



UNIVERSIDAD DON BOSCO

Dirección de Educación a Distancia

Ingeniería en Ciencias de la Computación

Diseño y Programación de Software Multiplataforma (DPS941)

Foro II

Presentado por

Ricardo Arturo De Paz Núñez (DN192246)

Kevin Alexander Fernandez Monge (FM150385)

Julio Danilo Flores Fuentes (FF201999)

Christian Alexander Hernández Funes (HF171856)

Kelvin Vladimir García Juárez (GJ111587)

Fecha de Entrega

3 de diciembre de 2023

Contenido

INTRODUCCION.....	3
DESARROLLO	4
1. INVESTIGACIÓN DE OPCIONES DE AUTENTICACIÓN CON FIREBASE EN REACT NATIVE	4
Opciones de autenticación de Firebase con Gmail.....	5
2. IMPLEMENTACIÓN DE AUTENTICACIÓN CON FIREBASE UTILIZANDO GOOGLE CLOUD	8
Demostración De Implementación De Autenticación Por Correo De Gmail.	19

INTRODUCCION

La autenticación en aplicaciones móviles se ha convertido en un componente esencial para garantizar la seguridad y la experiencia del usuario. En este contexto, React Native ha emergido como una tecnología prominente para el desarrollo de aplicaciones móviles multiplataforma, mientras que Firebase proporciona una amplia gama de servicios, incluida la autenticación, que simplifica el proceso de desarrollo.

En el marco de este trabajo, se llevó a cabo una exhaustiva investigación de las opciones disponibles para la autenticación en React Native utilizando Firebase. La finalidad fue explorar las funcionalidades y posibilidades que esta combinación de tecnologías ofrece, centrándonos particularmente en la autenticación a través del correo electrónico asociado a Gmail, una característica clave de Firebase.

DESARROLLO

1. INVESTIGACIÓN DE OPCIONES DE AUTENTICACIÓN CON FIREBASE EN REACT NATIVE

Las opciones de autenticación que Firebase proporciona para aplicaciones React Native. Firebase Authentication ofrece diversos métodos, como correo electrónico, Google, Facebook, etc.

Casi todas las aplicaciones requieren alguna forma de verificar quiénes son sus usuarios. Entender quién está utilizando la app es clave para almacenar sus datos de forma segura en la nube y proporcionarles una experiencia personalizada y consistente en todos sus dispositivos.

Firebase Authentication ofrece servicios de backend, SDKs sencillos de integrar y bibliotecas de interfaz de usuario ya listas para autenticar a los usuarios en mi aplicación. Puedo autenticar a los usuarios usando contraseñas, números de teléfono y también aprovechar proveedores populares de identidad federada, como Google, Facebook y Twitter, entre otros.

La integración de Firebase Authentication es muy sencilla y se conecta perfectamente con otros servicios de Firebase. Utiliza estándares de la industria como OAuth 2.0 y OpenID Connect, lo que facilita su incorporación a mi propio backend personalizado.

¿Como funciona?

Para que un usuario ingrese a mi aplicación, necesito obtener las credenciales de autenticación correspondientes. Estas credenciales pueden consistir en la dirección de correo electrónico y la contraseña del usuario, o un token OAuth proveniente de un proveedor de identidad federada. Luego, es necesario transmitir estas credenciales al SDK de Firebase Authentication. Posteriormente, nuestros servicios de backend verificarán estas credenciales y responderán al cliente.

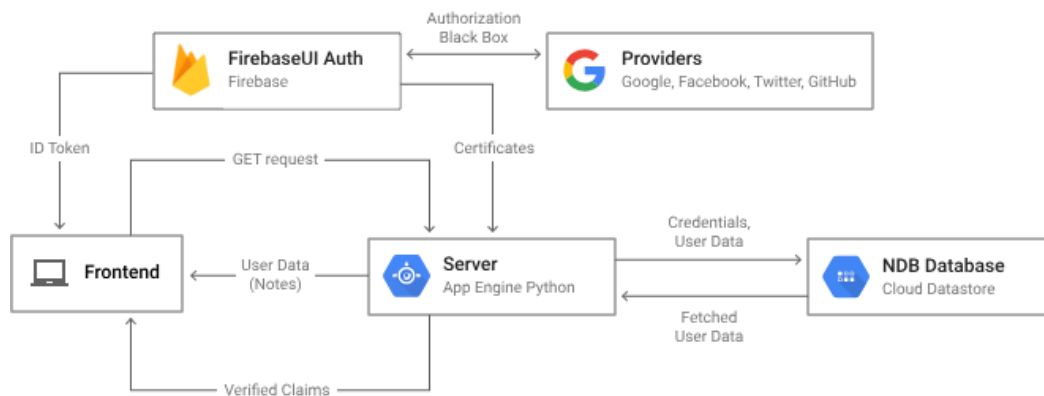


Figura1: Proceso que describe el uso de Gmail en Firebase para un sitio.

