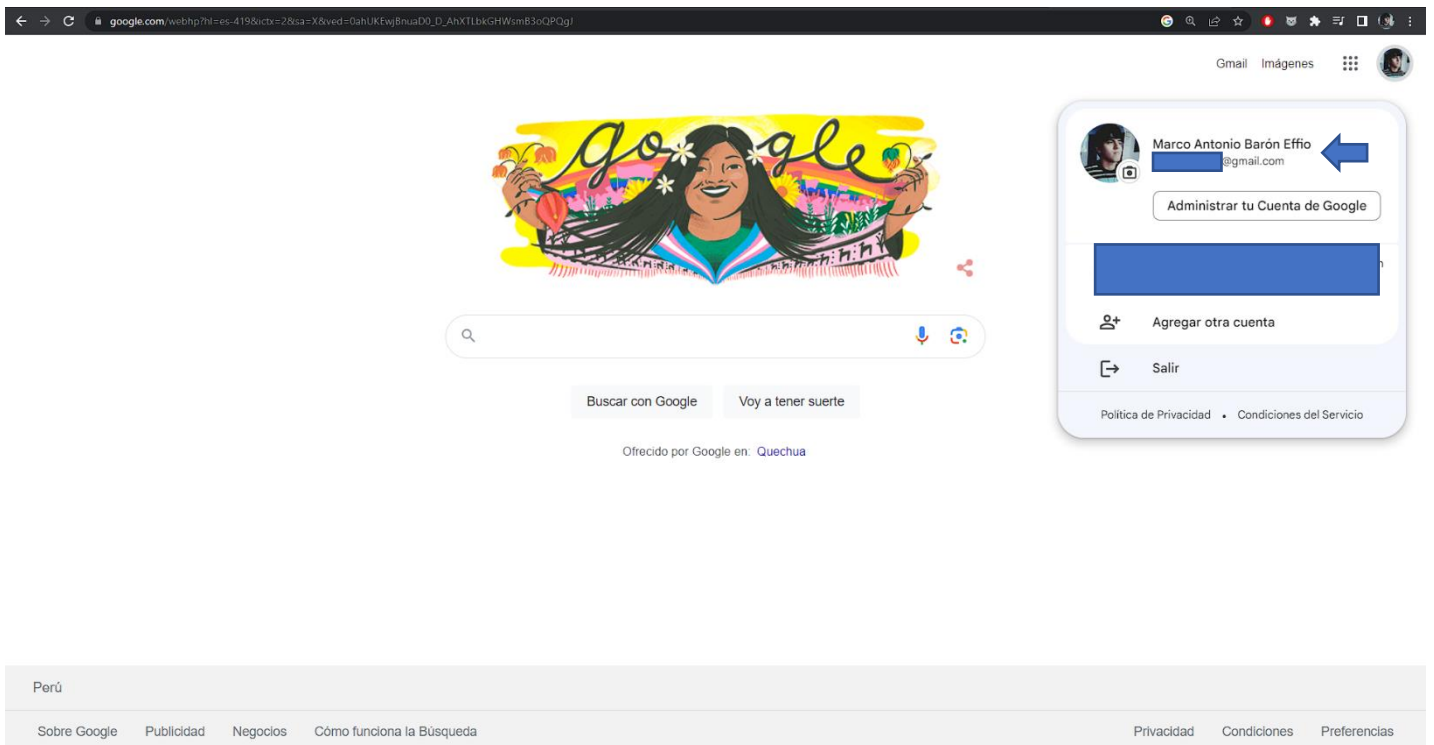
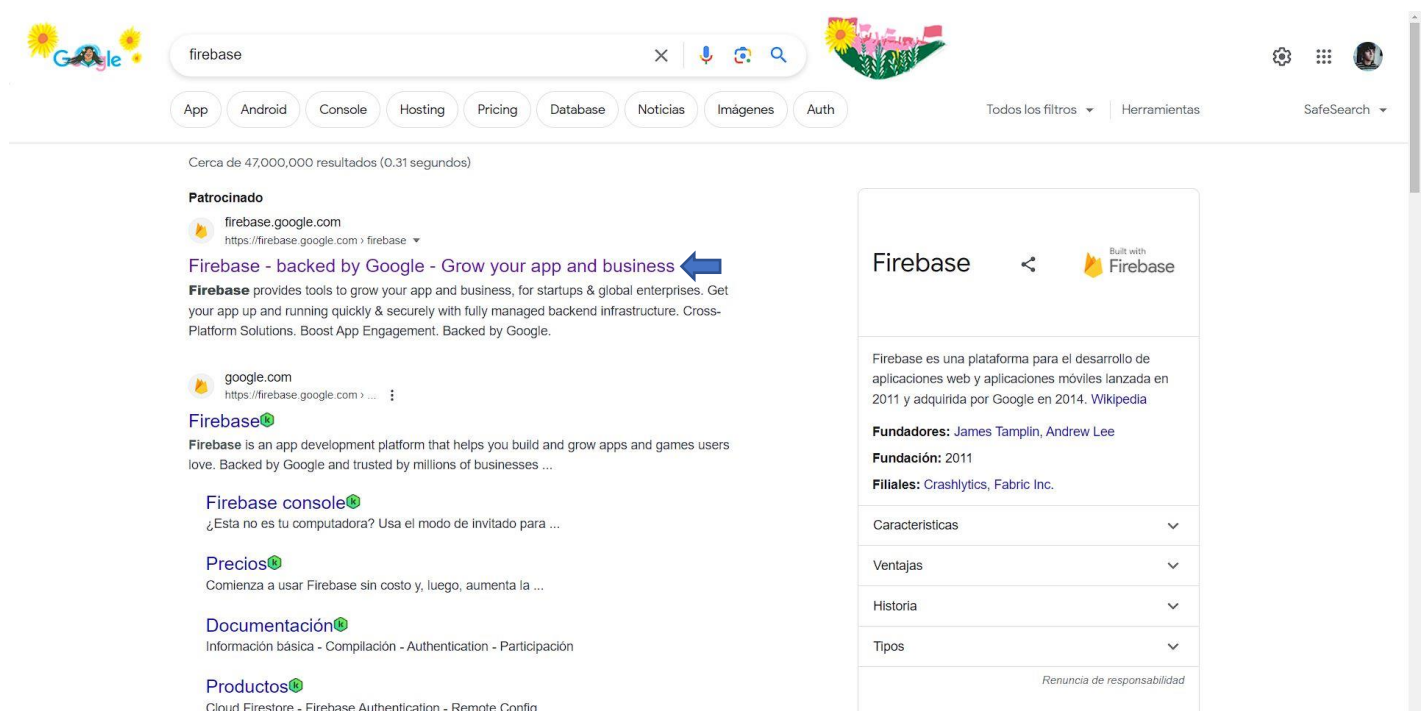


## Implementar Firebase en Android Studio

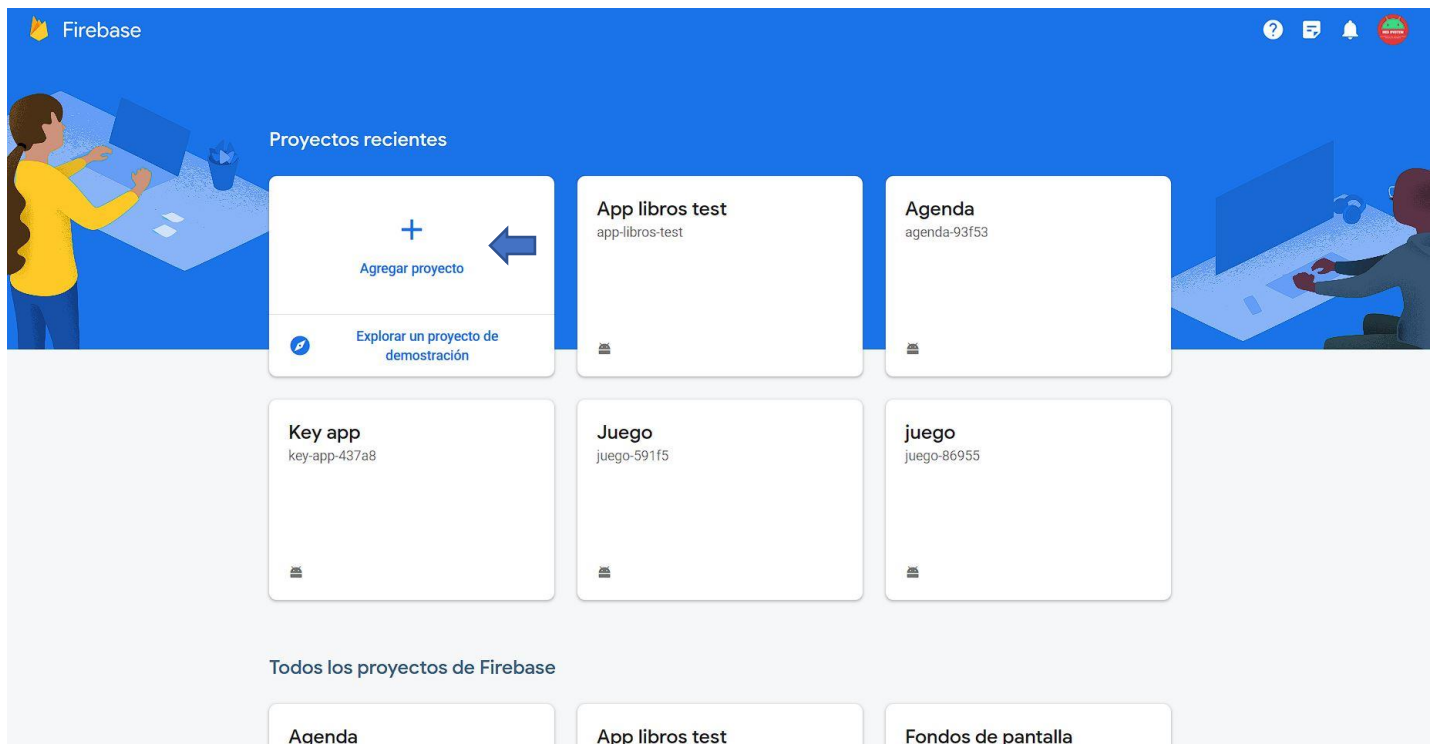
[1] Primero debemos iniciar sesión con una cuenta de **Google** u **Outlook**



[2] Nos dirigimos al navegador de nuestra preferencia y escribimos **Firebase**  
- Abrimos la primera página.



