

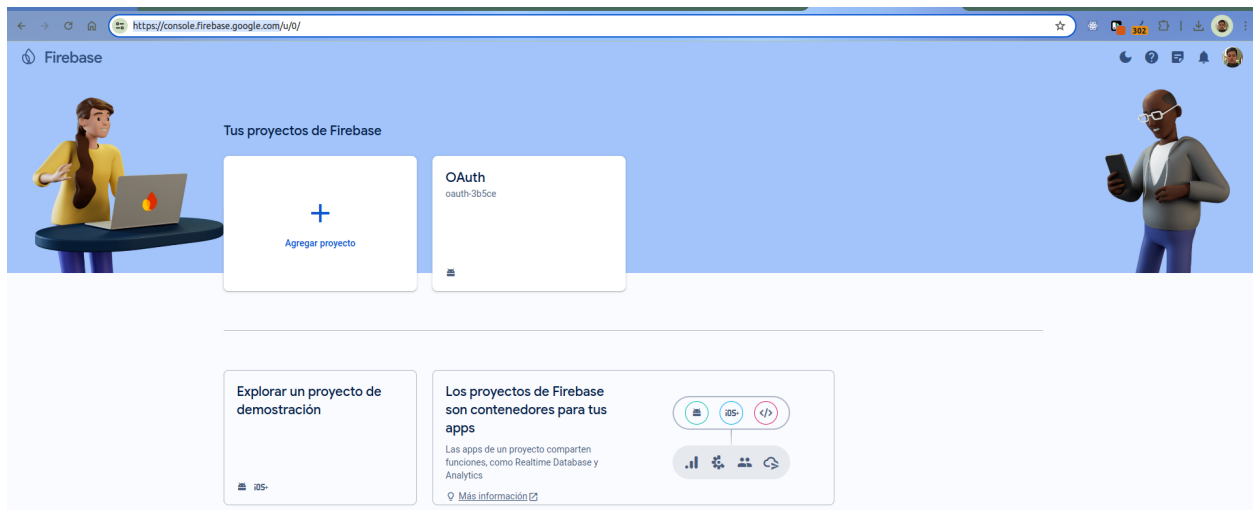
Inicio de sesión con Github

Requerimientos previos

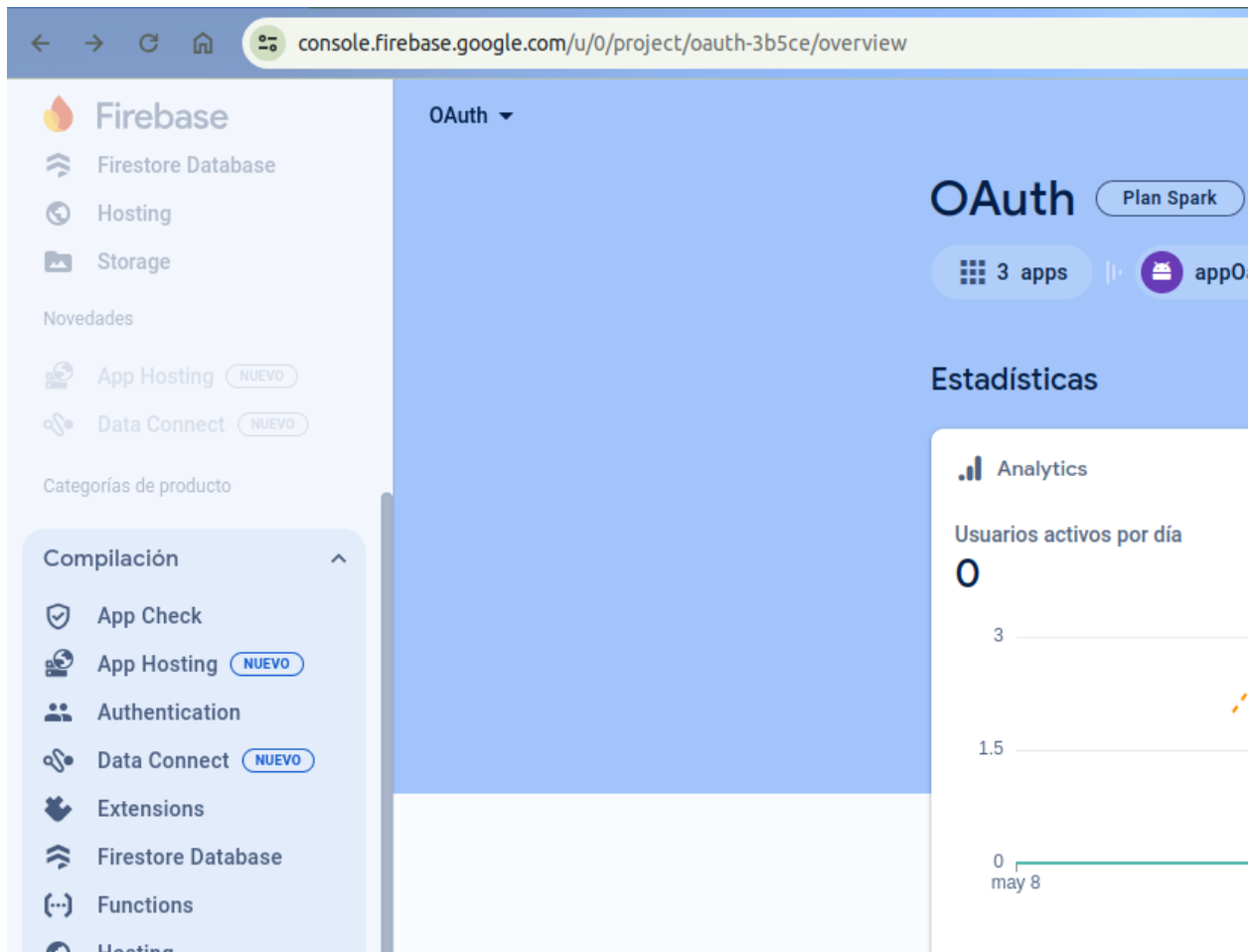
- Haber registrado y configurado tu app para usar los servicios