



Trabajo de investigación - Foro de Discusión 2

Instrucciones para la Investigación y Desarrollo de Autenticación en React Native

utilizando Firebase

Diseño y Programación de Software Multiplataforma DPS941 G01T

Docente:

Ing. Alexander Alberto Siguenza Campos

Integrantes:

William Alberto García Gómez - GG212522

Christian Yahir López Hernández - LH212531

Marco Rodrigo Funes Bonilla - FB200456

Edwin Walberto Palacios Mejía - PM140373

Alem Isai Vasquez Antillon - VA223253

Fecha:

24 de noviembre de 2024

ÍNDICE

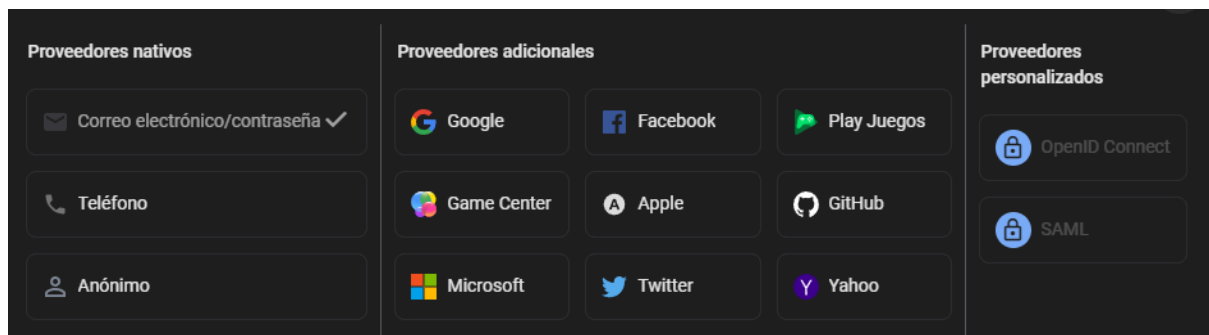
¿Qué es Firebase?	3
Tipos de autenticación	3
- Autenticación de correo electrónico y contraseña	3
- Autenticación redes sociales	4
- Autenticación anónima	4
- Autenticación telefónica	4
Implementación de la autenticación	5
Autenticación por correo y contraseña	5
Implementar lógica	7
Autenticación por Google	8
Vistas de la aplicación	12

¿Qué es Firebase?

Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Se destaca por su enfoque en la integración con aplicaciones de Android, iOS y web, y ofrece funcionalidades que abarcan desde bases de datos en tiempo real, autenticación de usuarios, almacenamiento de archivos, hasta análisis de datos y notificaciones push.

También, cuenta con diversas opciones de autenticación, lo que permite gestionar de manera sencilla el acceso seguro a aplicaciones móviles y web.

Tipos de autenticación

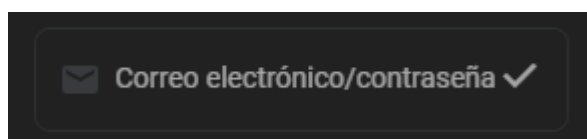


- Autenticación de correo electrónico y contraseña

Permite crear cuentas personalizadas, y verificar el correo electrónico y poder restablecer las contraseñas. Entre sus ventajas se encuentra:

- Familiar para los usuarios.
- Fácil de implementar, y bajo costo.
- Personalización de la experiencia del usuario.

Dentro de las desventajas, se tiene un riesgo y probabilidad de errores de contraseña, y al no tener acceso al correo se podría perder acceso.



- Autenticación redes sociales

Permite un inicio de sesión con cuentas de Google, Facebook, X (Twitter), Play Juegos, Apple, GitHub, Microsoft, Yahoo y Game Center, y garantiza un acceso rápido y seguro. Ventajas de inicio de sesión con redes sociales:

- Reducción del tiempo de registro.
- Uso de cuentas existentes en otras plataformas.

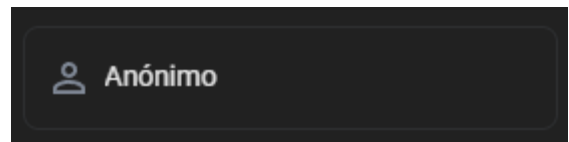
Desventajas de inicio de sesión con redes sociales:

- Dependencia de terceros.
- Posibles cambios en las API de los proveedores.

- Autenticación anónima

Garantiza un acceso a la aplicación sin necesidad de registrarse, y es ideal para pruebas o funcionalidades de invitado. La desventaja de este método de autenticación, es la limitante a ciertas funciones dependiendo la aplicación. Por otro lado, en sus ventajas se tiene:

- Reducción del proceso de registro.
- Posibilidad de visualizar y explorar la aplicación.



- Autenticación telefónica

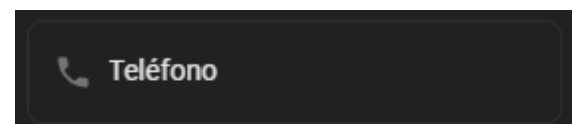
Esta autenticación permite una verificación mediante SMS, y ofrece ingreso a los usuarios sin correo electrónico.

Ventaja:

- Mayor seguridad al evitar ataques de phishing.
- Facilidad de uso en dispositivos móviles.

Desventajas:

- Costos asociados a los mensajes SMS.
- Limitaciones en algunos países o regiones.
- Dependencia de redes de telecomunicaciones.