[3] Presionamos en **Agregar proyecto**



[4] Establecemos el nombre de nuestro proyecto y presionamos en **Continuar**.



[5] En este apartado presionamos en **Continuar**


×


Crear un proyecto(paso 2 de 3)


## para tu proyecto de Firebase


Google Analytics es una solución de analítica ilimitada y gratuita que permite usar la segmentación, los informes y otras funciones en Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing y Cloud Functions.


Google Analytics habilita las siguientes funciones:


 Pruebas A/B [?](#)

 Usuarios que no experimentan fallas [?](#)

 Segmentación de usuarios y orientación a ellos en los productos de Firebase [?](#)


 Activadores de Cloud Functions basados en eventos [?](#)

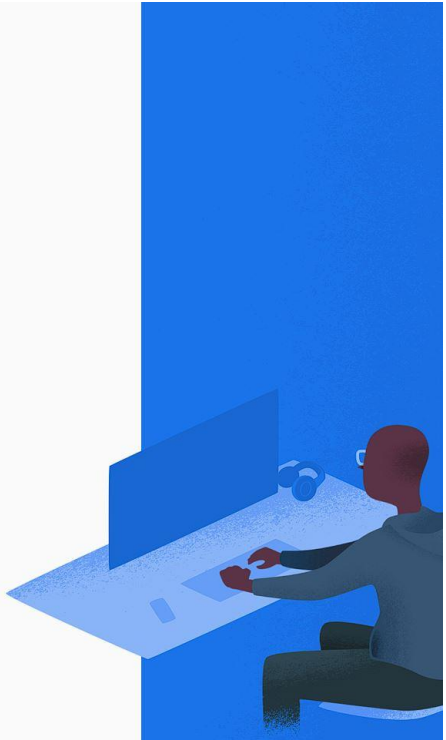
 Informes ilimitados y gratuitos [?](#)

 **Habilitar Google Analytics para este proyecto**  
Recomendado

Anterior

Continuar






[6] Seleccionamos **Default Account for Firebase** y luego presionamos en **Crear proyecto**.


×


Crear un proyecto(paso 3 de 3)

## Configurar Google Analytics

Elige o crea una cuenta de Google Analytics [?](#)

 Default Account for Firebase




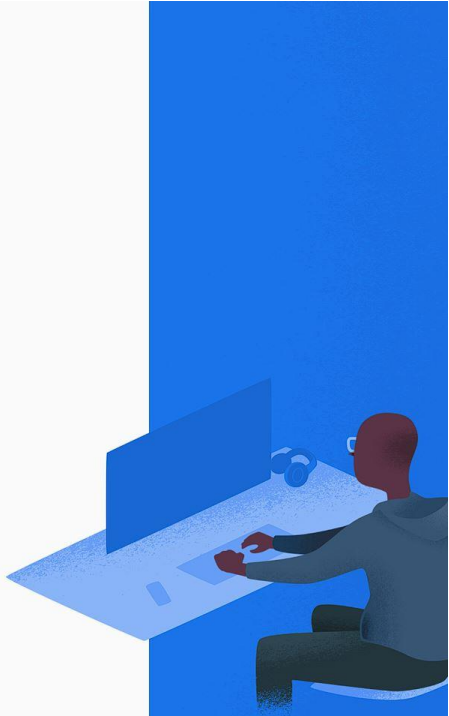
Crear automáticamente una nueva propiedad en esta cuenta 

Quando se cree el proyecto, también se creará una nueva propiedad de Google Analytics en la cuenta de Google Analytics que elijas y se vinculará a tu proyecto de Firebase. Este vínculo permitirá el flujo de datos entre los productos. Los datos que se exportan de tu propiedad de Google Analytics a Firebase están sujetos a las Condiciones del Servicio de Firebase, mientras que los datos de Firebase que se importan a Google Analytics están sujetos a las Condiciones del Servicio de Google Analytics. [Obtén más información](#)