de firebase (agregar el archivo google-services.json a tu proyecto).

Instrucciones

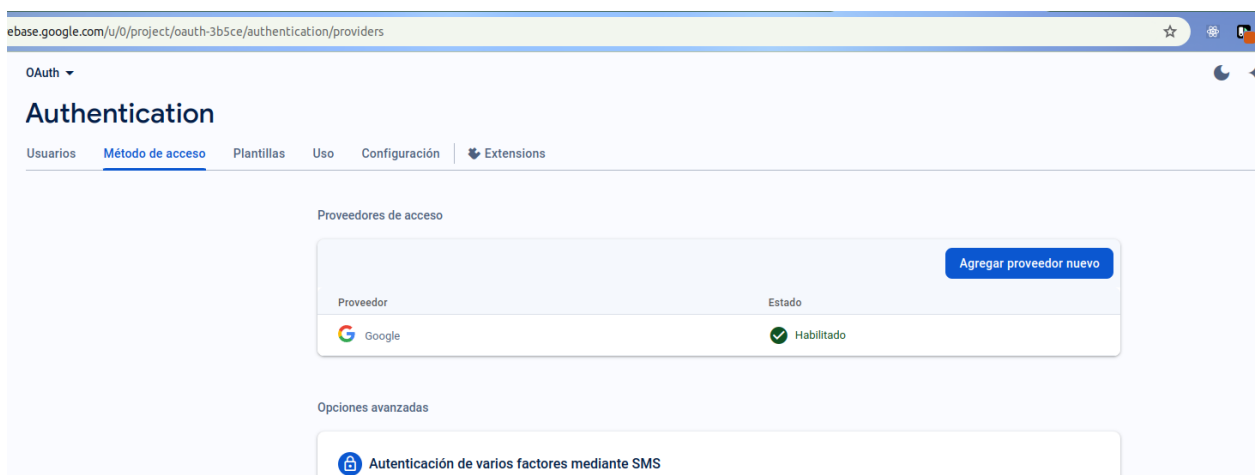
1. Dirigete a la consola de firebase y selecciona un proyecto.