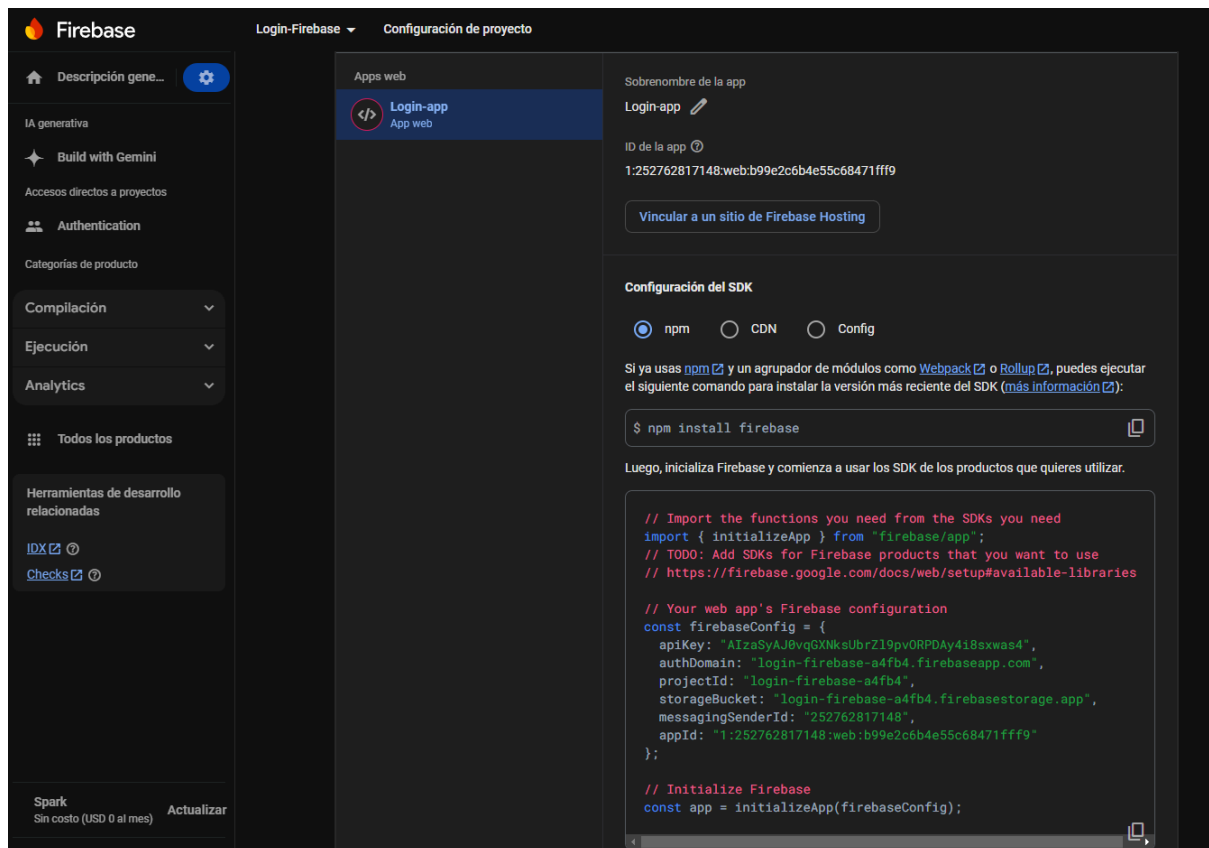


Implementación de la autenticación

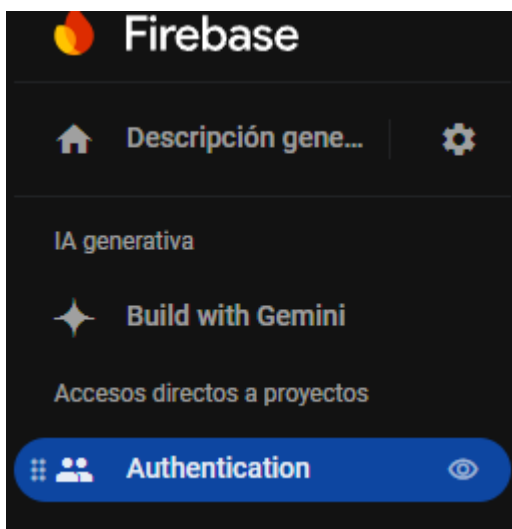
Autenticación por correo y contraseña

Para poder integrar una autenticación por correo electrónico y contraseña en Firebase se puede realizar lo siguiente:

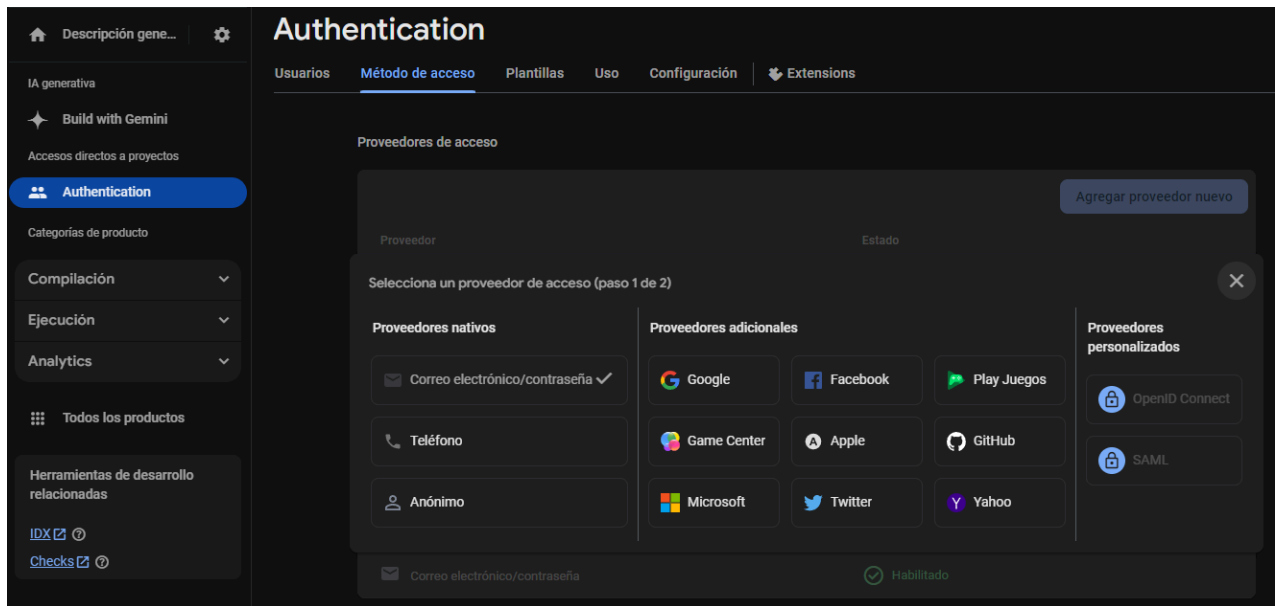
1. Crear un nuevo proyecto en Firebase
2. Configurar el nombre del proyecto
3. Cambiar configuración de proyecto
4. Agregar una App