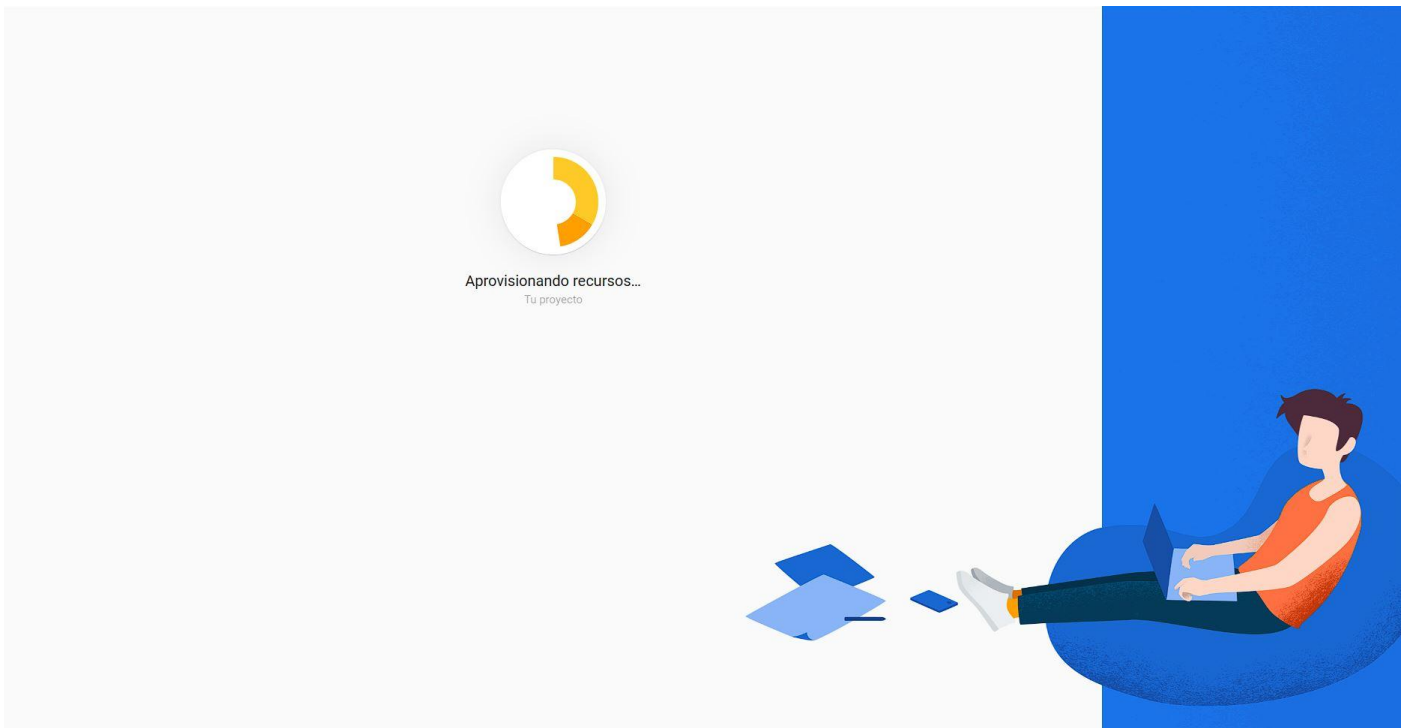
Anterior

Crear proyecto

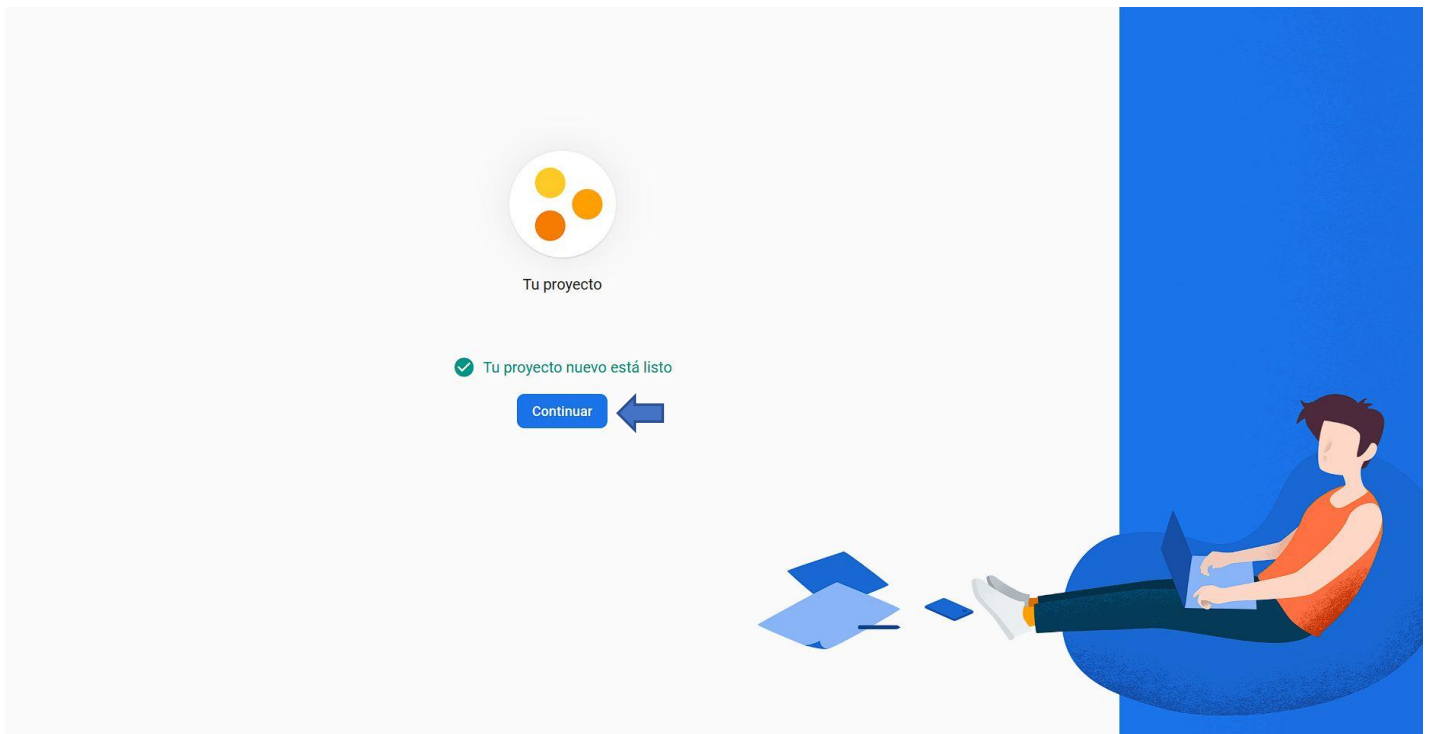




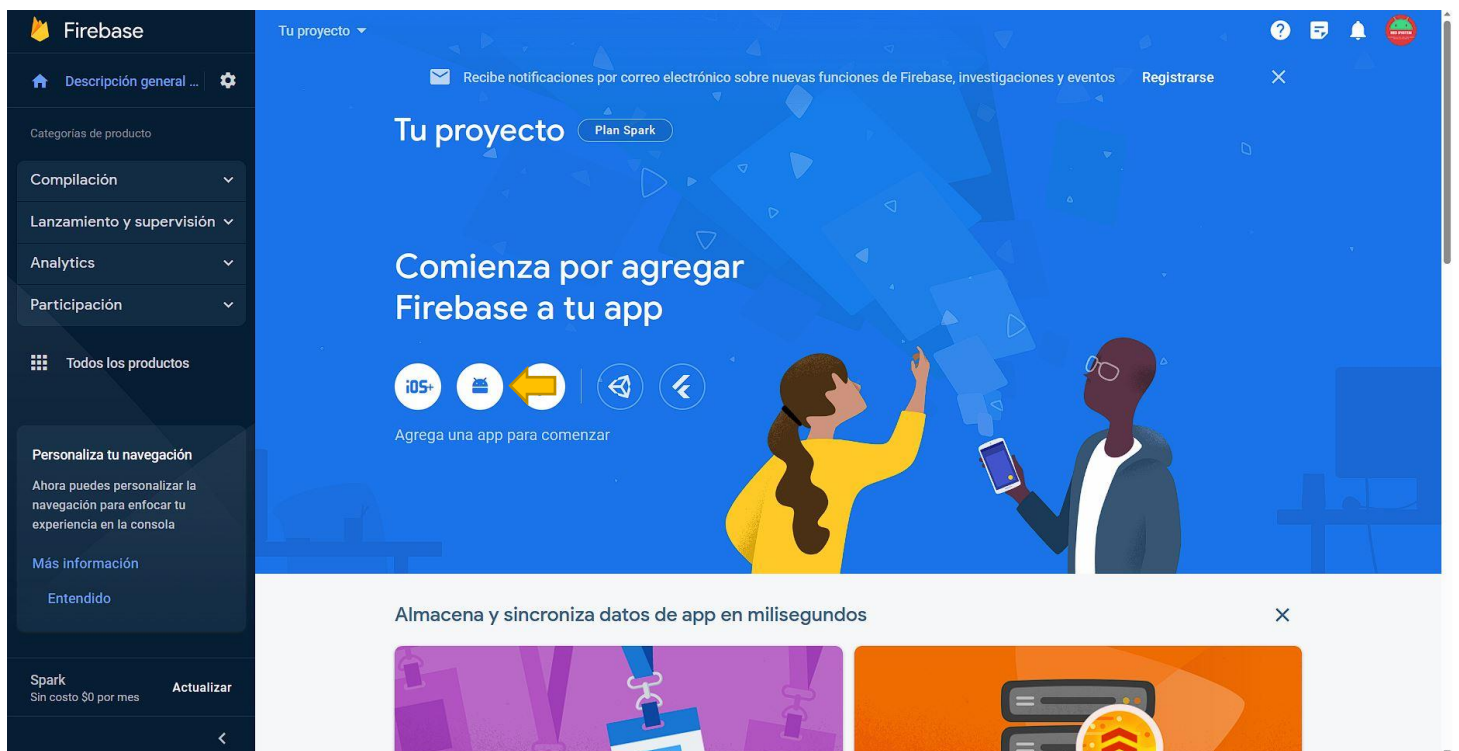
[7] Esperamos un momento



[8] Presionamos en **Continuar**



[9] Seleccionamos el icono de Android

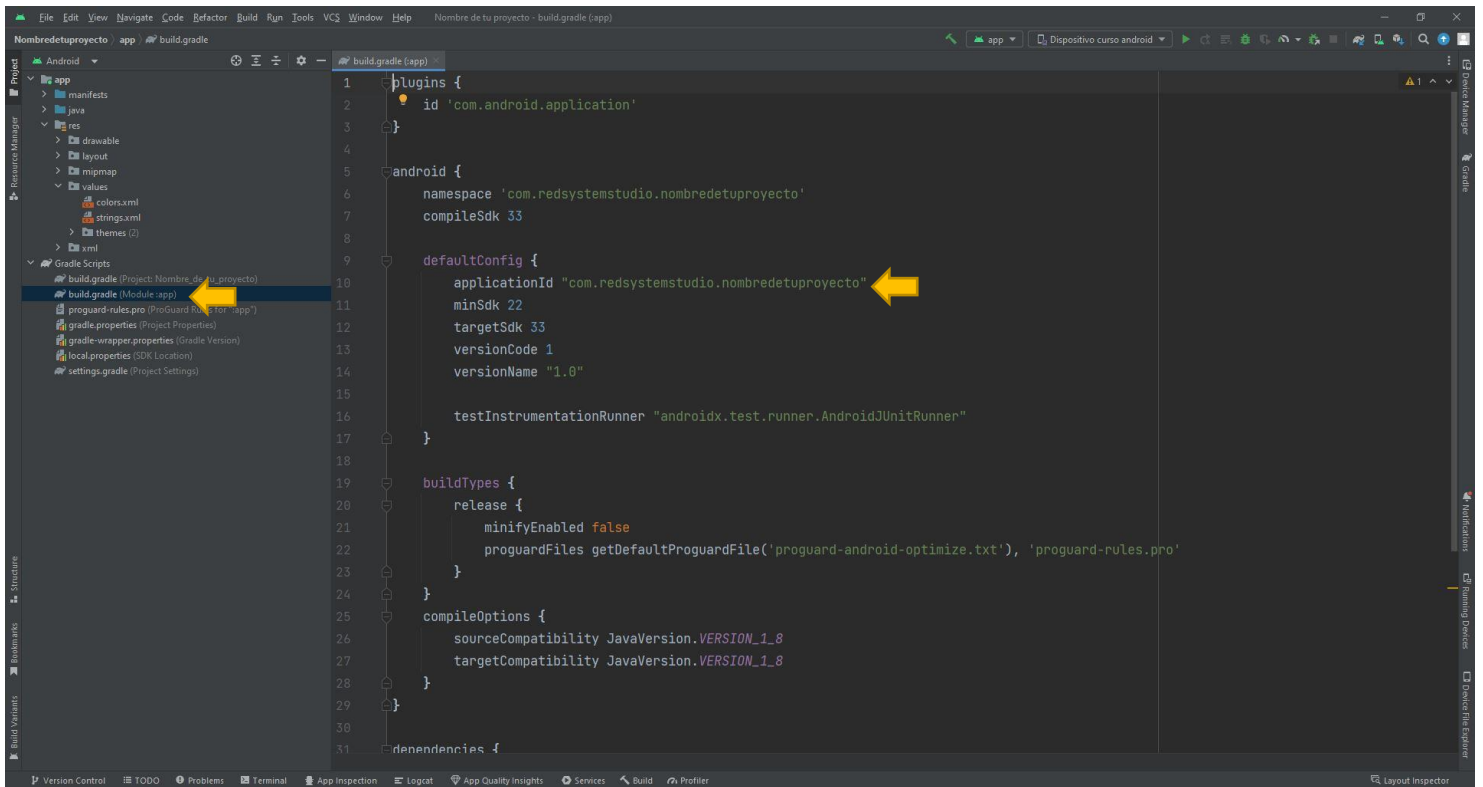


[10] Empezaremos implementando el nombre del paquete