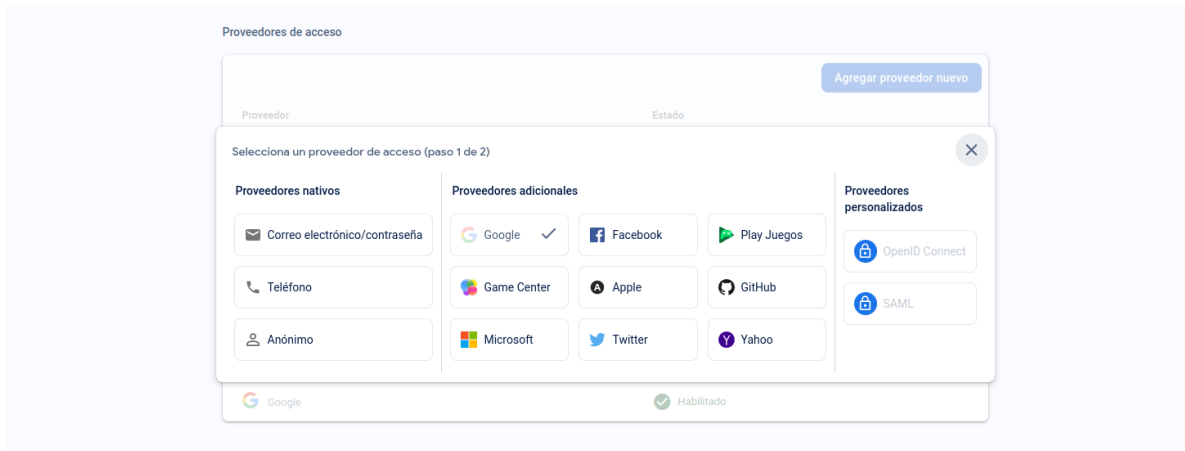
2. En la barra lateral selecciona la opción "Authentication" dentro del apartado "Compilación".