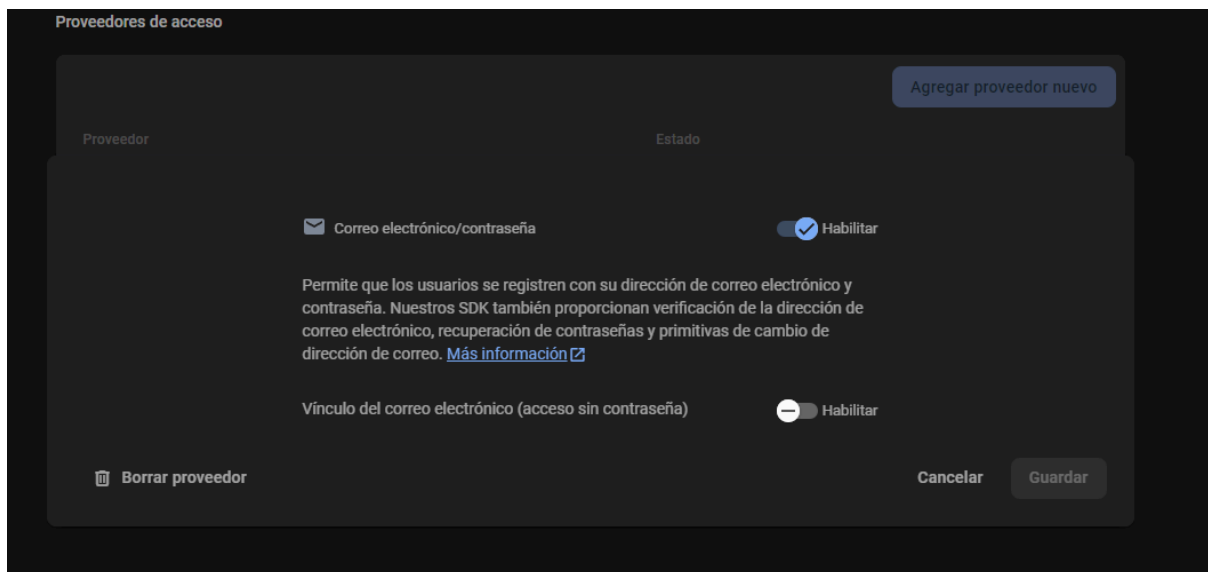
5. Seleccionar Authentication



6. Establecer Correo electrónico/contraseña



7. Guardar cambios



Implementar lógica

- Registro con correo y contraseña

Función signUpWithEmailAndPassword permite que los usuarios creen una cuenta ingresando un correo y una contraseña:

```
const signUpWithEmailAndPassword = async () => {  
  if (!email || !password) {  
    setError('Completa todos los campos para crear tu cuenta');  
    return;  
  }  
  try {  
    await auth().createUserWithEmailAndPassword(email, password);  
    setError('');  
  } catch (error) {  
    console.error(error);  
  }  
};
```

- Inicio de sesión

Función signInWithEmailAndPassword permite a los usuarios existentes iniciar sesión

```
const signInWithEmailAndPassword = async () => {  
  if (!email || !password) {  
    setError('Completa todos los campos antes de iniciar sesión');  
    return;  
  }  
  try {  
    await auth().signInWithEmailAndPassword(email, password);  
    setError('');  
  } catch (error) {  
    console.error(error);  
  }  
};
```

- Cierre de sesión

La función para cerrar sesión se ejecuta al presionar un botón, y desconecta al usuario actual y lo redirige a la pantalla de inicio de sesión.

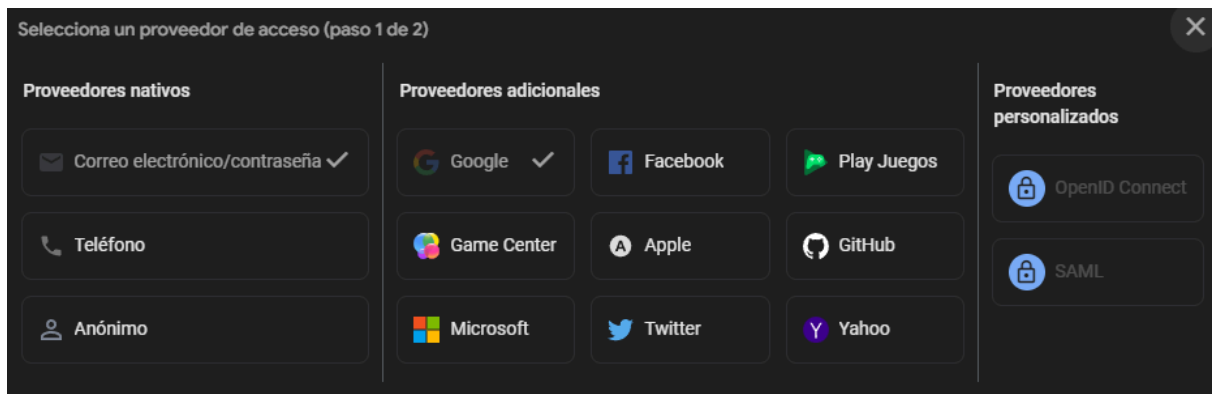
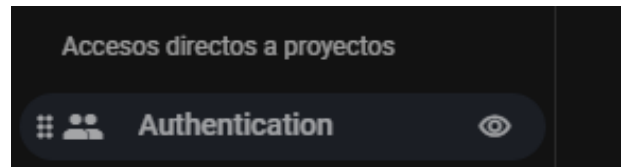
```
<TouchableOpacity onPress={() => auth().signOut()} style={styles.logoutButton}>
<Text style={styles.logoutButtonText}>Cerrar sesión</Text>
</TouchableOpacity>
```

Autenticación por Google

Para poder realizar una autenticación con Google, se deben realizar los siguientes pasos:

Configurar Firebase

1. Seleccionar el proyecto
2. Dirigirse a Authentication
3. Métodos de acceso
4. Establecer Google



- Instalar `@react-native-google-signin` ([Documentación](#))