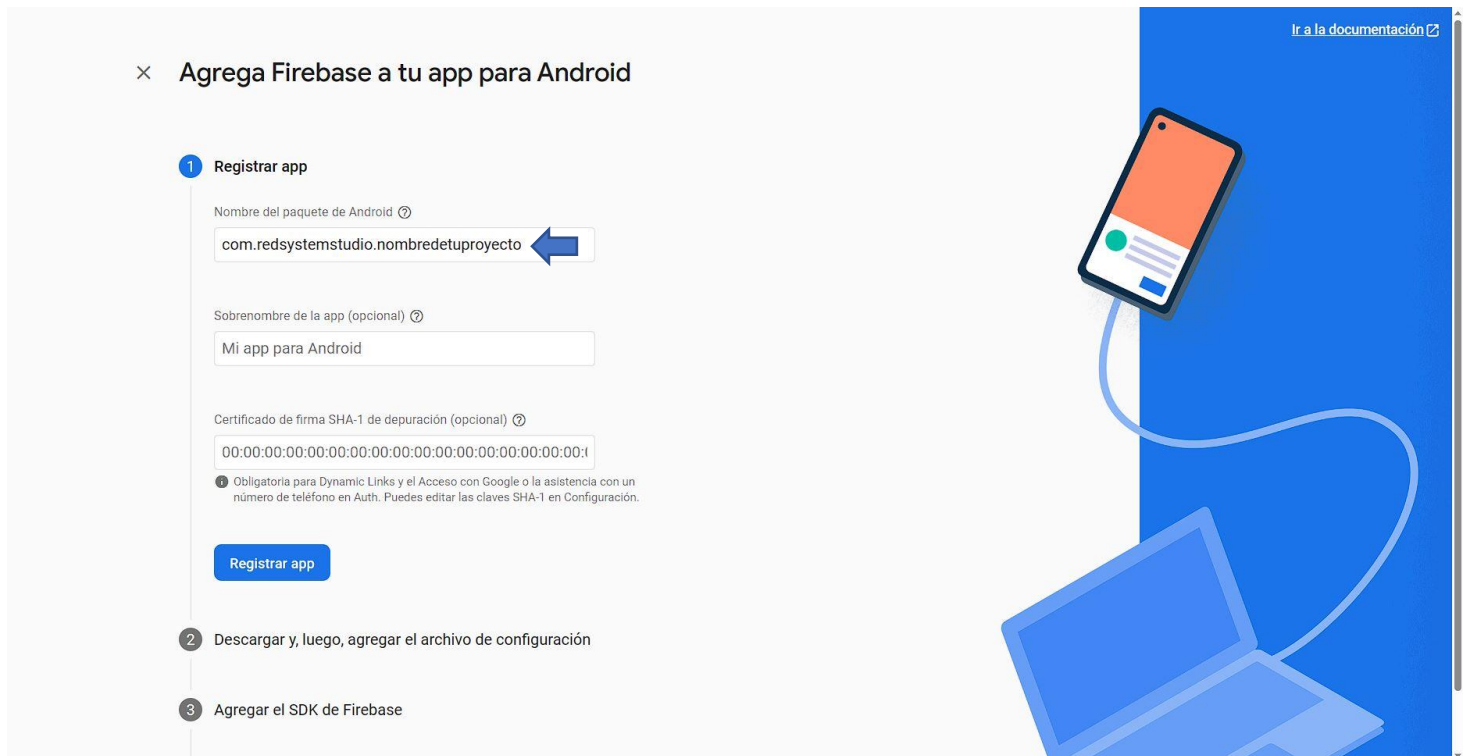
[illegible]



[11] La encontraremos en este apartado, lo seleccionamos sin los “ ”.



[12] Pegamos el nombre del paquete



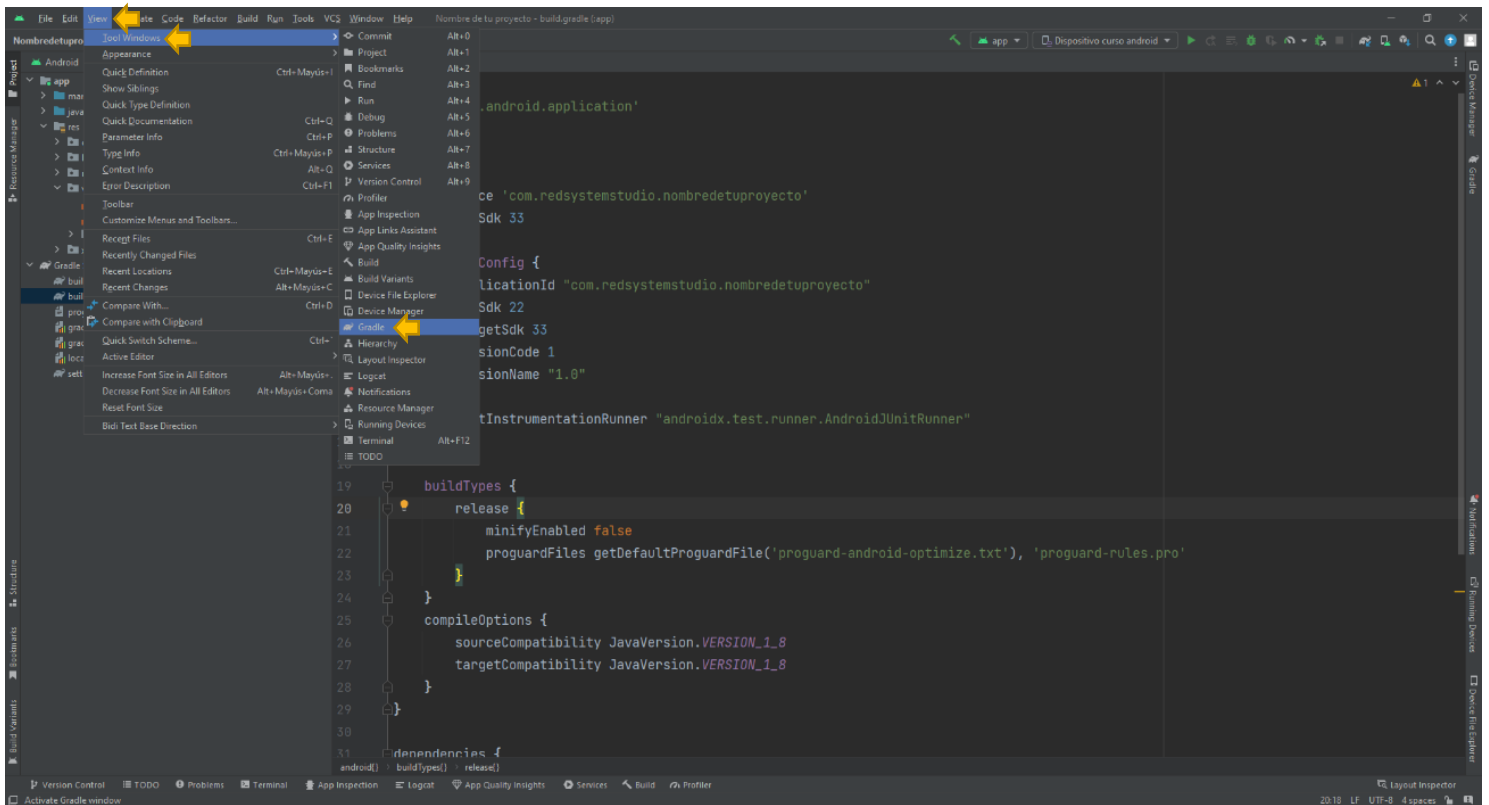
[13] Aquí asignaremos nuevamente el nombre de nuestro proyecto

[illegible]

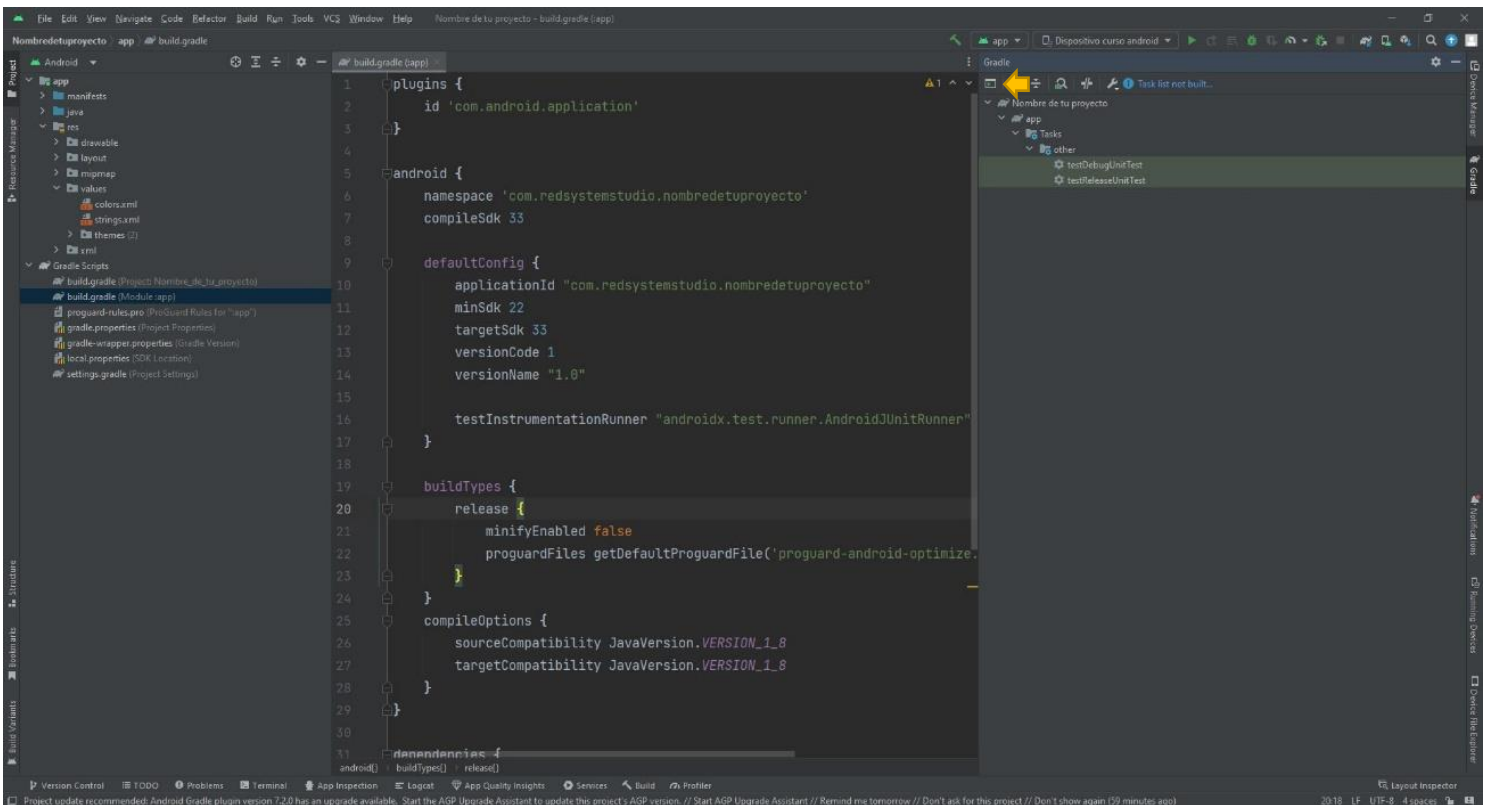
[14] Para obtener la firma SHA-1, haremos lo siguiente