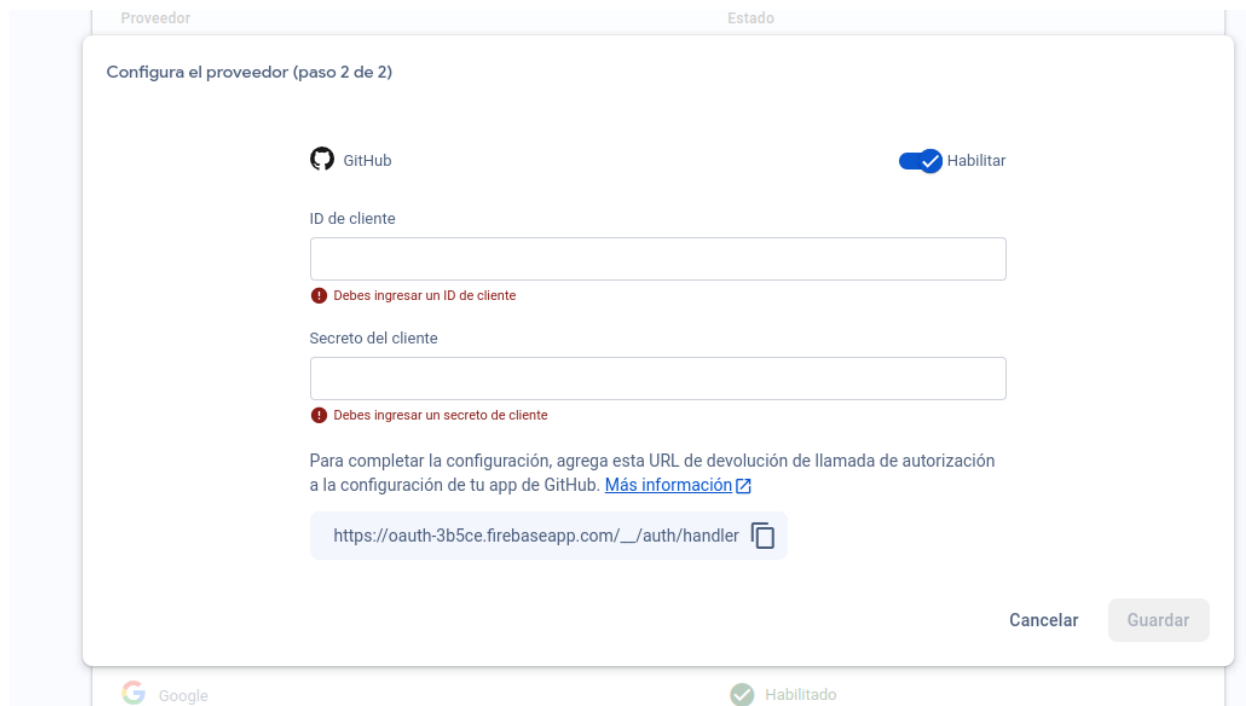
3. Selecciona la pestaña "Método de acceso" y da click en el botón "Agregar proveedor nuevo".



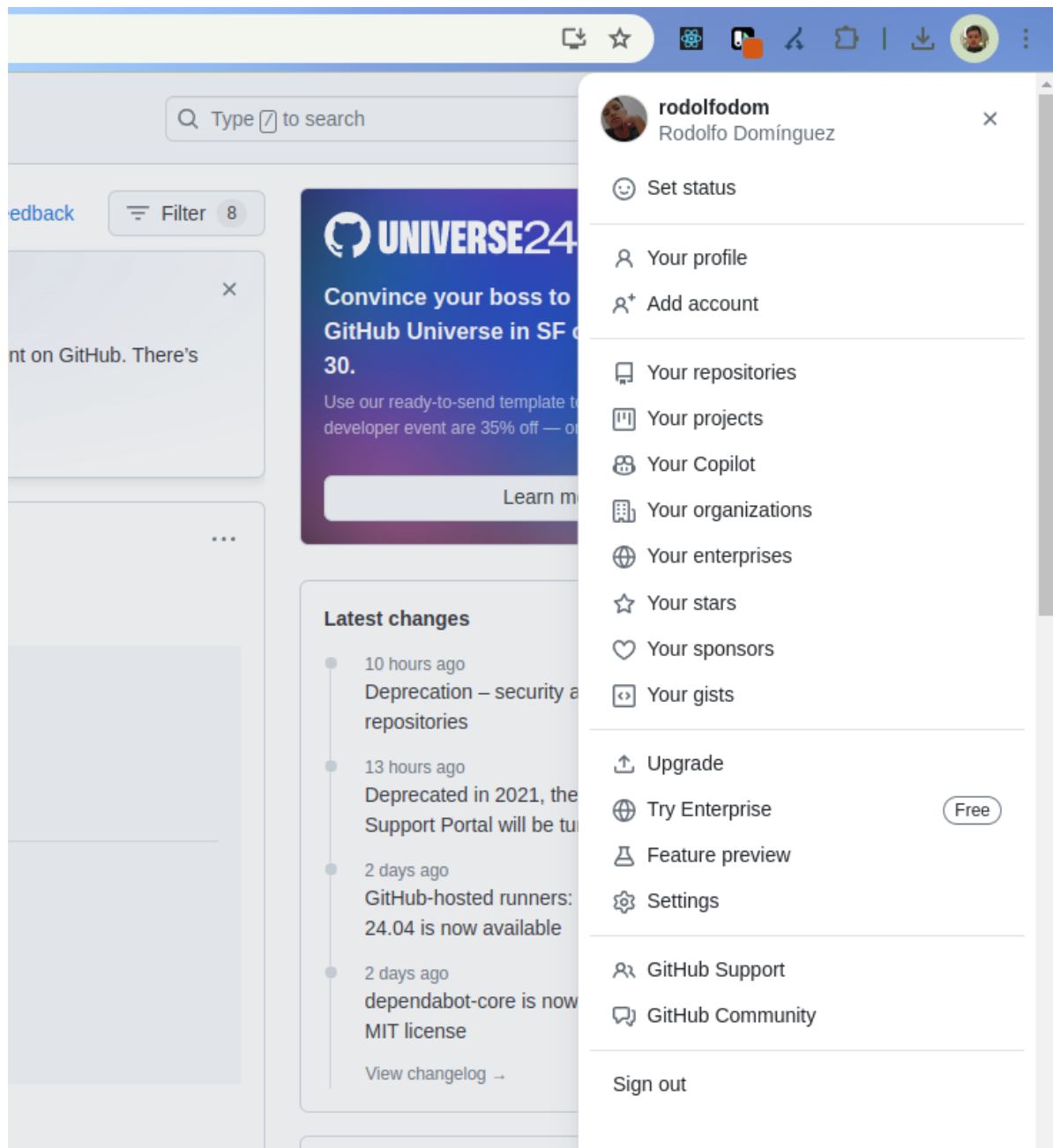
4. En el menú emergente selecciona "Github".