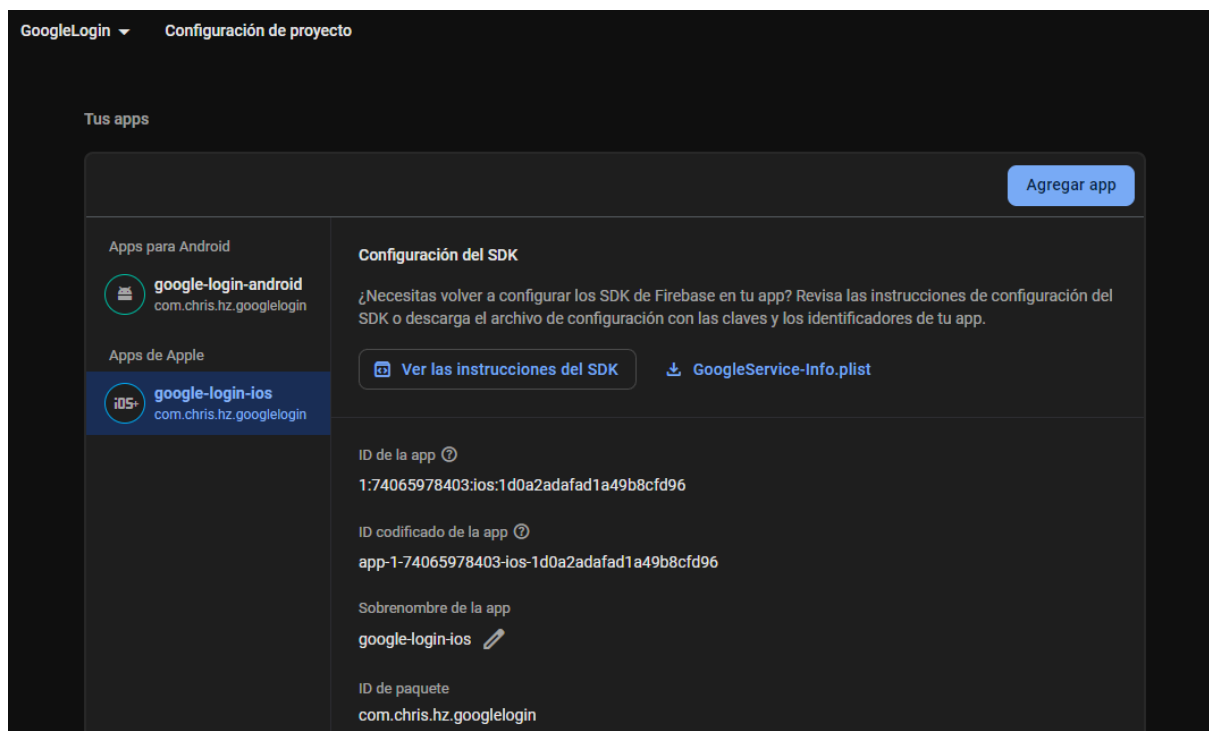
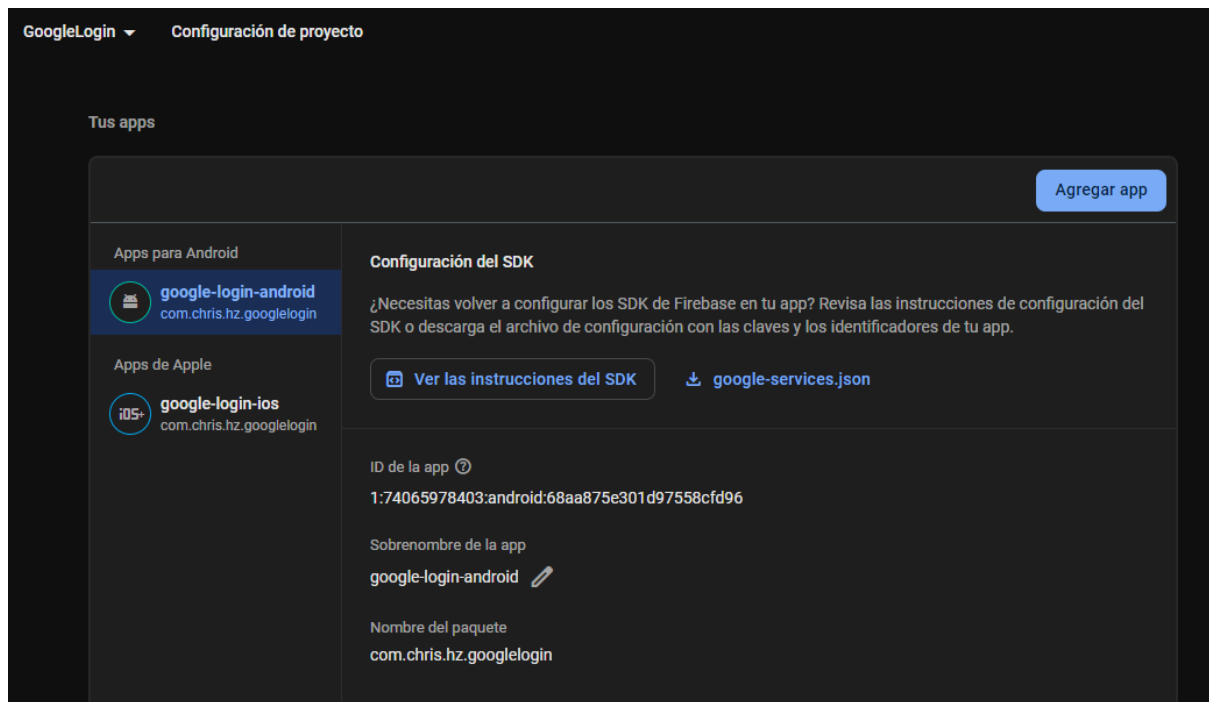
```
npx expo install @react-native-google-signin/google-signin
```

- Configurar archivo `app.json` y agregar plugins:
- Instalar `npx expo install expo-build-properties`

```
"plugins": [
  "expo-router",
  "@react-native-firebase/app",
  [
    "expo-build-properties",
    {
      "ios": {
        "useFrameworks": "static"
      }
    }
  ]
]
```

