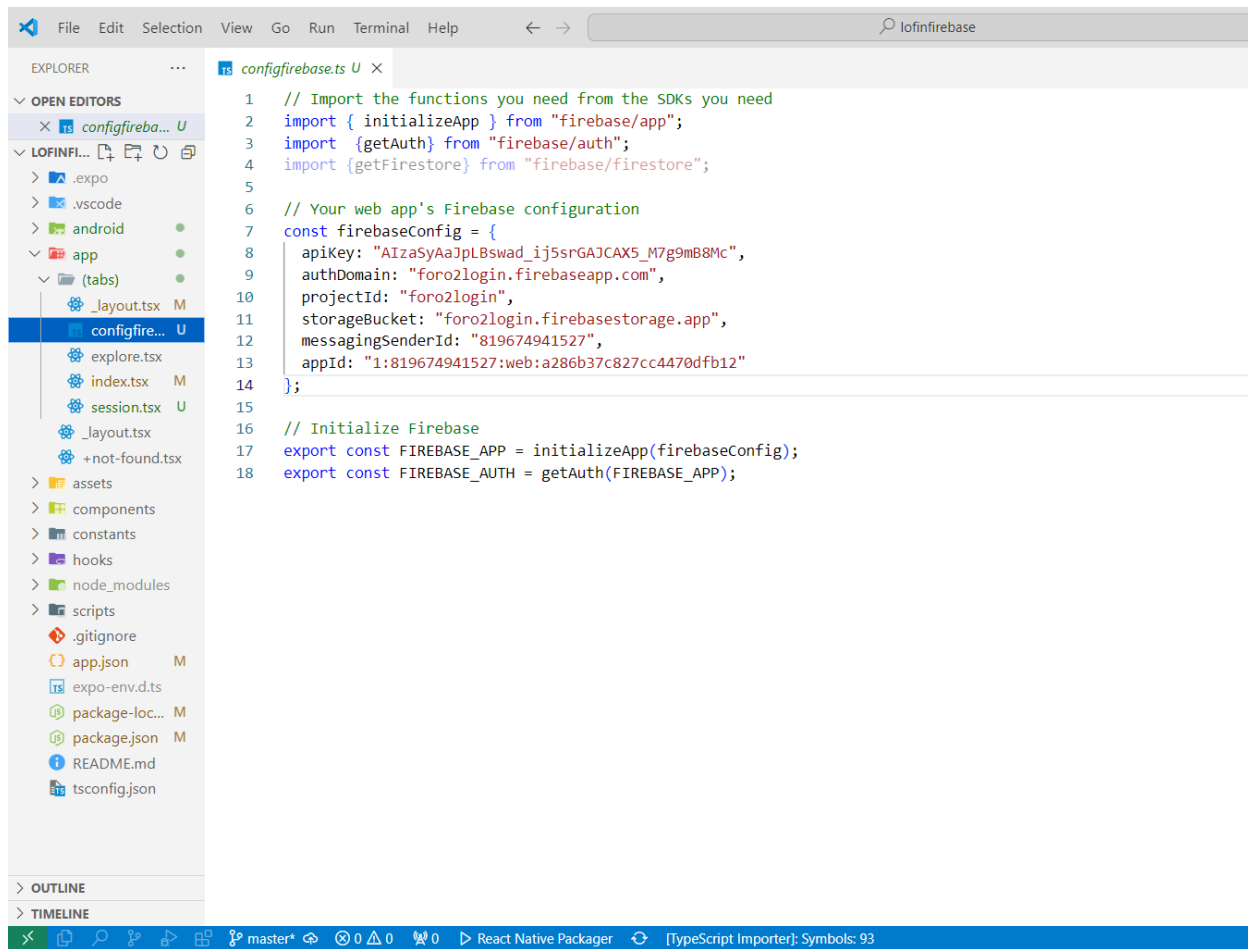
10. Una vez realizado lo anterior procedemos a realizar el proyecto React con Expo.