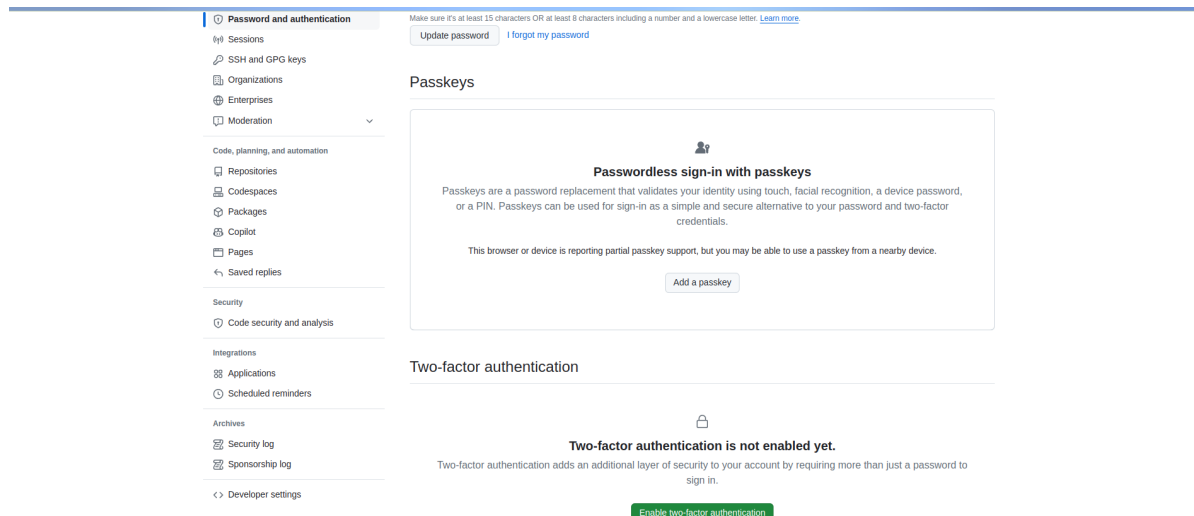
5. Habilita el proveedor haciendo click sobre el switch que se encuentra en la esquina superior derecha. Copia en enlace que aparece en la parte inferior de la ventana, lo usarás en pasos posteriores.