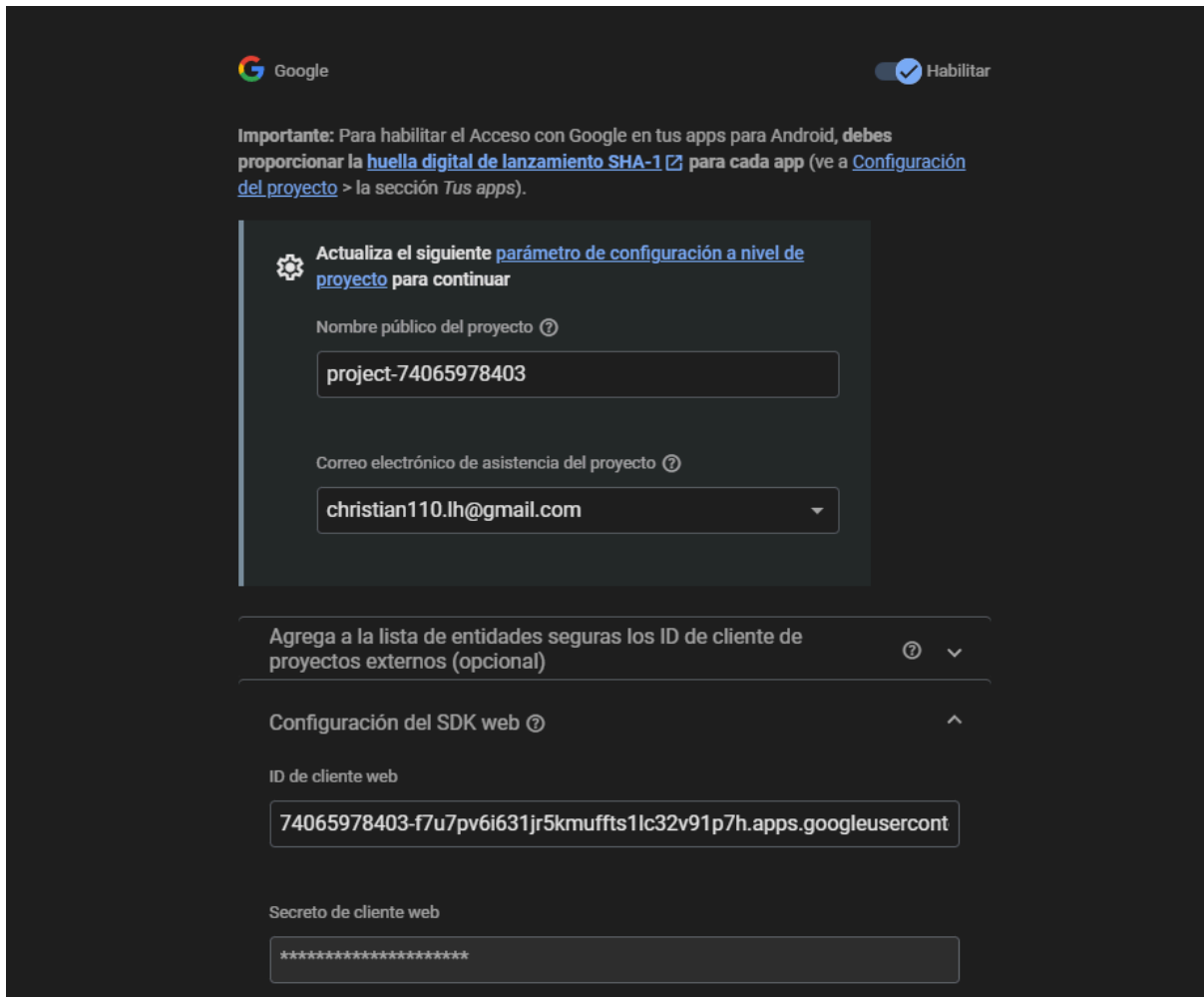
Configurar el cliente de OAuth en Firebase:

- Dirigirse a Configuración del proyecto
- Descargar el archivo google-services.json para Android y GoogleService-Info.plist para iOS.



Implementar lógica de inicio de sesión con Google

1. Copiar ID de cliente web en Authentication



Google Habilitar

Importante: Para habilitar el Acceso con Google en tus apps para Android, debes proporcionar la [huella digital de lanzamiento SHA-1](#) para cada app (ve a [Configuración del proyecto](#) > la sección [Tus apps](#)).

Actualiza el siguiente parámetro de configuración a nivel de proyecto para continuar

Nombre público del proyecto ?
project-74065978403

Correo electrónico de asistencia del proyecto ?
christian110.lh@gmail.com

Agrega a la lista de entidades seguras los ID de cliente de proyectos externos (opcional) ? ▼

Configuración del SDK web ? ^

ID de cliente web
74065978403-f7u7pv6i631jr5kmuffts1lc32v91p7h.apps.googleusercontent.com

Secreto de cliente web

2. Configurar Google Sign-In en el código (index):

```
import { GoogleSignin, isErrorWithCode, statusCodes, GoogleSignInButton }  
from '@react-native-google-signin/google-signin';
```

```
//CONFIGURACION FIREBASE ID  
  
GoogleSignin.configure({  
  webClientId: '74065978403-f7u7pv6i631jr5kmuffts1lc32v91p7h.apps.googleusercontent.com',  
});
```

3. Lógica de inicio de sesión con Google

Cuando el usuario presiona el botón de Google Sign-In, se ejecuta la función `signIn`. Esta función verifica primero la disponibilidad de los servicios de Google en el dispositivo, asegurándose de que el entorno es compatible para iniciar sesión.