11. Para poder integrar las funciones que ofrece firebase debemos instalar las siguientes librerías con el comando npm install:

```
npm install firebase/app  
npm install firebase/auth
```

12. Creamos un archivo llamado **configfirebase.ts**, en el cual colocaremos la información del punto 7



13. Una vez incorporado lo anterior procedemos a realizar los campos para el login y el registro de estos

```
import React, { useState } from 'react';
import { ActivityIndicator, Button, TextInput, View, StyleSheet } from 'react-native';
import { FIREBASE_AUTH } from './configfirebase';
import { createUserWithEmailAndPassword, signInWithEmailAndPassword } from 'firebase/auth';
import { useNavigation } from '@react-navigation/native';

export default function Index() {
  const [email, setEmail] = useState(''); //Constante que almacenará los datos del correo
  const [password, setPassword] = useState('');//Constante que almacenará los datos de la contraseña
  const [loading, setLoading] = useState(false); //verificamos el estado si esta o no logeado
  const auth = FIREBASE_AUTH; //llamamos el componente para la autenticacion
  const navigation:any = useNavigation(); //Utilizamos el navigation para moverse entre vistas

  const signIn = async () => {
    if (!email || !password) {
      alert('Por favor, complete todos los campos'); // validamos que los campos no esten vacíos
      return;
    }
    setLoading(true); //seteamos true
    try {
      //realizamos la peticion a firebase junto con el correo y la contraseña
      const response = await signInWithEmailAndPassword(auth, email, password);
      console.log(response);
      //En caso de no tener errores manda un alert y moviliza a la siguiente vista
      alert('Inicio de sesion exitoso');
      navigation.navigate('session');
    } catch (error: any) {
      console.log(error);
      //En caso de error mostrarlo
      alert('Inicio de sesión fallido: ' + error.message);
    } finally {
      setLoading(false);
    }
  };

  //Registrar nuevo usuario
  const signUp = async () => {
    if (!email || !password) {
      alert('Por favor, complete todos los campos');//Validando campos
      return;
    }

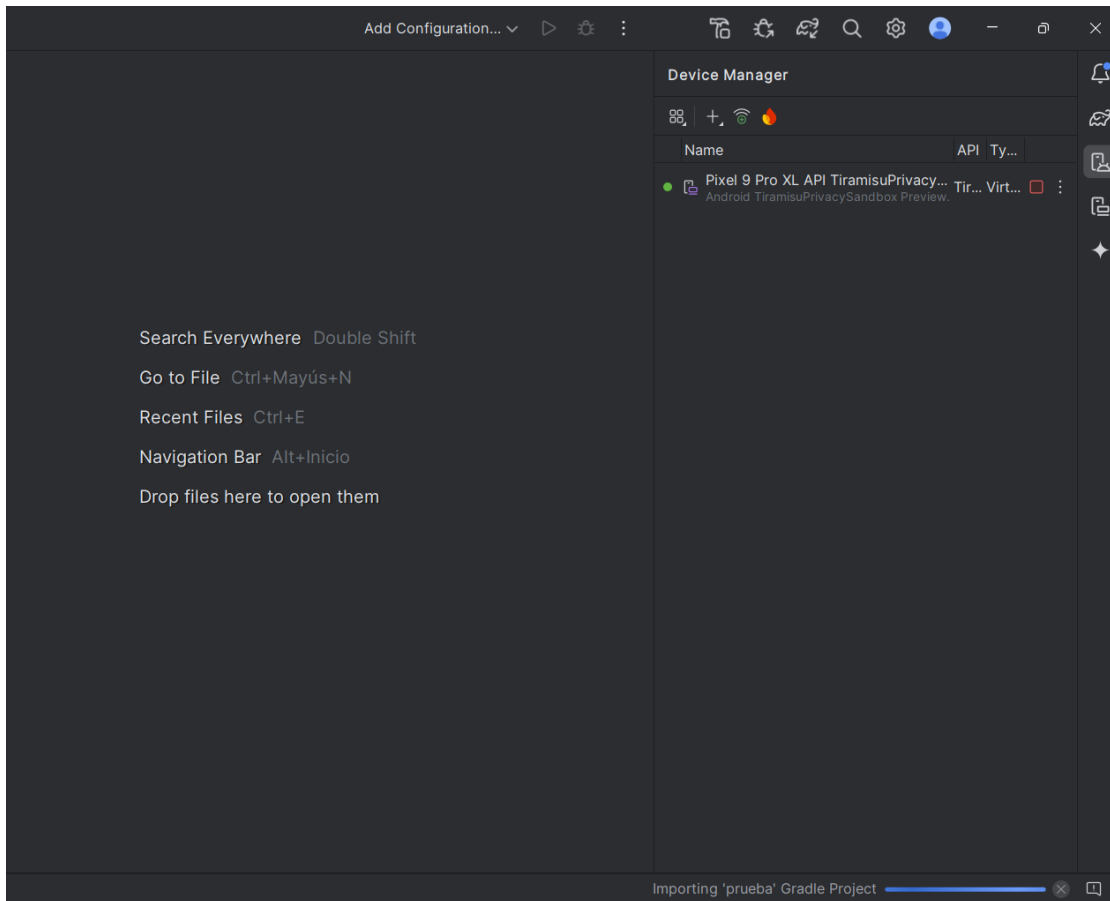
    setLoading(true);
    try {
      //Se manda la peticion para poder ingresar el correo a firebase
      const response = await createUserWithEmailAndPassword(auth, email, password);
      console.log(response);
      //En caso de ser exitoso manda un alert con el siguiente mensaje
      alert('Suscripcion Exitosa revisa tu correo');
    } catch (error: any) {
      //En caso de error mostrarlo
      console.log(error);
      alert('Registro fallido: ' + error.message);
    } finally {
      setLoading(false);
    }
  };
};
```

```

//Se realiza un formulario sencillo con 2 campos y 2 botones
return (
  <View style={styles.container}>
    <TextInput
      value={email}
      style={styles.input}
      placeholder="Email"
      autoCapitalize="none"
      onChangeText={(text) => setEmail(text)}
    />
    <TextInput
      secureTextEntry={true}
      value={password}
      style={styles.input}
      placeholder="Password"
      autoCapitalize="none"
      onChangeText={(text) => setPassword(text)}
    />
    {loading ?
      <ActivityIndicator size="large" color="#0000ff" /> :
      <>
        <Button title="Login" onPress={signIn} />
        <Button title="Crear Usuario" onPress={signUp} />
      </>
    }
  </View>
);
}