Una vez que el acceso se completa con éxito, puedo acceder a la información básica del perfil del usuario y administrar su acceso a los datos almacenados en otros productos de Firebase. Además, tenemos la opción de utilizar el token de autenticación proporcionado para validar la identidad de los usuarios en mis propios servicios de backend. Esto proporciona un nivel adicional de seguridad y control sobre la autenticación del usuario en mi aplicación

Opciones de autenticación de Firebase con Gmail

Si estamos buscando **añadir un botón de acceso de Google** a nuestro sitio web o aplicación, o si alguno utiliza la consola del administrador de Google Workspace para su dominio y si queremos autenticar a los usuarios mediante ese acceso, la opción que podemos considerar es implementar el Acceso con Google. Este servicio se presenta como nuestra biblioteca de cliente para el acceso construida sobre los protocolos OAuth 2.0 y OpenID Connect.

1. El **Acceso con Google**: Es compatible con aplicaciones web, iOS y Android. Se basa en la implementación de OAuth 2.0 de Google, alineada con la especificación de OpenID Connect y certificada por OpenID. **OpenID Connect**, funciona como una capa de identidad sobre el protocolo OAuth 2.0, permitiéndonos recuperar información del perfil del usuario.

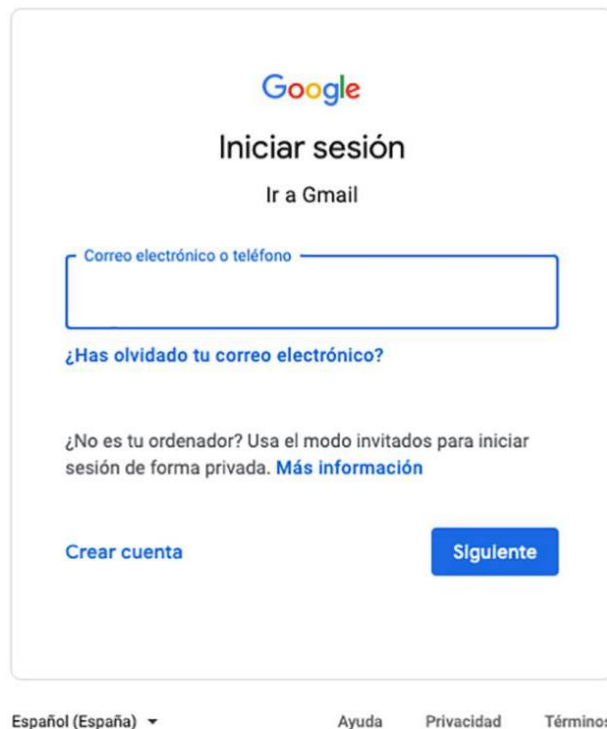
The image shows the Google login interface. At the top is the Google logo. Below it, the text 'Iniciar sesión' (Log in) is centered, followed by a link 'Ir a Gmail' (Go to Gmail). A large text input field is provided for the user's email or phone number, with the placeholder text 'Correo electrónico o teléfono'. Below the input field is a link '¿Has olvidado tu correo electrónico?' (Forgot your email?). Further down, there is a message: '¿No es tu ordenador? Usa el modo invitados para iniciar sesión de forma privada. Más información' (Is this not your computer? Use guest mode to sign in privately. More information). At the bottom left is a link 'Crear cuenta' (Create account), and at the bottom right is a blue button labeled 'Siguiendo' (Next). The footer contains a language selector 'Español (España)' with a dropdown arrow, and three links: 'Ayuda' (Help), 'Privacidad' (Privacy), and 'Términos' (Terms).

Figura 2: Acceso con Google

2. **Identity Platform:** Es otra alternativa que podemos explorar, es un servicio de identidad y autenticación personalizable que brinda opciones flexibles de integración, como SAML, OIDC, correo electrónico y contraseña, redes sociales, teléfono y autenticación personalizada. Este aprovecha la escala, el rendimiento, la red y la seguridad globales de Google Cloud, respaldado por asistencia de nivel empresarial y un ANS para satisfacer las necesidades de cualquier aplicación o servicio. También cuenta con su propio sistema de identidad de usuario. Si ya utiliza Google Workspace para su dominio y desea autenticar usuarios conforme a ese acceso es una buena opción.

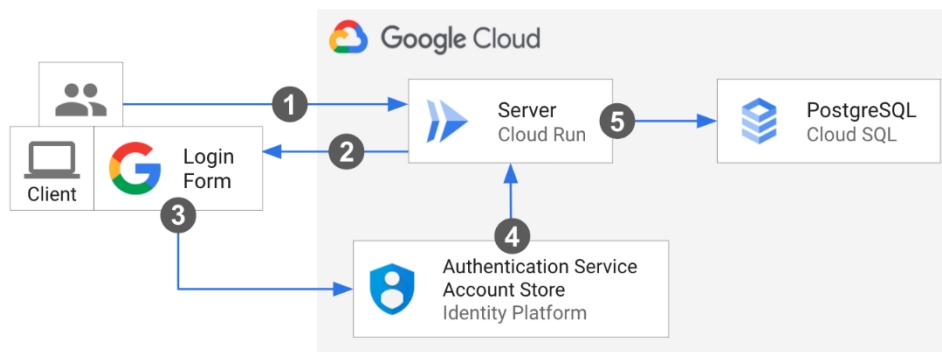


Figura 3: Identity Platform

3. **API de Users:** Adicionalmente, para facilitar tareas como detectar si un usuario completó el proceso de acceso, redireccionar al usuario a la página correspondiente y solicitar al usuario que cree una nueva cuenta de Google si aún no la tiene, podríamos considerar el uso de esta. La aplicación puede acceder a la dirección de correo electrónico del usuario y determinar si es un administrador al iniciar sesión.