[illegible]

- Nos dirigimos a Android Studio -> Presionamos en **View -> Tool Windows -> Gradle**

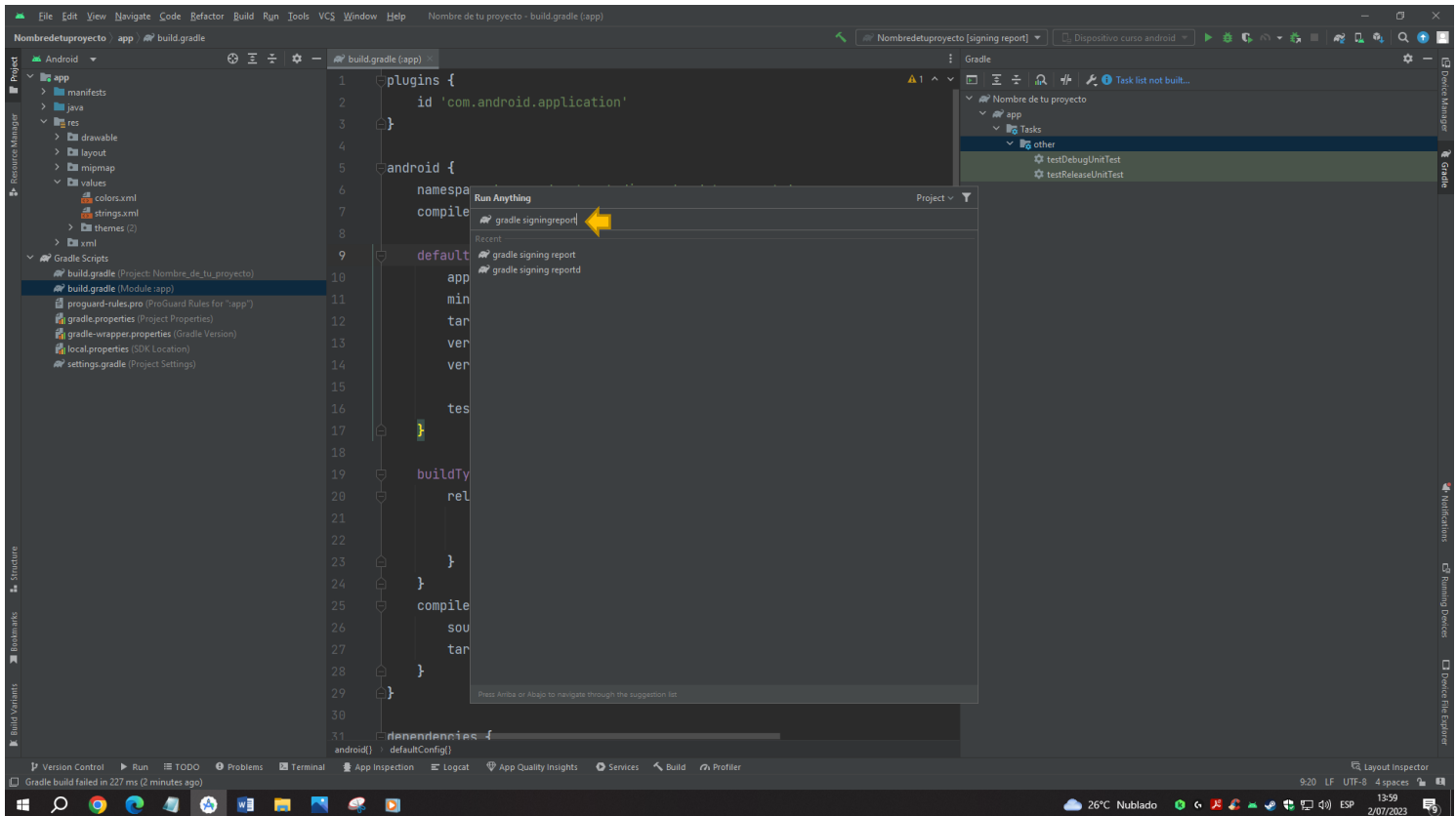


[15] Presionamos en este icono

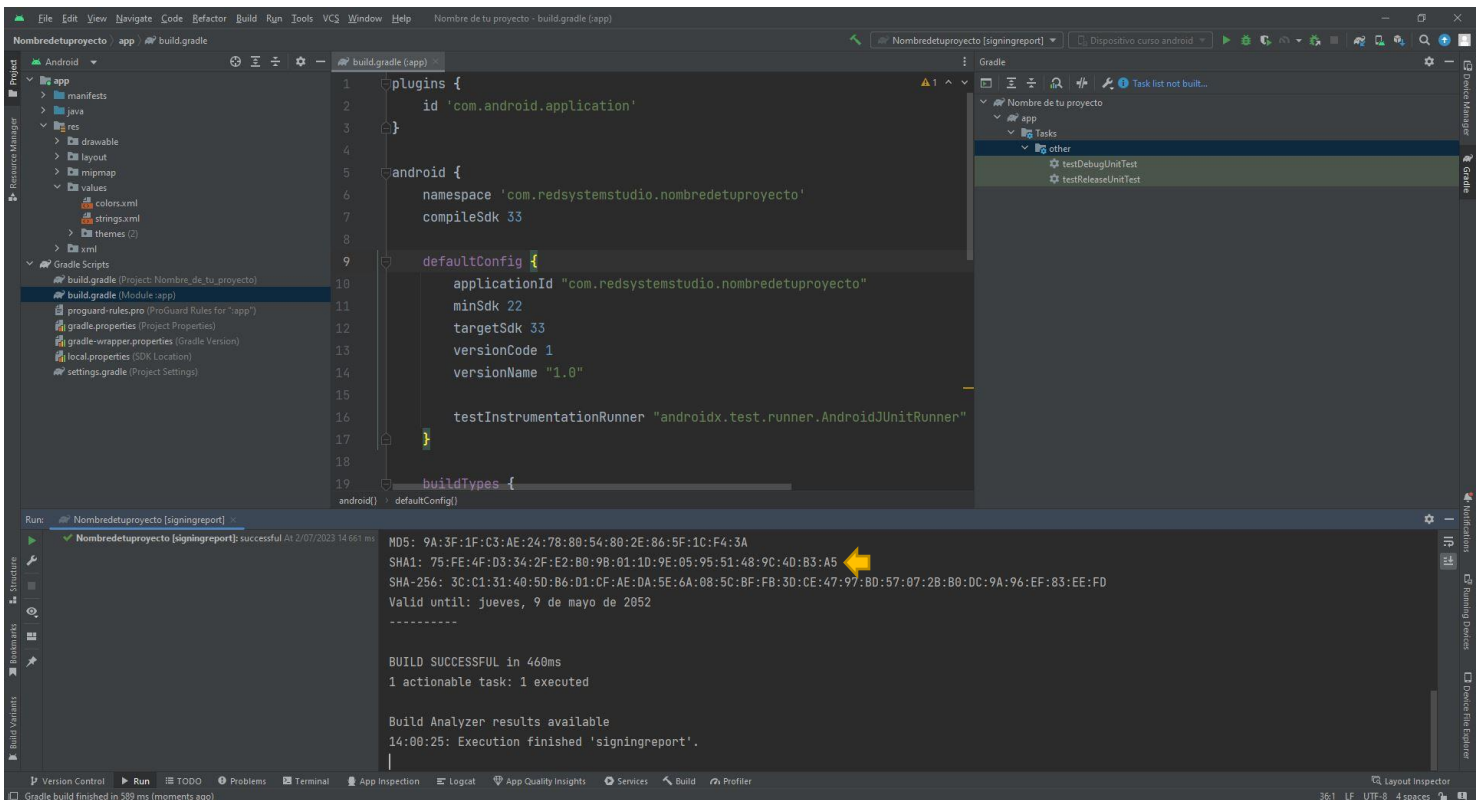




[16] Escribimos **signingreport** en la consola y presionamos tecla **Intro**



[17] Aquí localizamos la firma SHA-1, lo copiaremos



## [18] Pegamos la firma y presionamos en **Registrar App**

× **Agrega Firebase a tu app para Android**

**1 Registrar app**

Nombre del paquete de Android ⓘ  
`com.redsystemstudio.nombredetuproyecto`

Sobrenombre de la app (opcional) ⓘ  
`Mi proyecto`

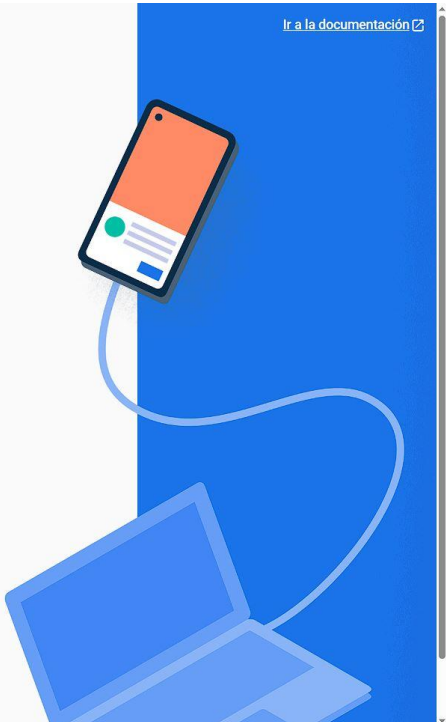
Certificado de firma SHA-1 de depuración (opcional) ⓘ  
`75:FE:4F:D3:34:2F:E2:B0:9B:01:1D:9E:05:95:51:48:9C` ←

● Obligatoria para Dynamic Links y el Acceso con Google o la asistencia con un número de teléfono en Auth. Puedes editar las claves SHA-1 en Configuración.