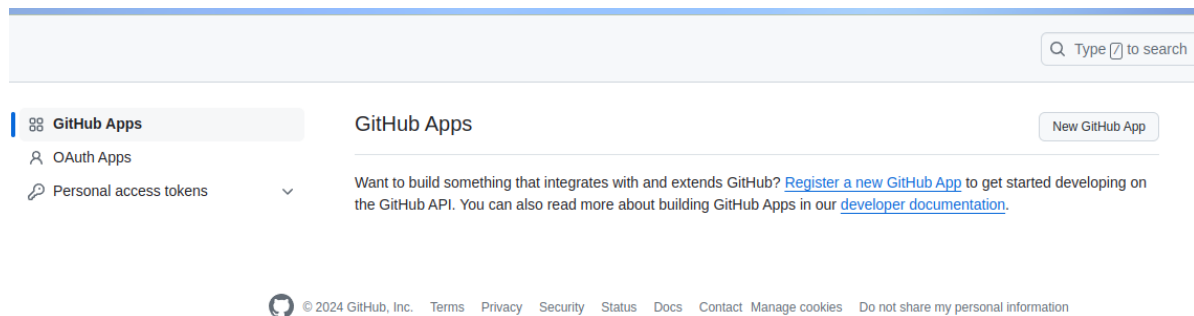
6. Dirígete a [Github](#) e inicia sesión con tu cuenta. Una vez dentro da click en la imagen de tu perfil en la esquina superior derecha y selecciona la opción "Settings".



7. Dentro de la página de configuración, selecciona la opción "Developer Settings" en la barra lateral izquierda (es la última de la lista).



8. Da click en el botón "New Github app".



9. Llena el campo "Github app name" con el nombre de tu elección para identificar tu app. En el campo "Homepage URL" deberas ingresar un URL válido, si no cuentas con uno propio puedes poner cualquier enlace, esto no afectará el proceso de autenticación. Pega el enlace que copiaste en el paso 5 en el campo "Callback URL"

Register new GitHub App

GitHub App name *

mi-nueva-app

The name of your GitHub App.

Write

Preview

Markdown supported

This is displayed to users of your GitHub App

Homepage URL *

https://rodolfodom.github.io/

The full URL to your GitHub App's website.

Identifying and authorizing users

Add Callback URL

The full URL to redirect to after a user authorizes an installation.

Read our [Callback URL documentation](#) for more information.

Callback URL

https://oauth-3b5ce.firebaseio.com/__/auth/handler

Delete

☒ Expire user authorization tokens

This will provide a `refresh_token` which can be used to request an updated access token when this access token expires.

☐ Request user authorization (OAuth) during installation

Requests that the installing user grants access to their identity during installation of your App

Read our [Identifying and authorizing users for GitHub Apps documentation](#) for more information.

☐ Enable Device Flow

Allow this GitHub App to authorize users via the Device Flow

10. Desmarca la casilla "Active" dentro de la sección de webhooks.

Users will be redirected to this URL after installing your GitHub App to complete additional setup.

☐ **Redirect on update**

Redirect users to the 'Setup URL' after installations are updated (E.g. repositories added/removed).

Webhook

☐ **Active**

We will deliver event details when this hook is triggered.

Webhook URL

Events will POST to this URL. Read our [webhook documentation](#) for more information.

Webhook secret (optional)

Read our [webhook secret documentation](#) for more information.

11. En la sección **"Where can this GitHub App be installed?"** selecciona la opción "Any account" y presiona "Create Github app".

Based on the permissions you've selected, what events would you like to subscribe to?

☐ **Installation target** ⓘ

A GitHub App installation target is renamed.

☐ **Meta** ⓘ

When this App is deleted and the associated hook is removed.

☐ **Security advisory** ⓘ

Security advisory published, updated, or withdrawn.

Where can this GitHub App be installed?