Figura 4: API de Users.

4. **Identity-Aware Proxy (IAP):** añade una capa adicional de seguridad y control, podríamos explorar **Identity-Aware Proxy (IAP)**, que agrega una capa de autenticación y autorización frente a recursos para solicitudes externas entrantes. **IAP** protege nuestra aplicación y realiza verificaciones de autenticación y autorización, permitiendo el acceso solo a personas con las funciones de administración de identidades y accesos (IAM) adecuadas. Podríamos habilitar **IAP** para toda la aplicación o para servicios específicos.

Obtener más información sobre cómo configurar **IAP** podría ser útil para nosotros.

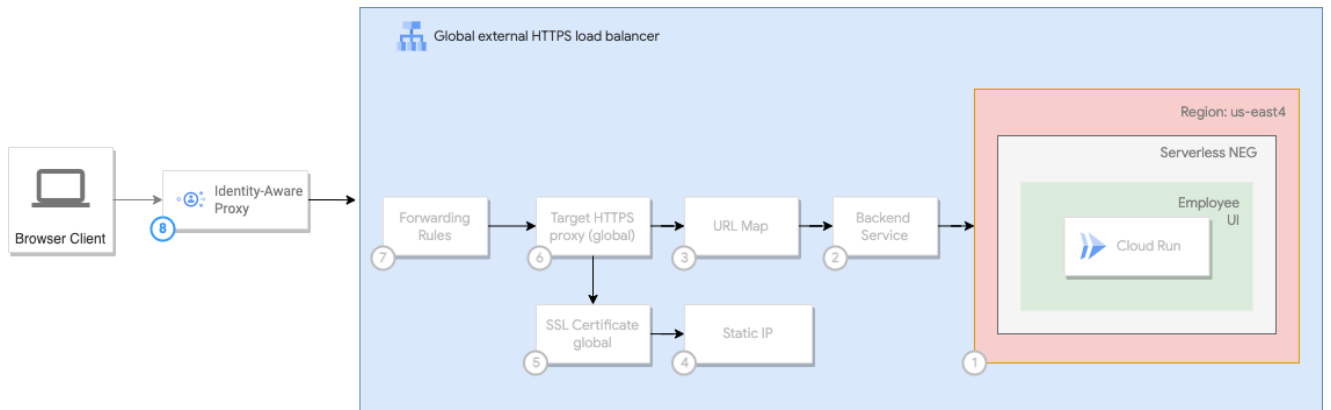
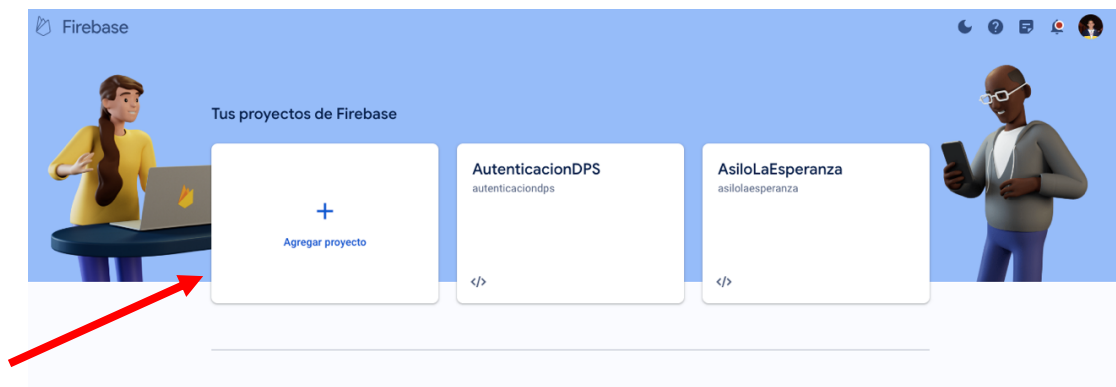


Figura 5: Identity-Aware Proxy (IAP)

2. IMPLEMENTACIÓN DE AUTENTICACIÓN CON FIREBASE UTILIZANDO GOOGLE CLOUD

Para comenzar, es necesario crear el proyecto en la consola de Firebase.



Se coloca un nombre al proyecto

✕ Crear un proyecto(paso 1 de 3)

Comencemos con el nombre de tu proyecto[?]


Nombre del proyecto

logindps-11c1e


Continuar

Se escoge el servicio de autenticación para activarlo en el proyecto.

Almacena y sincroniza datos de app en milisegundos ✕



Authentication
Autentica y administra usuarios



Cloud Firestore
Actualizaciones en tiempo real, consultas poderosas y ajuste de escala automático

[Ver todas las funciones de Compilación](#)

Hacemos clic en “Google” como proveedor de acceso.