Luego, se muestra un selector de cuentas de Google donde el usuario puede elegir una cuenta para autenticarse. Al seleccionar una cuenta, se obtiene un `idToken`, que es un token de autenticación seguro emitido por Google.

```
<GoogleSignInButton
  style={styles.googleButton}
  size={GoogleSignInButton.Size.Icon}
  color={GoogleSignInButton.Color.Dark}
  onPress={signIn}
/>
```

```
//LOGIN GOOGLE (DOCUMENTACION)
const signIn = async () =>{

  try{
    await GoogleSignIn.hasPlayServices({showPlayServicesUpdateDialog: true});
    const response = await GoogleSignIn.signIn();
    console.log('respuesta', response);

    const googleCredential = auth.GoogleAuthProvider.credential(response.data?.idToken);

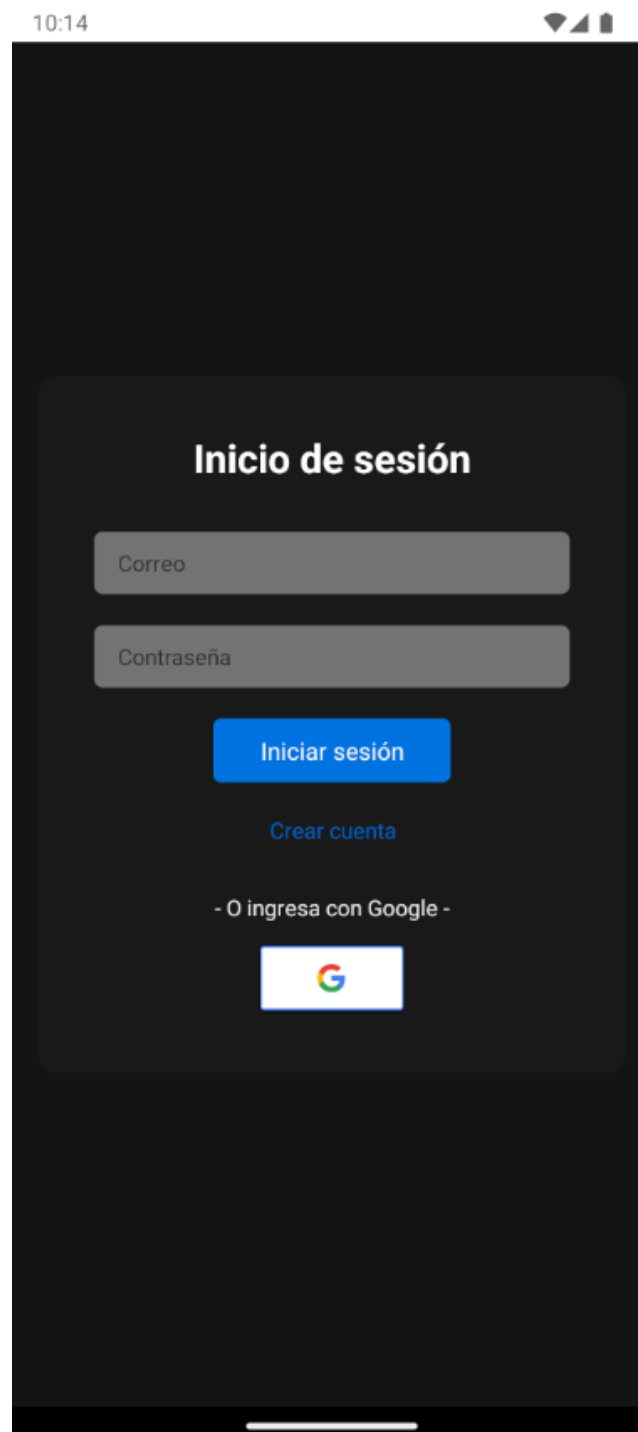
    return auth().signInWithCredential(googleCredential);
  }catch(error){
    if (isErrorWithCode(error)){
      switch(error.code){
        case statusCodes.IN_PROGRESS:
          break;
        case statusCodes.PLAY_SERVICES_NOT_AVAILABLE:
          break;
        default:
      }
    }else{
    }
  }
}
```

Este token se utiliza para generar una credencial de Google mediante el método `auth.GoogleAuthProvider.credential`. Finalmente, esta credencial se pasa a `auth().signInWithCredential()`, que autentica al usuario en Firebase. Si es la primera vez que el usuario inicia sesión, Firebase crea automáticamente una cuenta. Una vez autenticado, se puede acceder a la información básica del usuario, como su nombre, correo electrónico y foto de perfil, desde el objeto `user`.

Vistas de la aplicación

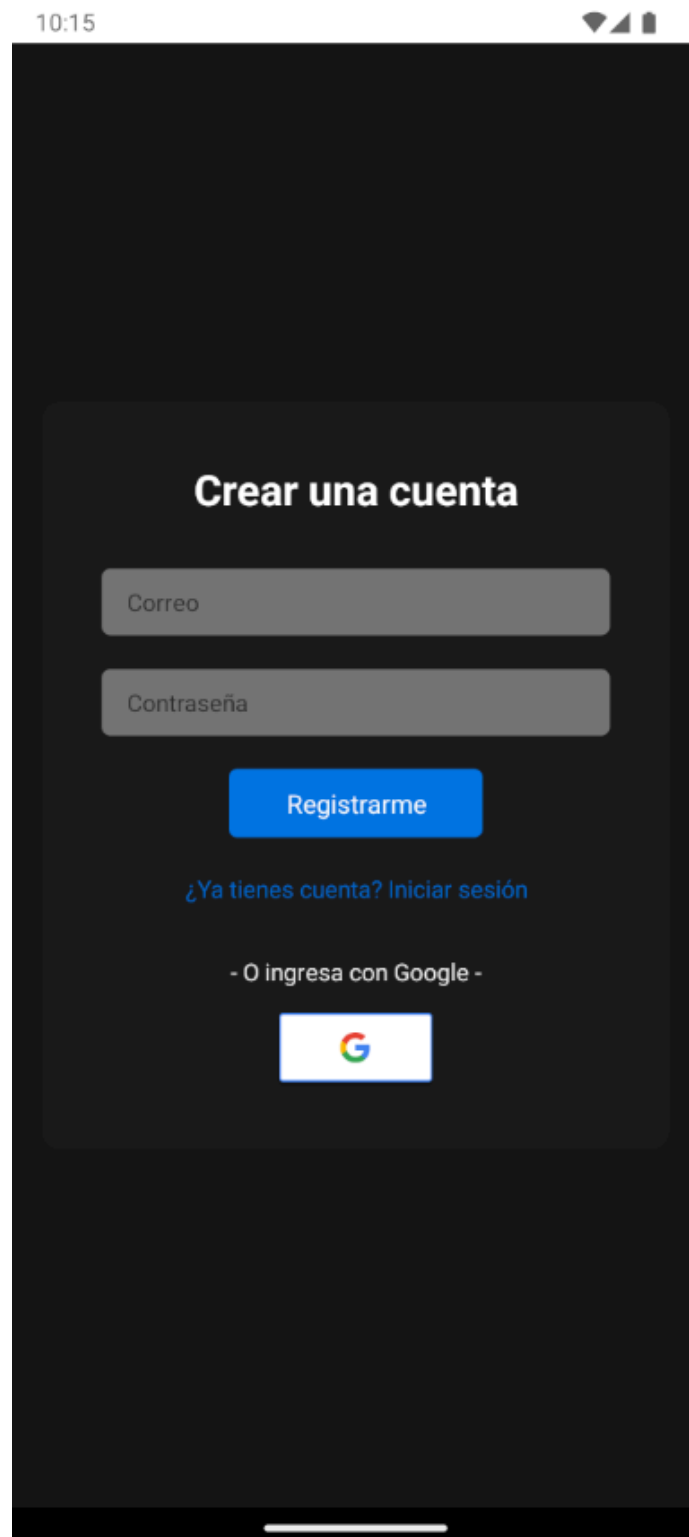
- Inicio de sesión

En esta pantalla se muestran dos input text para ingresar el correo y contraseña, y el respectivo botón de inicio de sesión. También, se tiene un botón para poder ingresar al apartado de Registro de cuenta, y un botón para iniciar sesión por medio de Google.