☐ **Only on this account**

Only allow this GitHub App to be installed on the @rodolfoodom account.

☒ **Any account**

Allow this GitHub App to be installed by any user or organization.

Create GitHub App

Cancel

12. Serás redirigido a la siguiente página.

Registration successful. You must [generate a private key](#) in order to install your GitHub App.

[Settings](#) / [Developer settings](#) / [GitHub Apps](#) / [mi-nueva-app](#)


General

Permissions & events

Install App

Advanced

Optional features

Public page 

About

Owned by: [@rodolfodom](#)

App ID: 899580

Using your App ID to get installation tokens? You can now [use your Client ID instead](#).

Client ID: lv23lip0KNwD3psBHDBF

[Revoke all user tokens](#)

GitHub Apps can use OAuth credentials to identify users. Learn more about identifying users by reading our [integration developer documentation](#).

Client secrets

[Generate a new client secret](#)

You need a client secret to authenticate as the application to the API.

Basic information


GitHub App name *

mi-nueva-app

The name of your GitHub App.

Write

Preview

 Markdown supported


This is displayed to users of your GitHub App


13. Genera un nuevo Secreto de cliente dando click en "Generate a new client secret" y copialo.

Client secrets

[Generate a new client secret](#)

Make sure to copy your new client secret now. You won't be able to see it again.


Client secret

✓ 1071ace5942a4e861d74033113759ce2fa493cdb 

Added now by [rodolfodom](#)

Never used

You cannot delete the only client secret. Generate a new client secret first.

[Delete](#)

14. Pega tu token secreto en la configuración de firebase.

Proveedor Estado

Configura el proveedor (paso 2 de 2)

GitHub Habilitar

ID de cliente

Debes ingresar un ID de cliente

Secreto del cliente

1071ace5942a4e861d74033113759ce2fa493cdb

Para completar la configuración, agrega esta URL de devolución de llamada de autorización a la configuración de tu app de GitHub. [Más información](#)

https://oauth-3b5ce.firebaseio.com/_/auth/handler

Cancelar Guardar

15. Identifica tu id de cliente y copialo.

[Settings](#) / [Developer settings](#) / [GitHub Apps](#) / [mi-nueva-app](#)

General

Permissions & events

Install App

Advanced

Optional features

Public page

About

Owned by: [@rodolfodom](#)

App ID: 899580

Using your App ID to get installation tokens? You can now [use your Client ID instead](#).

Client ID: [lv23lip0KNwD3psBHDBF](#)


[Revoke all user tokens](#)

GitHub Apps can use OAuth credentials to identify users. Learn more about identifying users by reading our [integration developer documentation](#).

Client secrets

[Generate a new client secret](#)

Make sure to copy your new client secret now. You won't be able to see it again.

 **Client secret**

✓ 1071ace5942a4e861d74033113759ce2fa493cdb

Added 4 minutes ago by [rodolfodom](#)

Never used

[Delete](#)

You cannot delete the only client secret. Generate a new client secret first.

16. Pega tu id de cliente en la configuración de firebase y presiona "Guardar".

Inicio de sesión con Github

9

Proveedor

Estado

Agregar proveedor nuevo

Configura el proveedor (paso 2 de 2)

GitHub

Habilitar

ID de cliente

lv23lip0KNwD3psBHDBF

Secreto del cliente

1071ace5942a4e861d74033113759ce2fa493cdb

Para completar la configuración, agrega esta URL de devolución de llamada de autorización a la configuración de tu app de GitHub. [Más información](#)

https://oauth-3b5ce.firebaseio.com/_/auth/handler

Cancelar

Guardar

Google

Habilitado

- No olvides guardar los cambios cada vez que generes un nuevo token secreto.

Webhook

☐ Active

We will deliver event details when this hook is triggered.

Webhook URL

Events will POST to this URL. Read our [webhook documentation](#) for more information.

Webhook secret (optional)

Read our [webhook secret documentation](#) for more information.