Authentication

Users [Sign-in method](#) Templates Usage Settings [Extensiones](#)

Proveedores de acceso

Agrega tu primer método de acceso y comienza a utilizar Firebase Auth

Proveedores nativos

✉ Correo electrónico/contraseña

☎ Teléfono

👤 Anónimo

Proveedores adicionales

🌐 Google

📘 Facebook

🎮 Play Juegos

🎮 Game Center

🍏 Apple

🐙 GitHub

📱 Microsoft

🐦 Twitter

📧 Yahoo

Proveedores personalizados

🔒 OpenID Connect

🔒 SAML

Se activa el servicio de autenticación con Google y se hace clic en “Guardar”.



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 la siguiente [configuración a nivel de proyecto](#) para continuar

Nombre público del proyecto ?

project-1040588492446

Correo electrónico de asistencia del proyecto ?

ricar.arturo@gmail.com

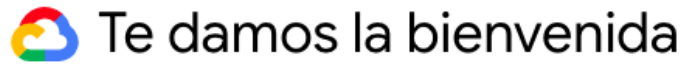
Agrega a la lista de entidades seguras los ID de cliente de proyectos externos (opcional) ?

Configuración del SDK web ?

Cancelar

Guardar

En la consola de administración de Google Cloud, hay que seleccionar el proyecto que tendrá el mismo nombre que el creado en Firebase.



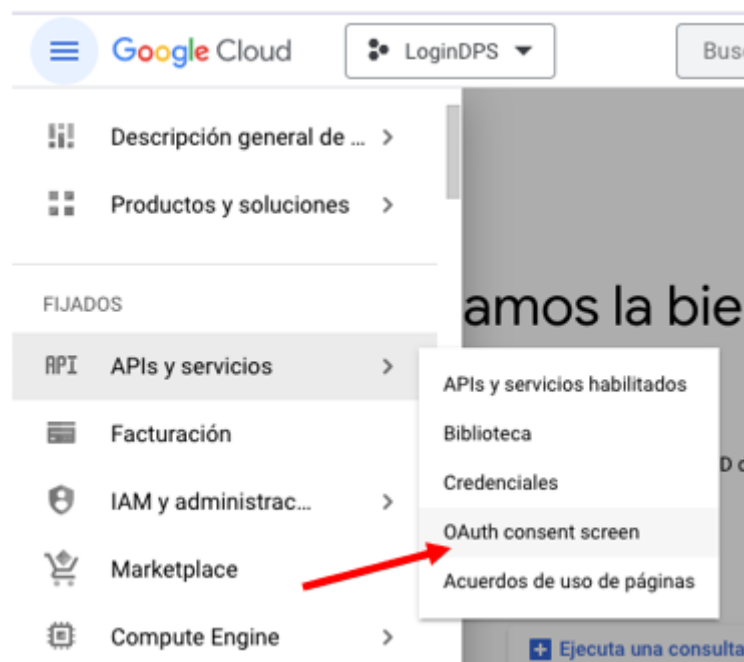
Estás trabajando en [LoginDPS](#)

Número de proyecto: 1040588492446

ID del proyecto: logindps-11c1e

[Panel](#) [Recomendaciones](#)

Posteriormente, se escogerá, en el menú lateral, la configuración de la pantalla de consentimiento de inicio de sesión.



Se escoge "Externos" para los tipos de usuario y se da clic en "Crear".

User Type

☐ Interno ?

Solo está disponible para los usuarios de tu organización. No necesitarás enviar tu app para verificarla. [Obtén más información sobre el tipo de usuario](#)

☒ Externos ?

Está disponible para cualquier usuario de prueba con una Cuenta de Google. Tu app se iniciará en modo de prueba y solo estará disponible para los usuarios que agregues a la lista de usuarios de prueba. Una vez que la app esté lista para enviarse a producción, puede que debas verificarla. [Obtén más información sobre el tipo de usuario](#)

CREAR

Se colocará el nombre de la aplicación y el correo electrónico de contacto.

Información de la aplicación

Esta información aparece en la pantalla de consentimiento y permite que los usuarios finales sepan quién eres y cómo comunicarse contigo

Nombre de la aplicación *

Login DPS

El nombre de la aplicación que solicita el consentimiento

Correo electrónico de asistencia del usuario *

ricar.arturo@gmail.com

For users to contact you with questions about their consent. [Learn more](#)

También será necesario colocar el correo electrónico de desarrollador.

Información de contacto del desarrollador

Direcciones de correo electrónico *

ricar.arturo@gmail.com

Google enviará notificaciones sobre cualquier cambio en tu proyecto a estas direcciones de correo electrónico.

GUARDAR Y CONTINUAR

CANCELAR

Será necesario presionar “Guardar y Continuar” hasta llegar al resumen.