```

14. Corremos la aplicación en con el comando **npx expo start**, para este punto es importante que en Android Studio tengamos un dispositivo virtual.



15. La aplicación se ve de la siguiente forma
16. Llenamos los campos y le damos en el botón crear usuario.

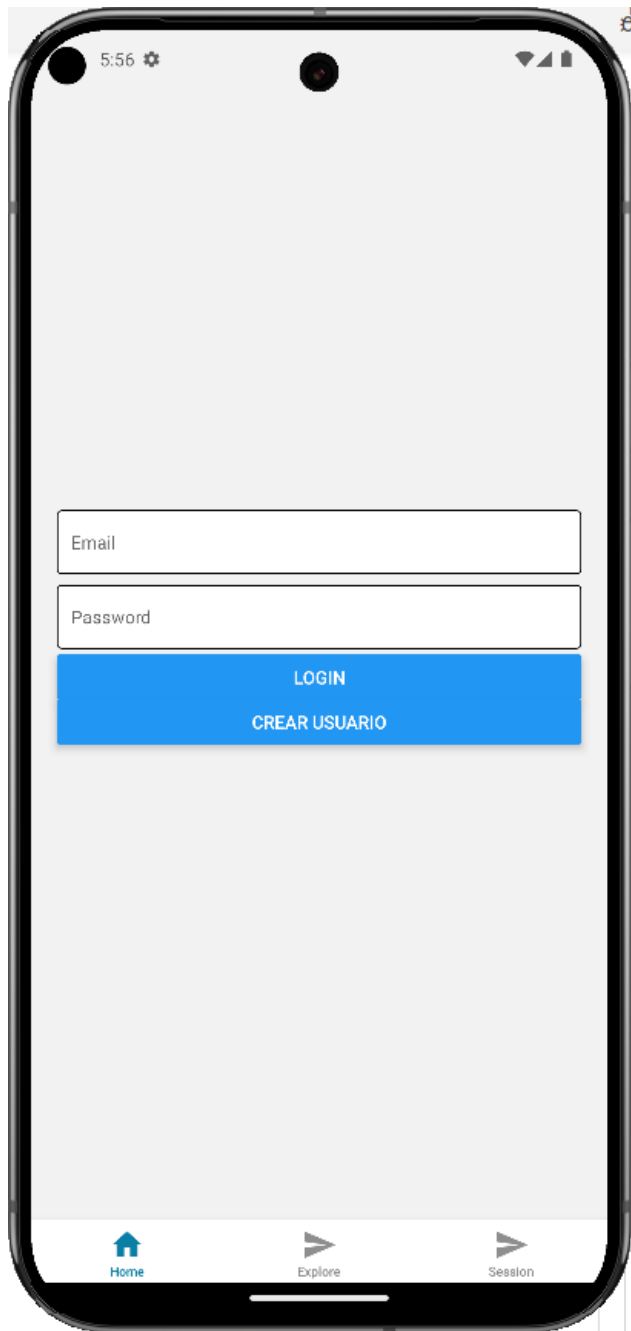


Ilustración 2 Punto 15

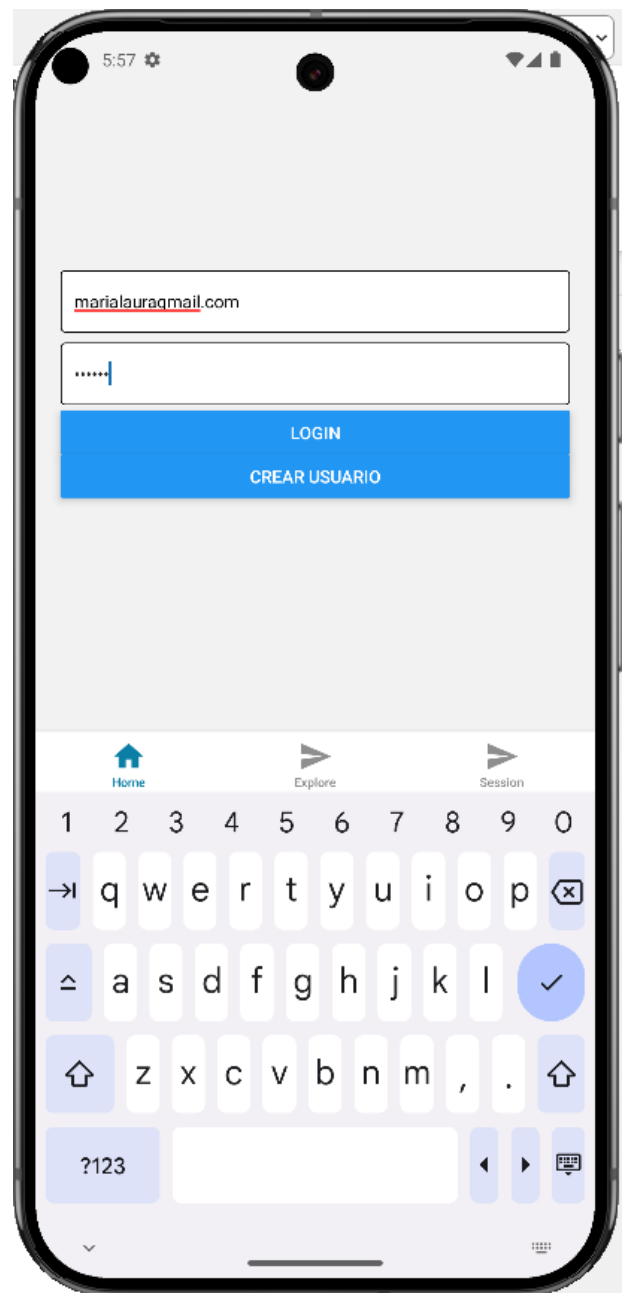
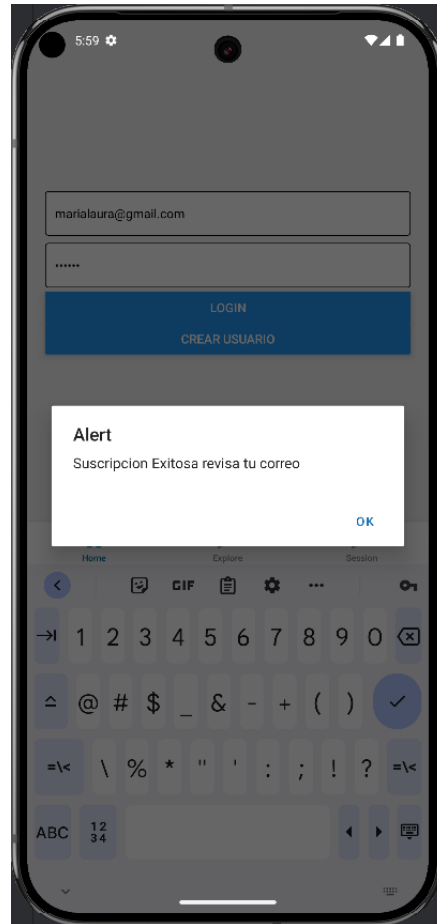