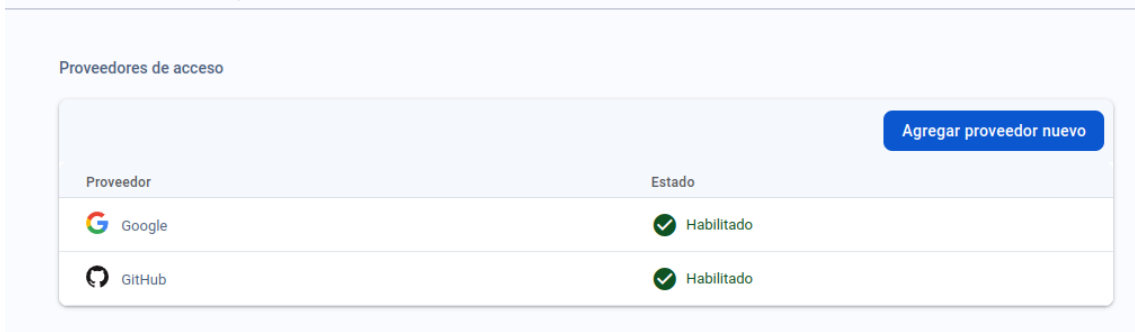
Save changes

Cancel

- Con esto finaliza la configuración de firebase. Podras ver que Github ahora esta habilitado como porveedor de acceso.

Inicio de sesión con Github

10



Ahora escribiremos el código necesario para implementar el inicio de sesión con Github en nuestra app. Para este punto deberías haber creado un proyecto nuevo en Android Studio y haber activado el data binding.

1. Asegurate de incluir las siguientes dependencias en tu archivo build.gradle.kts (Module:app).

```
dependencies {  
    // Import the BoM for the Firebase platform  
    implementation(platform("com.google.firebase:firebase-bom:32.0.0"))  
  
    // Add the dependency for the Firebase Authentication library  
    // When using the BoM, you don't specify versions in Firebase dependencies  
    implementation("com.google.firebase:firebase-auth")  
}
```

2. agrega las siguientes variables en la clase en la que implementarás la autenticación.

```
private lateinit var firebaseAuth: FirebaseAuth  
private lateinit var binding: ActivityMainBinding // data binding  
private val provider = OAuthProvider.newBuilder("github.com")
```

3. Agrega la siguiente función a tu código, esta se encargará de ejecutar el proceso de autenticación.

```
fun signInWithProvider(provider: OAuthProvider.Builder) {  
    // [START auth_oidc_provider_signin]
```

```

        firebaseAuth
            .startActivityForSignInWithProvider(this, provider.I
            .addOnSuccessListener {
                // User is signed in.
                // IdP data available in
                // authResult.getAdditionalUserInfo().getProfile
                // The OAuth access token can also be retrieved
                // ((OAuthCredential)authResult.getCredential())
                // The OAuth secret can be retrieved by calling
                // ((OAuthCredential)authResult.getCredential())
                Toast.makeText(this, "User is signed in with gi
            }
            .addOnFailureListener {
                // Handle failure.
                Toast.makeText(this, "Authentication failed", To
            }
        // [END auth_oidc_provider_signin]
    }
}

```

Asegurate de editar el código en los bloques `addOnSuccessListener` y `addOnFailureListener` para definir las acciones que se realizarán cuando la autenticación se exitosa y cuando falle.

4. La definición de tu metodo `onCreate` deberá ser parecida a esta.

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)
    firebaseAuth = FirebaseAuth.getInstance() //obtener la :

    /*
    Aisgnar la ejecución del método signInWithProvider al l
    el proceso de autenticación en tu app. Recuerda cambiar
    que le asignaste a tu botón en archivo cml de tu activid
    */
}

```

```
*/  
binding.btnGithub.setOnClickListener {  
    signInWithProvider(provider)  
}  
}
```