Pantalla de consentimiento de OAuth

[EDITAR](#)

Tipo de usuario

Externa

Nombre de la app

LoginDPS

Correo electrónico de asistencia

ricar.arturo@gmail.com

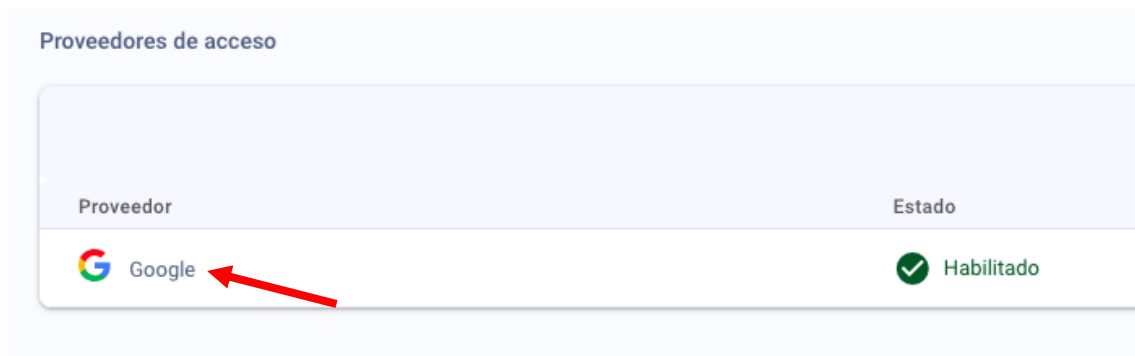
Logotipo de la app

No se proporcionó

Vínculo a la página principal de la aplicación

No se proporcionó

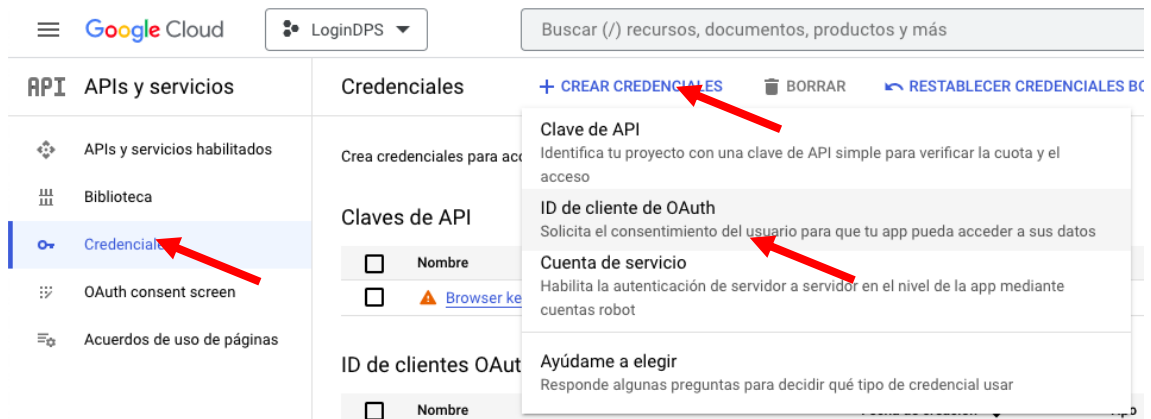
Ahora, regresaremos a Firebase y haremos clic en el proveedor Google.



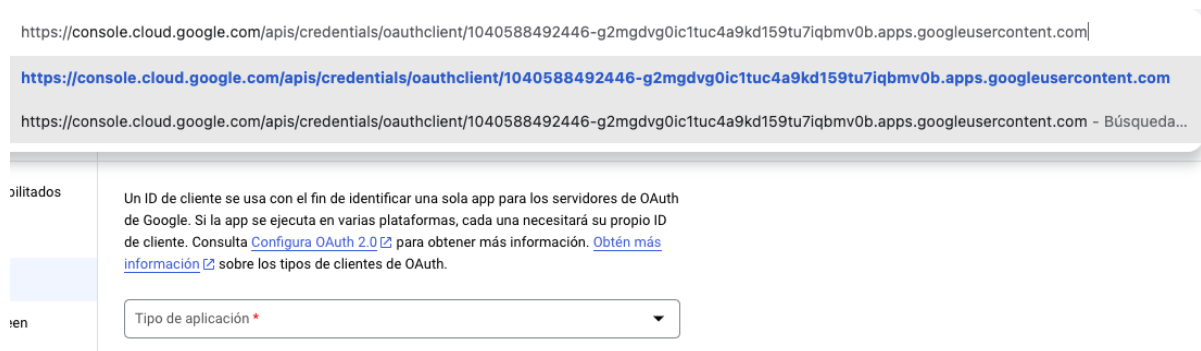
Desplegamos la pestaña de “Configuración del SDK del cliente web” y copiamos el valor de “ID de cliente web”.



En Google Cloud, de nuevo, seleccionamos “Credenciales” -> “Crear Credenciales” -> “ID de cliente de OAuth”.



En la barra de navegación, luego de “.../credentials/oauthclient/” agregaremos el ID copiado desde Firebase y presionaremos ENTER.



Haremos clic en “Guardar”.

URI de redireccionamiento autorizados ?