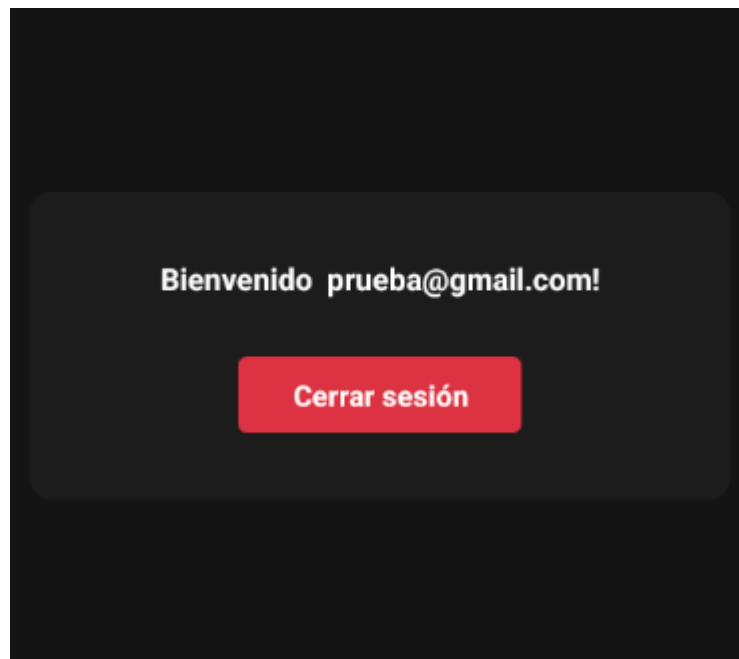
- Registro

Pantalla donde los usuarios se pueden registrar a la aplicación, y crear su cuentas mediante correo y contraseña, o mediante la opción de Google. Una vez creado, se redirige a la ventana Home (ver siguiente apartado).

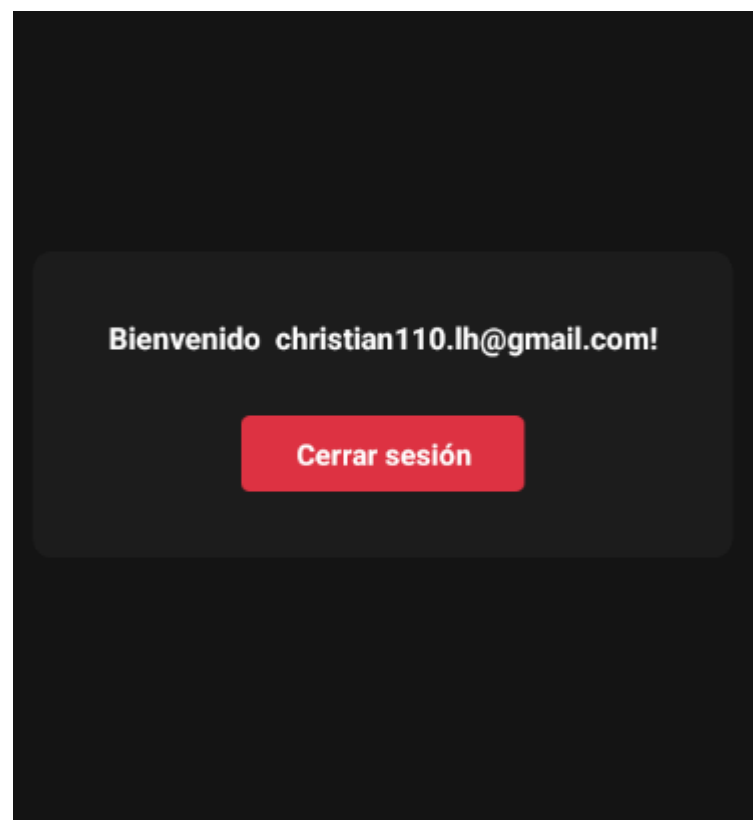
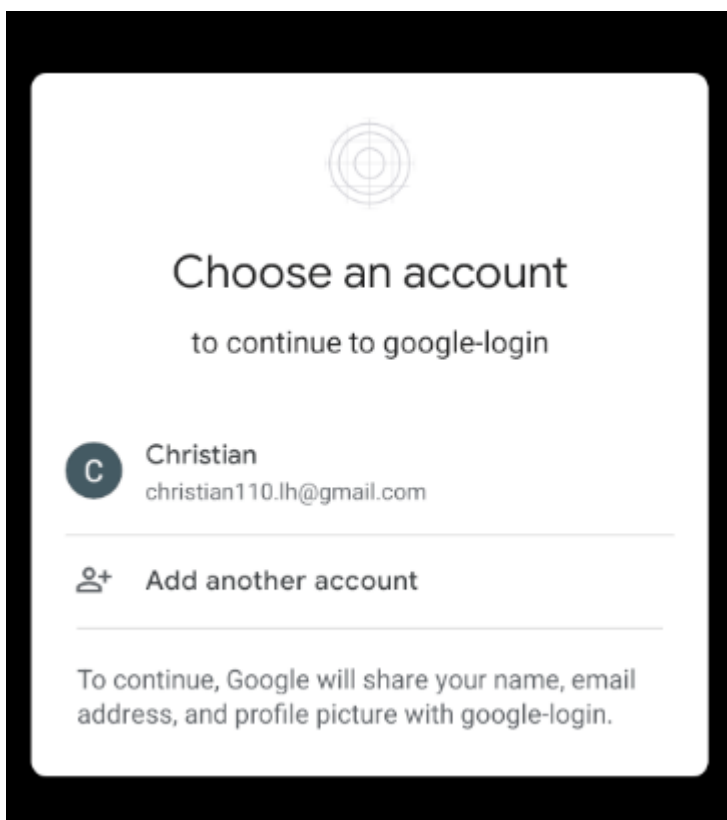


The screenshot shows a mobile application interface for user registration. At the top, the status bar displays the time 10:15 and signal/battery icons. The main content area has a dark background with a central light gray box containing the registration form. The form is titled 'Crear una cuenta' in bold white text. It includes two input fields: 'Correo' and 'Contraseña', both with light gray backgrounds and rounded corners. Below these fields is a blue button with the text 'Registrarme' in white. Underneath the button is a link in blue text that reads '¿Ya tienes cuenta? Iniciar sesión'. Further down is the text '- O ingresa con Google -' in white, followed by a white rectangular button featuring the Google 'G' logo. The bottom of the screen shows a white horizontal line, likely representing the home indicator on an iPhone.