**Registrar app** ←

**2** Descargar y, luego, agregar el archivo de configuración

**3** Agregar el SDK de Firebase



## [19] Ahora presionamos en **Descargar google-services.json**

× **Agrega Firebase a tu app para Android**


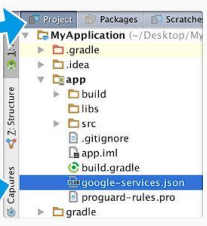
✓ **Registrar app**  
Nombre del paquete de Android: `com.redsystemstudio.nombredetuproyecto`; sobrenombre de la app: `Mi proyecto`

**2 Descargar y, luego, agregar el archivo de configuración** Android Studio a continuación | [Unity](#) | [C++](#)

**Descargar google-services.json** ←

Cambia a la vista **Proyecto** en Android Studio para ver el directorio raíz de tu proyecto.

Mueve el archivo `google-services.json` descargado al directorio raíz de tu módulo (nivel de app).

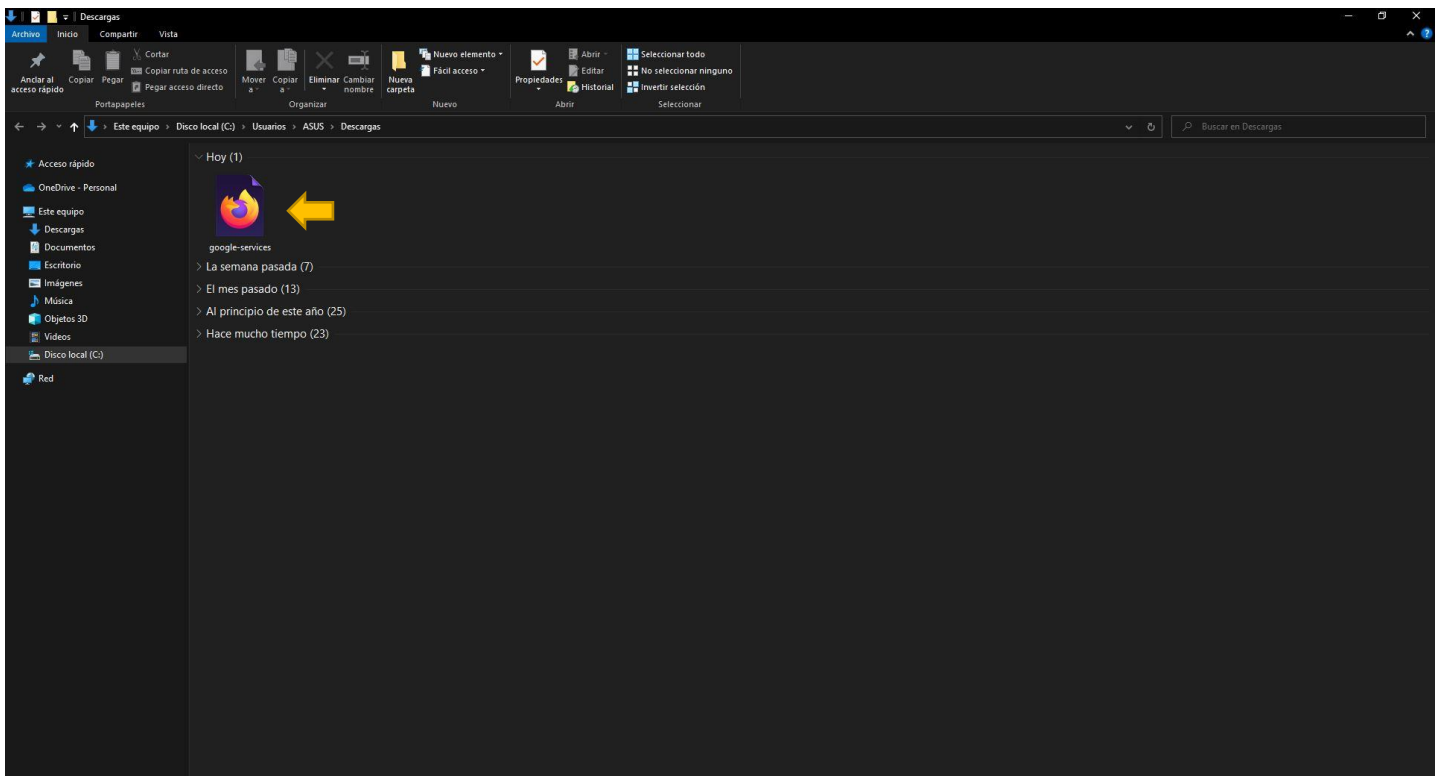


**Siguiente**

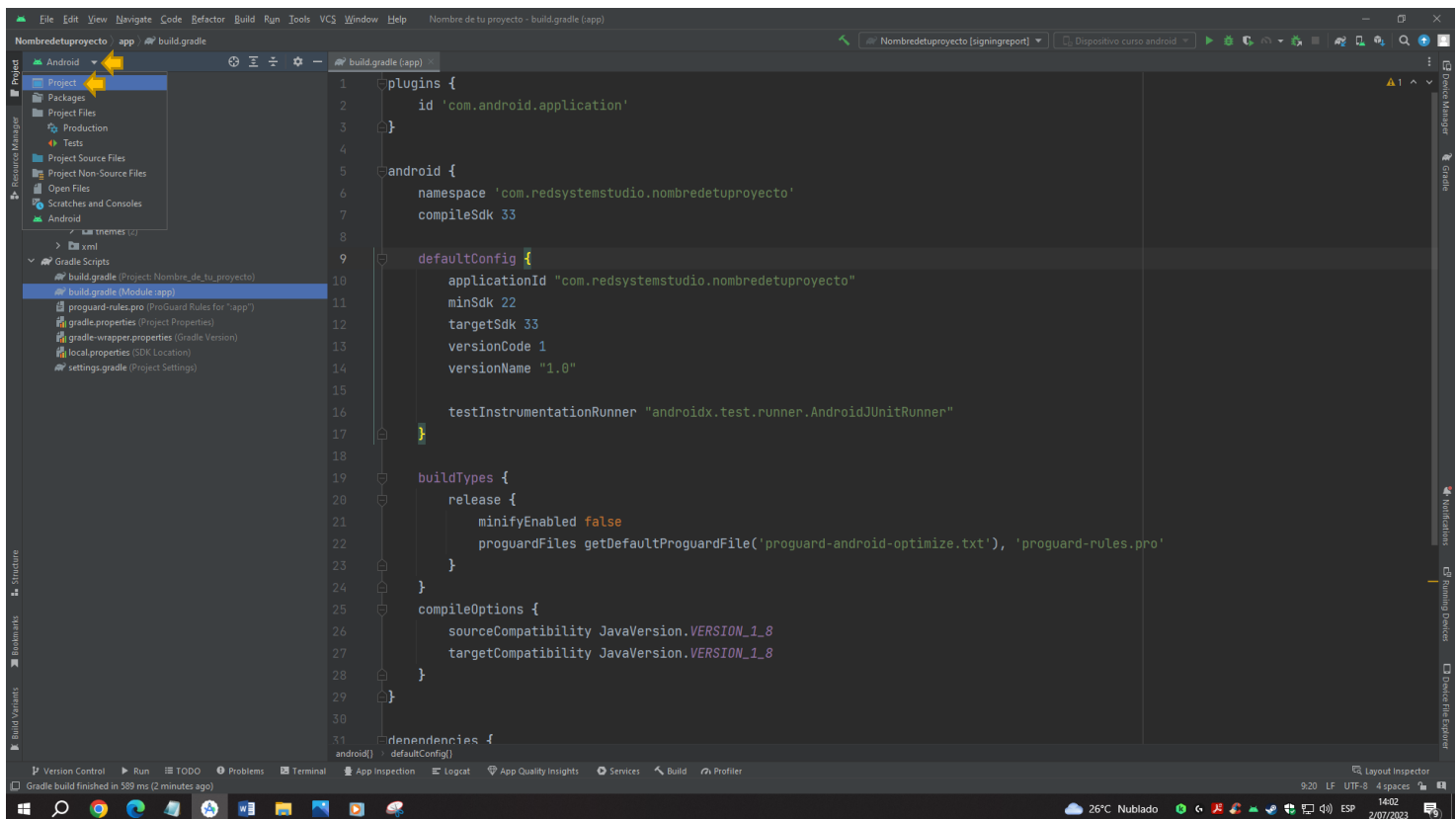
**3** Agregar el SDK de Firebase



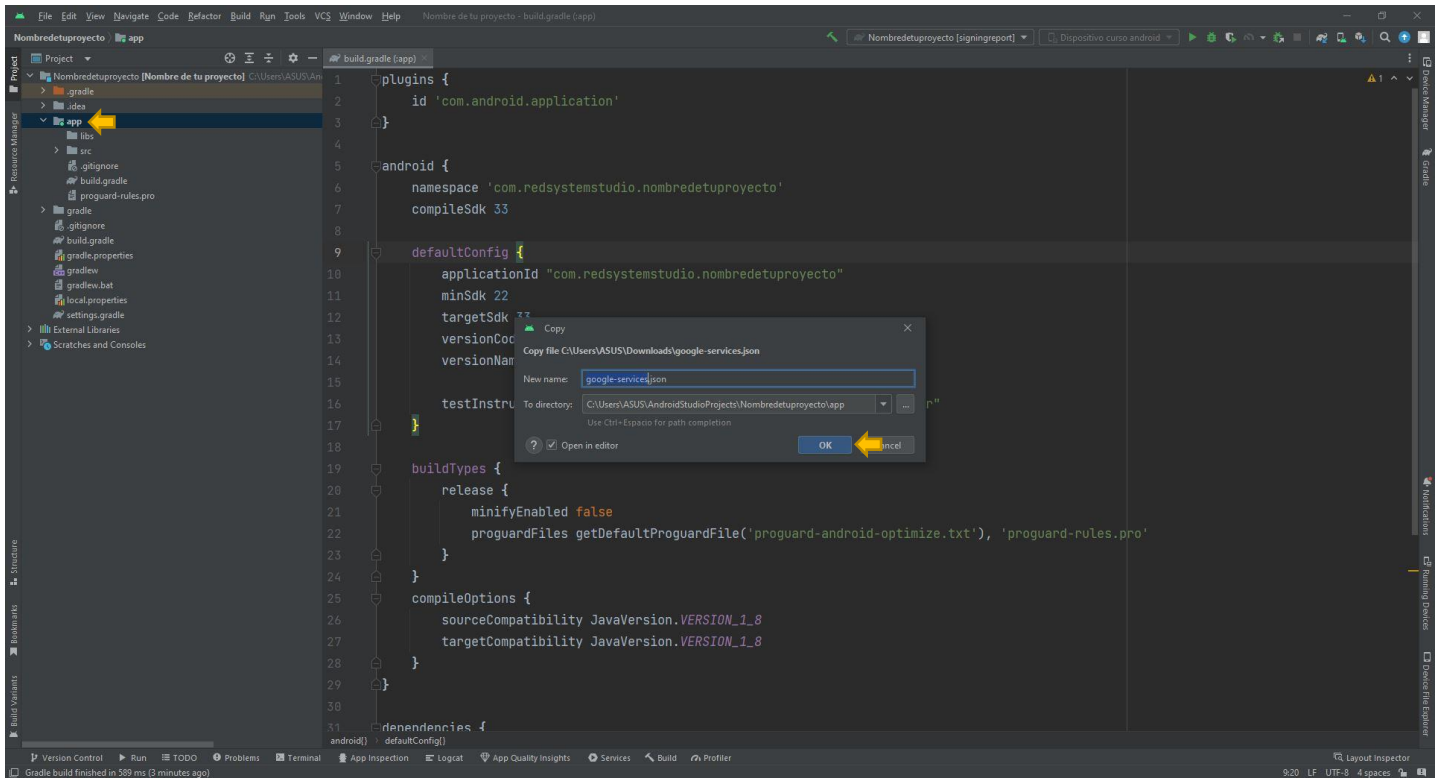
[20] Hemos descargado el archivo en nuestro ordenador, ahora lo copiaremos



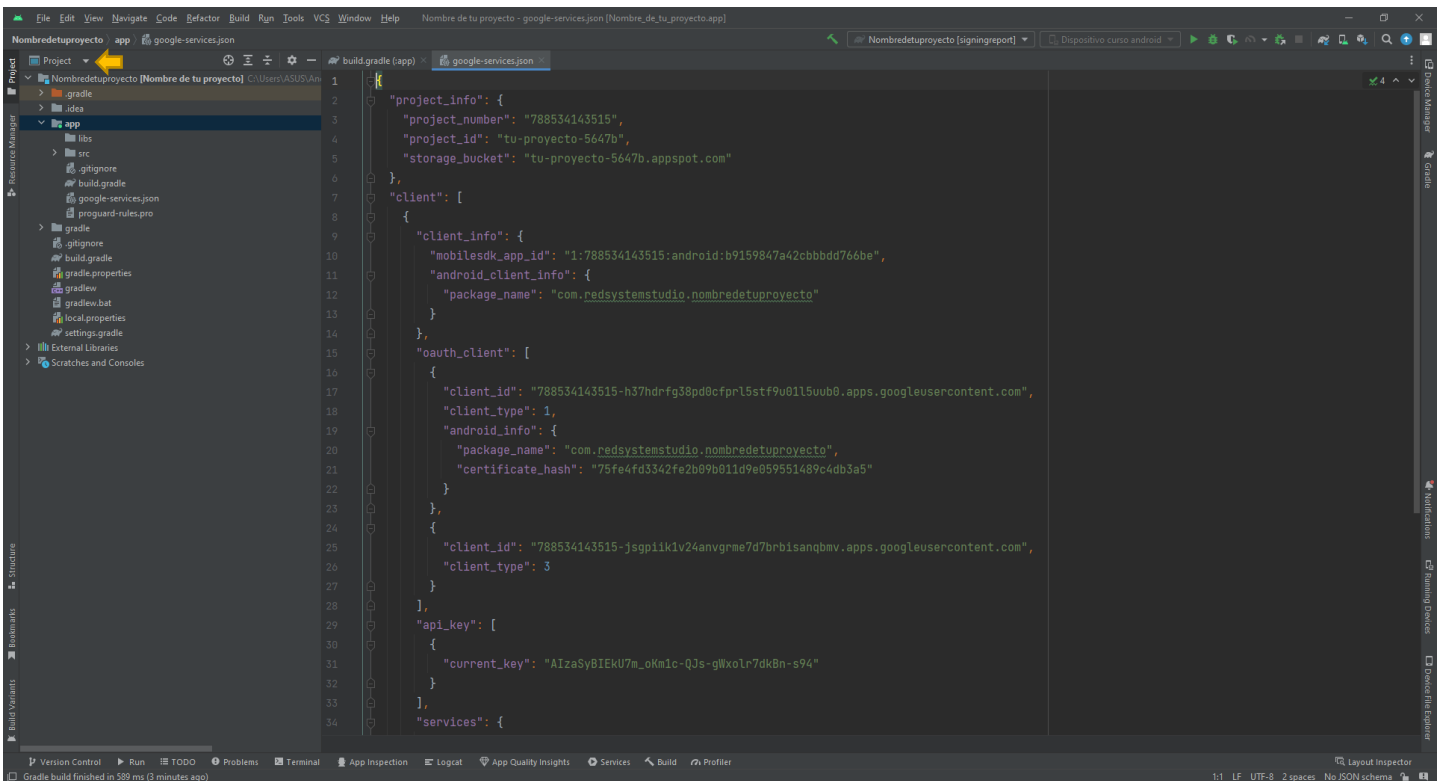
[21] Nos dirigimos a Android Studio y cambiamos la perspectiva de **Android** a **Project**



[22] Pegamos el archivo **google-services.json** en la carpeta **app**, presionamos en **OK**