Para usar con solicitudes de un servidor web

URI 1 *

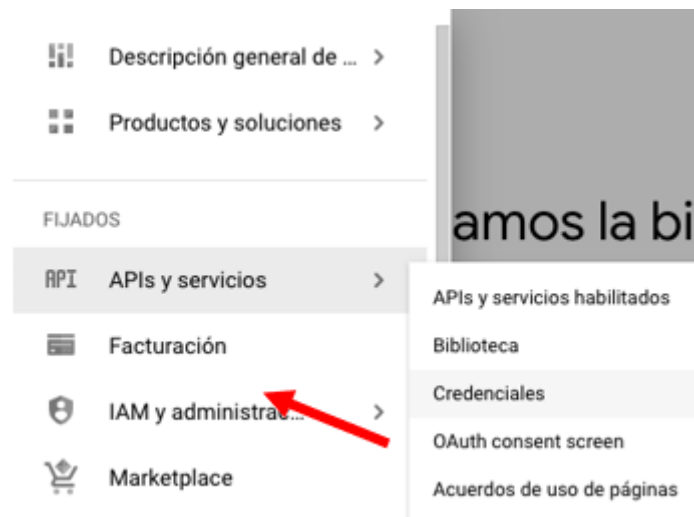
[+ AGREGAR URI](#)

Nota: La configuración puede tardar entre 5 minutos y algunas horas en aplicarse

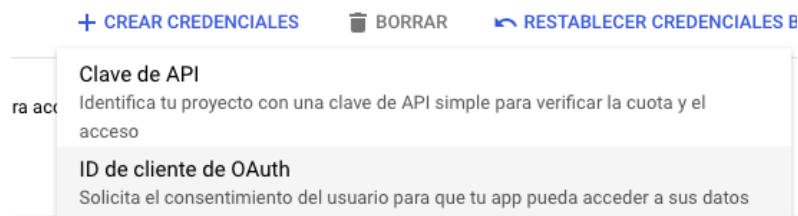
[GUARDAR](#)

[CANCELAR](#)

De nuevo en Google Cloud, en el menú lateral izquierdo, se buscará la opción “Credenciales”.



Clic en “Crear Credenciales” y luego “ID de cliente de OAuth”.



Se escoge la opción “iOS”. Y se mostrará un formulario solicitando información.

Un ID de cliente se usa con el fin de identificar una sola app para los servidores de OAuth de Google. Si la app se ejecuta en varias plataformas, cada una necesitará su propio ID de cliente. Consulta [Configura OAuth 2.0](#) para obtener más información. [Obtén más información](#) sobre los tipos de clientes de OAuth.

Tipo de aplicación *
iOS

Nombre *
Cliente de iOS 1

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

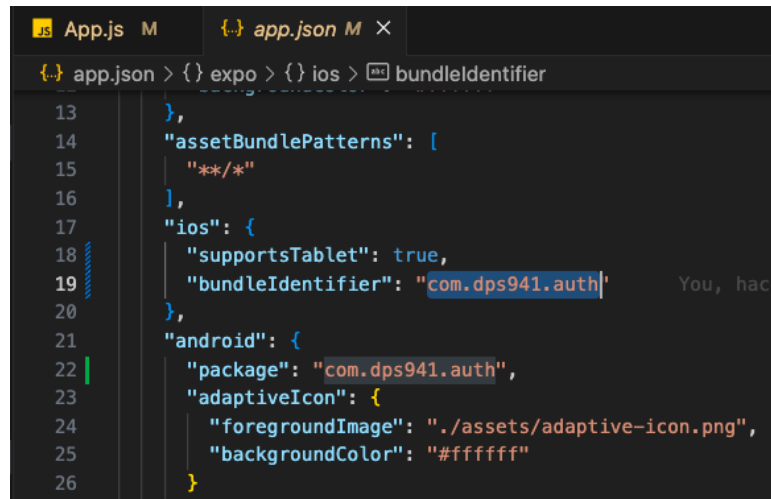
ID del paquete *

Es el identificador de paquete que aparece en el archivo Info.plist de la app

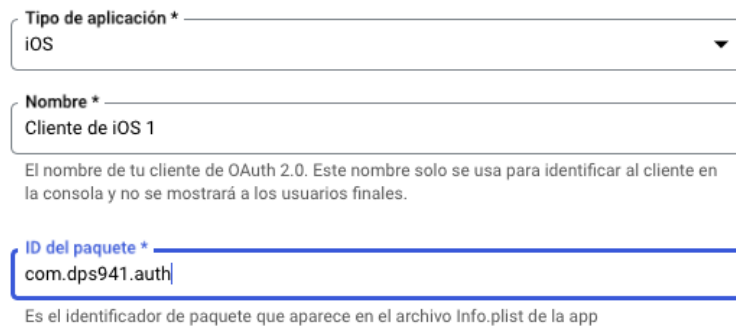
El ID de paquete se obtendrá ejecutando el siguiente comando en la terminal, ubicada en la carpeta del proyecto.

```
○ Ricardo@Rick-MacBook-Air Auth % sudo npx expo prebuild
✓ Created native projects | gitignore skipped
✓ Updated package.json and added index.js entry point for iOS and Android
> Installing using npm
> npm install
```