- Inicio de sesión mediante correo y contraseña

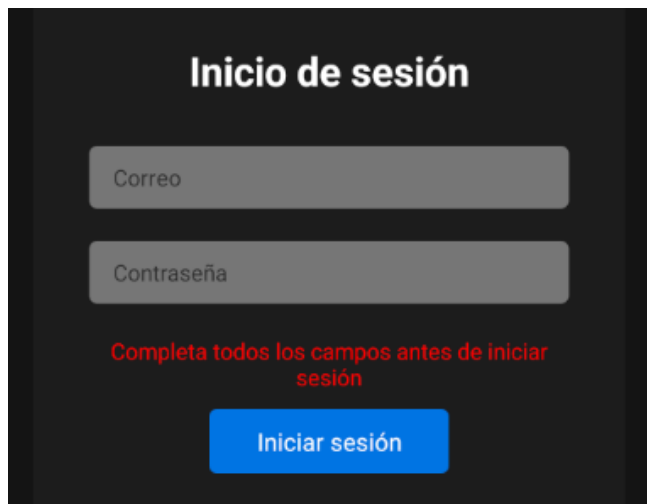


- Inicio de sesión mediante Google



- Manejo de errores

Si el usuario no ingresa ambos campos de texto, se muestra un mensaje que deben ingresar correo y contraseña.



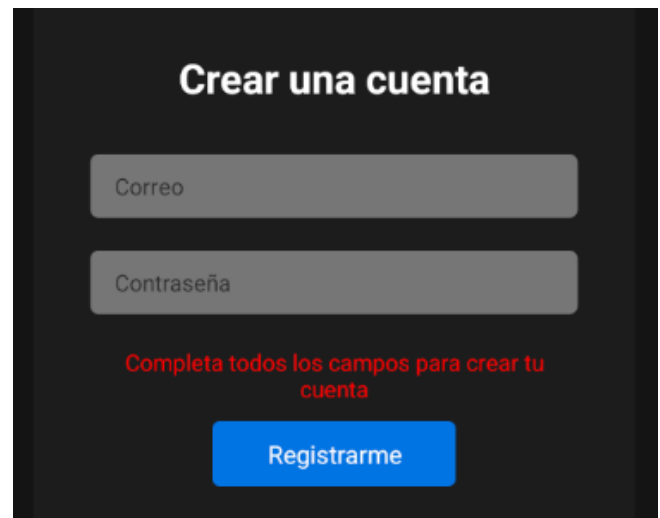
Inicio de sesión

Correo

Contraseña

Completa todos los campos antes de iniciar sesión

Iniciar sesión



Crear una cuenta

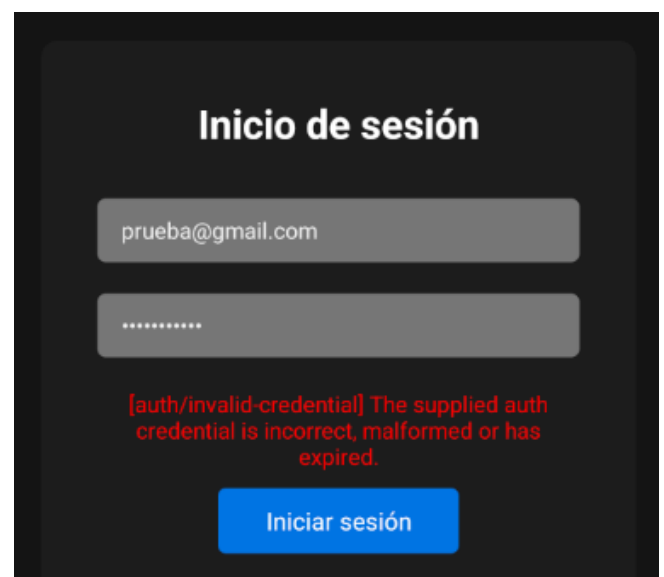
Correo

Contraseña

Completa todos los campos para crear tu cuenta

Registrarme

Cuando las credenciales son incorrectas, se muestra un mensaje de error de Firebase

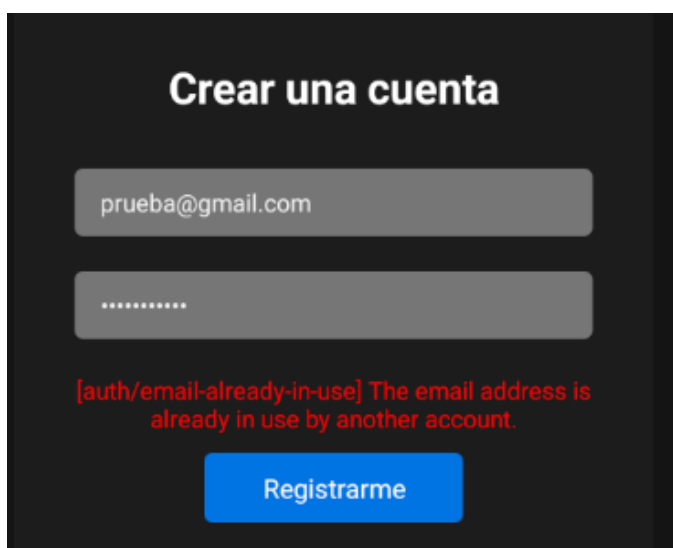


Inicio de sesión

prueba@gmail.com

[auth/invalid-credential] The supplied auth credential is incorrect, malformed or has expired.

Iniciar sesión



Crear una cuenta

prueba@gmail.com

[auth/email-already-in-use] The email address is already in use by another account.

Registrarme

Error cuando hay una cuenta existente con el correo electrónico