[23] Cambiamos la perspectiva de **Project** a **Android**



## [24] Regresamos a Firebase y presionamos en **Siguiente**

### × Agrega Firebase a tu app para Android

#### ✓ Registrar app

Nombre del paquete de Android: com.redsystemstudio.nombredetuproyecto; sobrenombre de la app: Mi proyecto

#### 2 Descargar y, luego, agregar el archivo de configuración

Descargar google-services.json

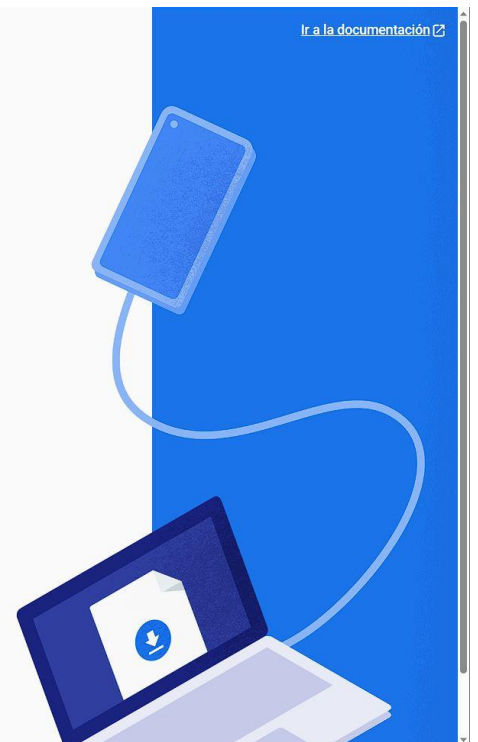
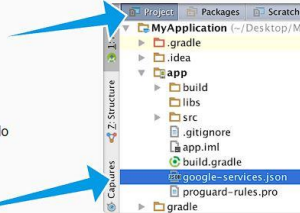
Cambia a la vista Proyecto en Android Studio para ver el directorio raíz de tu proyecto.

Mueve el archivo google-services.json descargado al directorio raíz de tu módulo (nivel de app).



Siguiente

#### 3 Agregar el SDK de Firebase



## [25] Seleccionamos Groovy y copiamos lo que tenemos dentro de **plugins**

### 3 Agregar el SDK de Firebase

Instrucciones para Gradle | Unity | C++

★ ¿Todavía usas la sintaxis **buildscript** para administrar complementos? Aprende a agregar complementos de Firebase con esa sintaxis.

☐ DSL de Kotlin (build.gradle.kts) ☒ Groovy (build.gradle)

1. Para que los SDK de Firebase puedan acceder a los valores de configuración de google-services.json, necesitas el complemento Gradle de los servicios de Google.