Ilustración 1 Punto 16

17. Si todo se realizó con éxito nos aparecerá el siguiente mensaje



18. Verificamos en Firebase el ingreso del correo

foro2login ▾

Authentication

Usuarios Método de acceso Plantillas Uso Configuración Extensions

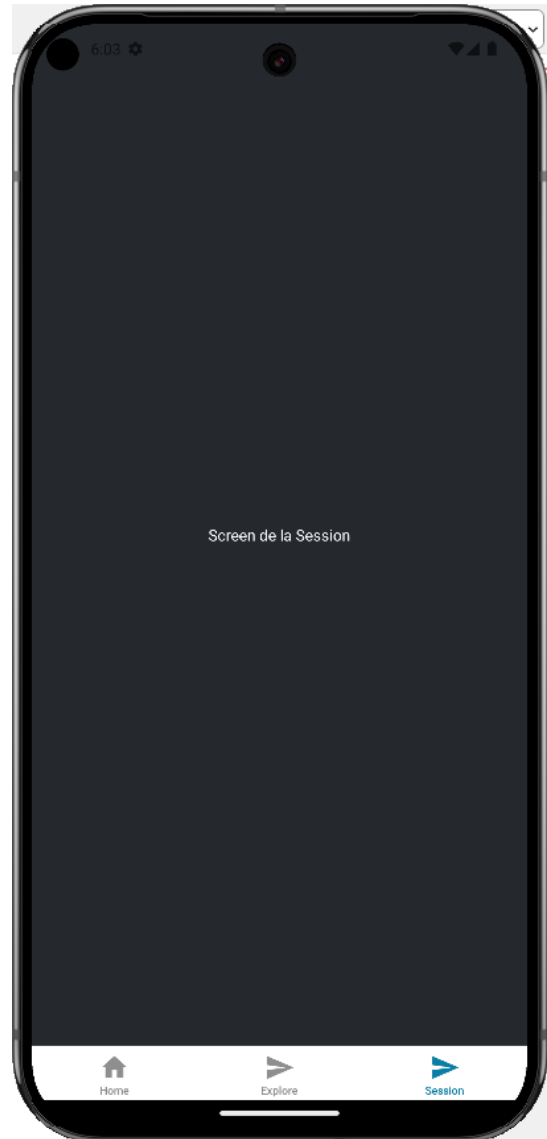
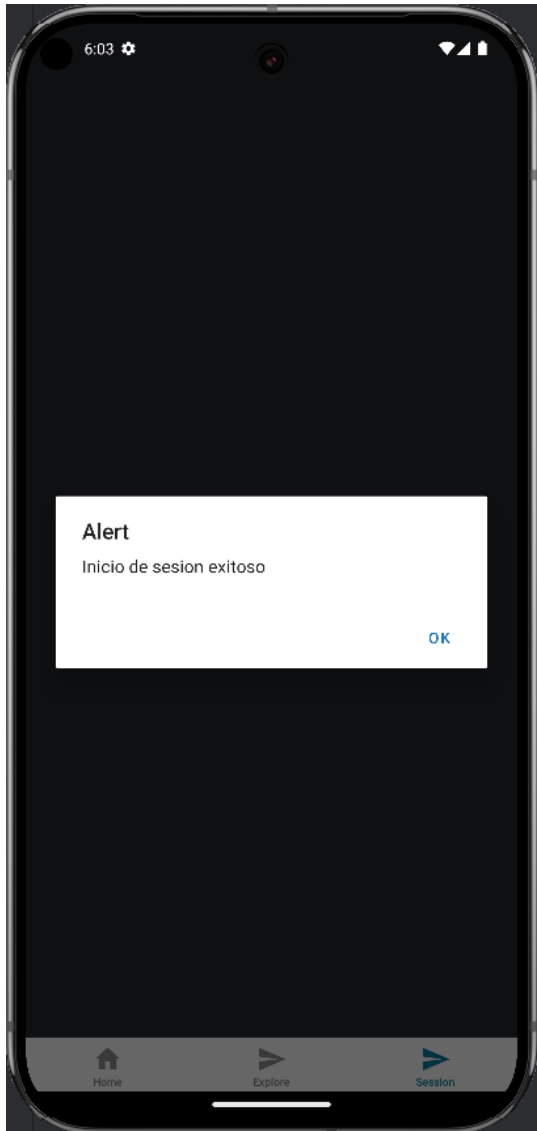
⚠ El acceso con redireccionamiento de origen cruzado ya no es compatible con varios navegadores. Actualiza tu app para asegurarte de que tus usuarios puedan seguir accediendo a ella.

Descartar Más información ↗

🔍 Buscar por dirección de correo electrónico, número de teléfono o UID de usuario Agregar usuario ↻ ⋮

Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario	
marialaura@gmail.com	✉	24 nov 2024	24 nov 2024	S78X7hK5XZNJGcE5BzJheluk...	📄 ⋮

19. Probaremos con el inicio de sesión con el correo ingresado



ANEXO

<https://github.com/Elizabeth914/lofinfirebase>

BIBLIOGRAFIA

Get Started with React Native · React Native. (2024, October 23).

<https://reactnative.dev/docs/environment-setup>

Authentication | React Native Firebase. (n.d.).

<https://rnfirebase.io/auth/usage>