En app.json aparecerá la información respectiva del nombre del paquete. Este es el nombre que se colocará en Google Cloud.



```
13 },
14 "assetBundlePatterns": [
15   "*/*"
16 ],
17 "ios": {
18   "supportsTablet": true,
19   "bundleIdentifier": "com.dps941.auth",
20 },
21 "android": {
22   "package": "com.dps941.auth",
23   "adaptiveIcon": {
24     "foregroundImage": "./assets/adaptive-icon.png",
25     "backgroundColor": "#ffffff"
26   }
27 }
```



Tipo de aplicación *
iOS

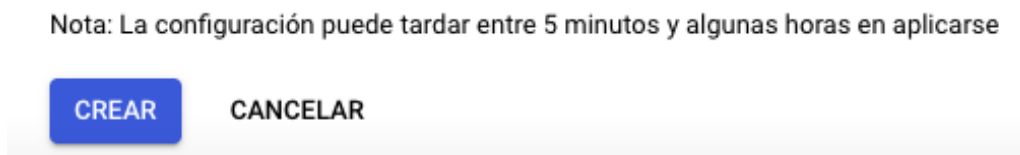
Nombre *
Cliente de iOS 1

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

ID del paquete *
com.dps941.auth

Es el identificador de paquete que aparece en el archivo Info.plist de la app

Haremos clic en “Crear”.



Nota: La configuración puede tardar entre 5 minutos y algunas horas en aplicarse

CREAR CANCELAR

Se creará el ID de Cliente. Es necesario copiar el valor generado.

Se creó el cliente de OAuth

Puedes acceder al ID de cliente y el secreto desde "Credenciales" en API y servicios



OAuth tiene un límite de 100 [registros de alcance confidencial](#) hasta que se verifique la [pantalla de consentimiento de OAuth](#). Es posible que esto requiera un proceso de comprobación que puede demorar varios días.

ID de cliente	1040588492446- b18h6bpmr77qh0vb72sfkoapunflpclr.apps .googleusercontent.com
---------------	---

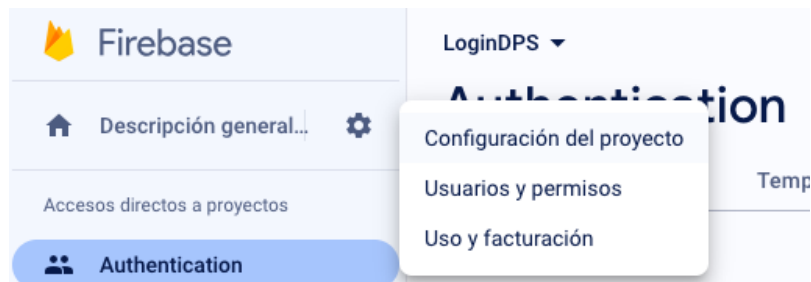
Se sigue el mismo procedimiento para crear las credenciales para Android.

Tipo de aplicación *	Android
Nombre *	Cliente de Android 1
El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.	
Nombre del paquete *	com.dps941.auth
En tu archivo AndroidManifest.xml	
Huella digital del certificado SHA-1 *	5B:3F:AF:D0:58:C6:CD:98:57:EF:4F:AE:5D:A8:4D:E2:5A:B6:45:08

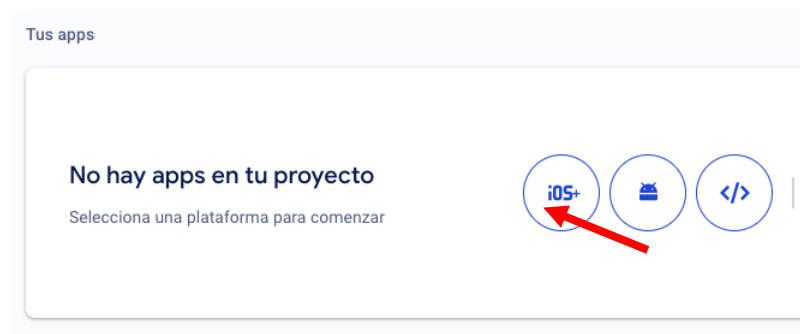
Ahora, en Firebase, en la configuración del proveedor Google, se colocarán los IDs generados en Google Cloud en el apartado de Entidades Seguras.

Agrega a la lista de entidades seguras los ID de cliente de proyectos externos (opcional)	?	^
<input type="text" value="pmr77qh0vb72sfkoapunflpclr.apps.googleusercontent.com"/>	<button>Agregar</button>	
Configuración del SDK web ?		
Agrega a la lista de entidades seguras los ID de cliente de proyectos externos (opcional)	?	^
<input type="text"/>	<button>Agregar</button>	
1040588492446- b18h6bpmr77qh0vb72sfkoapunflpclr.apps.googleusercontent.com 1040588492446- piu0jm9762i8pb4i876qdrkev0jeiu3b.apps.googleusercontent.com		

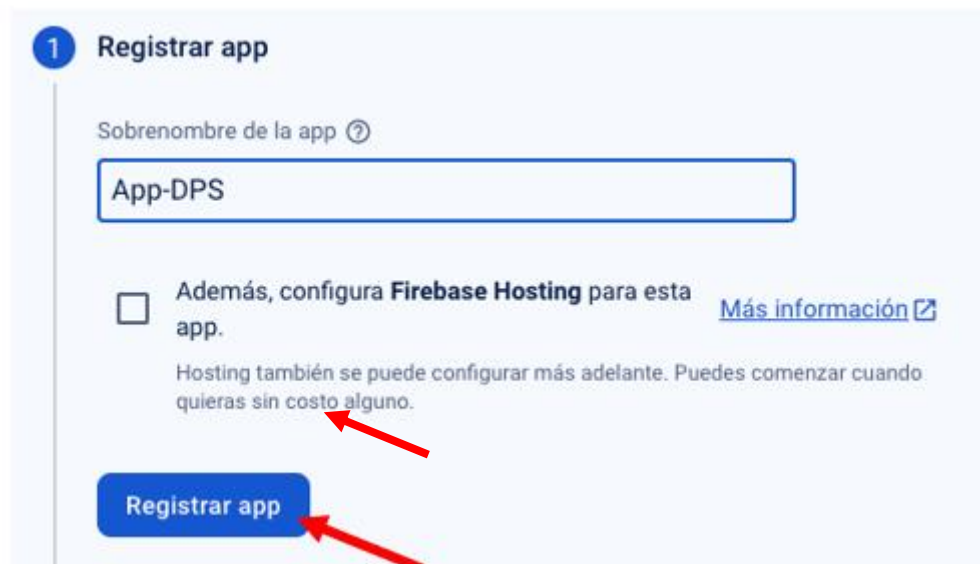
En Firebase, se seleccionará “Configuración del Proyecto”.



Se creará una aplicación web.



Se colocará el nombre de la aplicación y se registrará la aplicación.



Firebase proveerá código que se pegará en un archivo del proyecto.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAwp4fvnajKLqxjLokRqt1mI0JMpXYBAPM",
  authDomain: "logindps-11c1e.firebaseio.com",
  projectId: "logindps-11c1e",
  storageBucket: "logindps-11c1e.appspot.com",
  messagingSenderId: "1040588492446",
  appId: "1:1040588492446:web:928252cffee66024b4b58f"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```



Demostración De Implementación De Autenticación Por Correo De Gmail.

Enlace de video: <https://www.youtube.com/watch?v=92d0qOHfP-k>

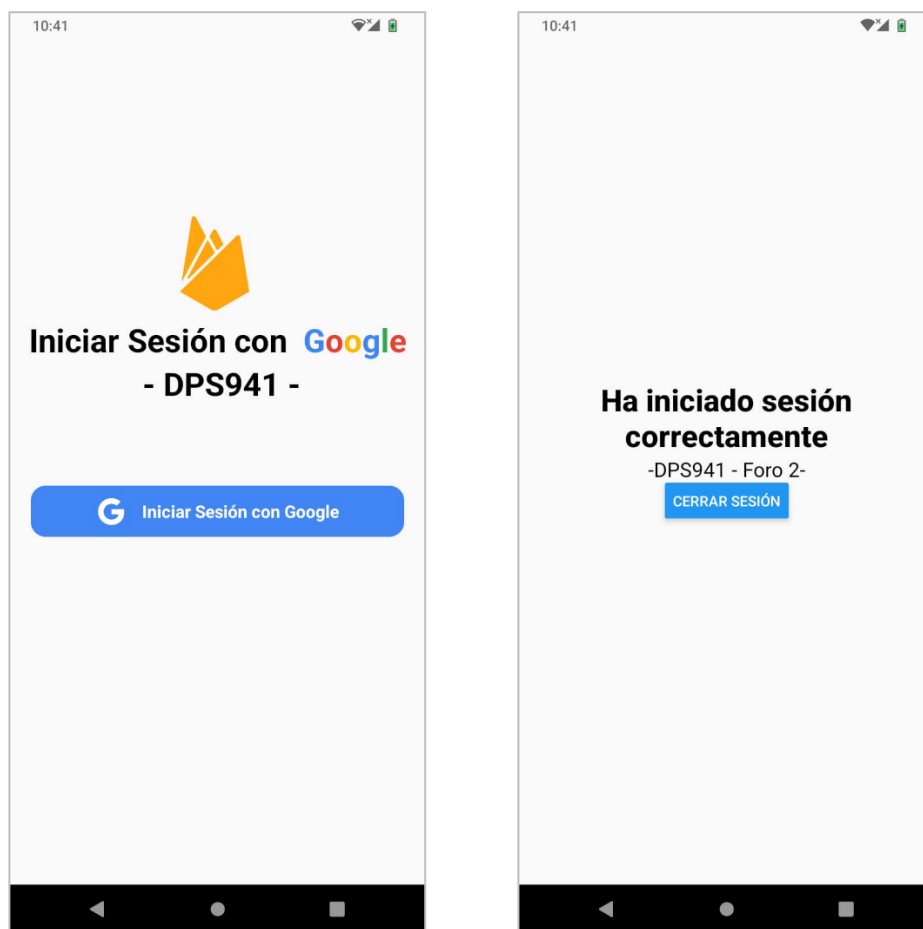


Figura 6: Inicio de sesión exitosa por medio de cuenta de Google.