Agrega el complemento como una dependencia a tu archivo build.gradle de nivel de proyecto:

Archivo de Gradle de nivel de raíz (nivel de proyecto) (<project>/build.gradle):

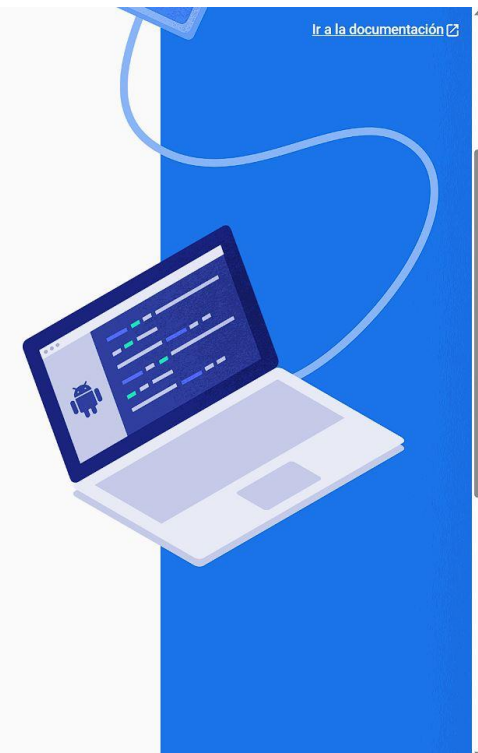
```
plugins {  
    // ...  
  
    // Add the dependency for the Google services Gradle plugin  
    id 'com.google.gms.google-services' version '4.3.15' apply false  
}
```

2. Luego, en el archivo build.gradle del módulo (nivel de app), agrega los complementos google-services y cualquier SDK de Firebase que quieras usar en tu app:

☐ Kotlin ☒ Java

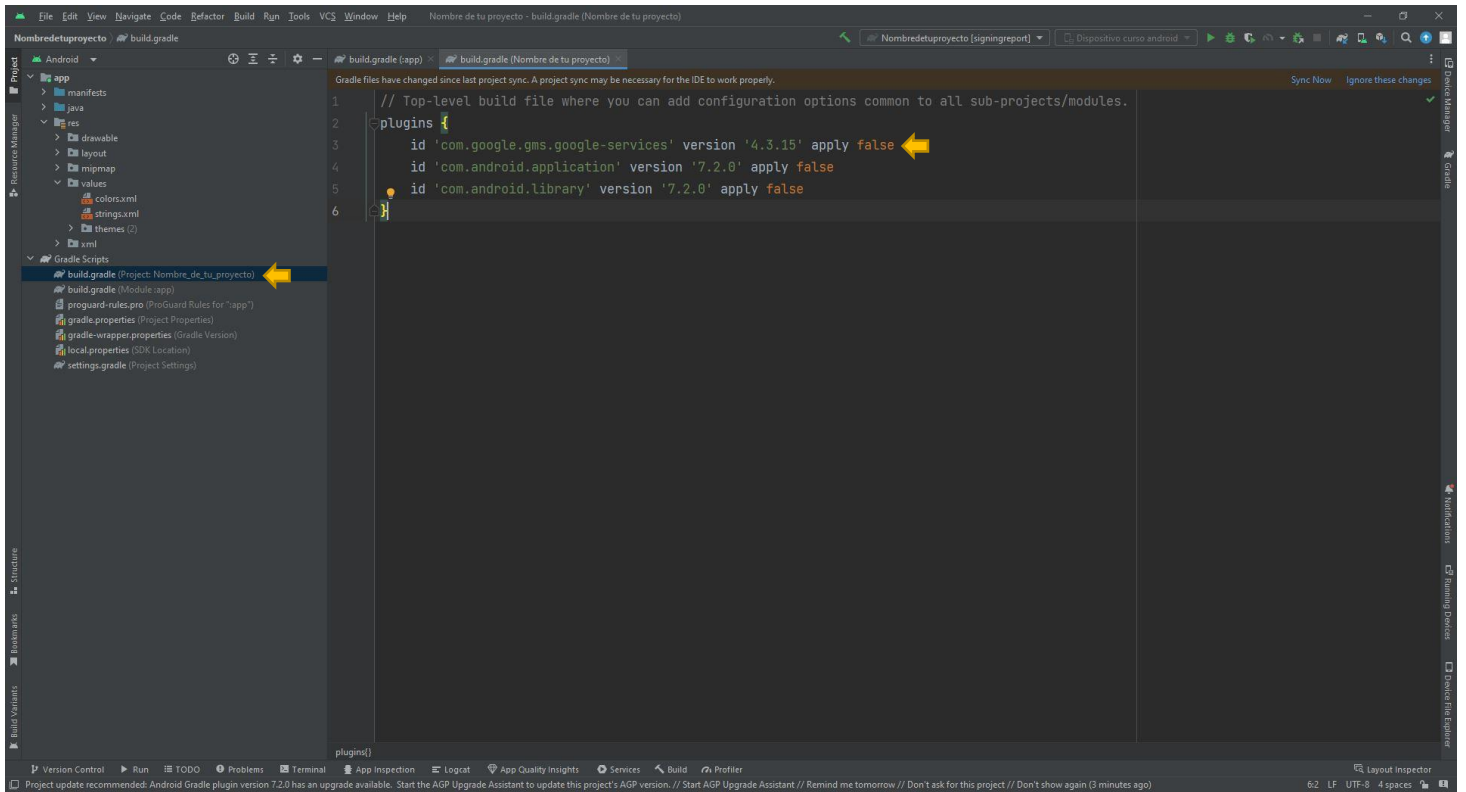
Archivo de Gradle del módulo (nivel de app) (<project>/<app-module>/build.gradle):

```
plugins {  
    id 'com.android.application'  
    // Add the Google services Gradle plugin  
    id 'com.google.gms.google-services'  
    ...  
}
```





## [26] Lo pegamos en **build.gradle** a nivel de proyecto



## [27] Regresamos a Firebase, presionamos en **Java**, copiamos lo que tenemos en **plugins**

2. Luego, en el archivo `build.gradle` del módulo (nivel de app), agrega los complementos `google-services` y cualquier SDK de Firebase que quieras usar en tu app:

☐ Kotlin ☒ Java

Archivo de Gradle del módulo (nivel de app) (<project>/<app-module>/build.gradle):

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:32.1.1')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics'

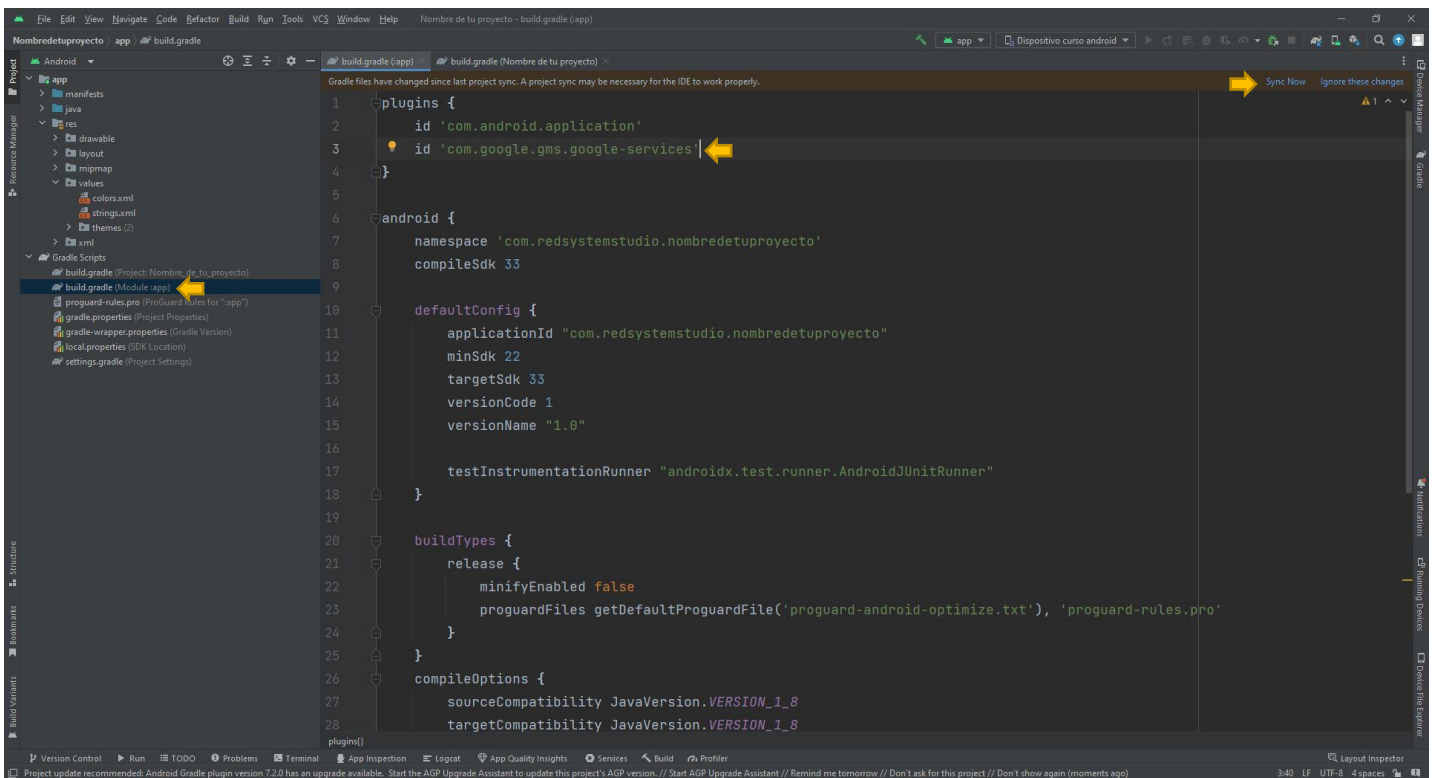
    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

Si usas la BoM de Firebase para Android, tu app siempre utilizará versiones compatibles de la biblioteca de Firebase.  
[Más información](#)

3. Después de agregar el complemento y los SDK deseados, sincroniza tu proyecto de Android con archivos de Gradle.

[Anterior](#) [Siguiente](#)

[28] Nos dirigimos a **build.gradle** a nivel de Módulo, pegamos y presionamos en **Sync Now**



[29] Regresamos a Firebase y presionamos en Siguiente

2. Luego, en el archivo `build.gradle` del módulo (nivel de app), agrega los complementos `google-services` y cualquier SDK de Firebase que quieras usar en tu app:

☐ Kotlin ☒ Java

Archivo de Gradle del módulo (nivel de app) (<project>/<app-module>/build.gradle):

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:32.1.1')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

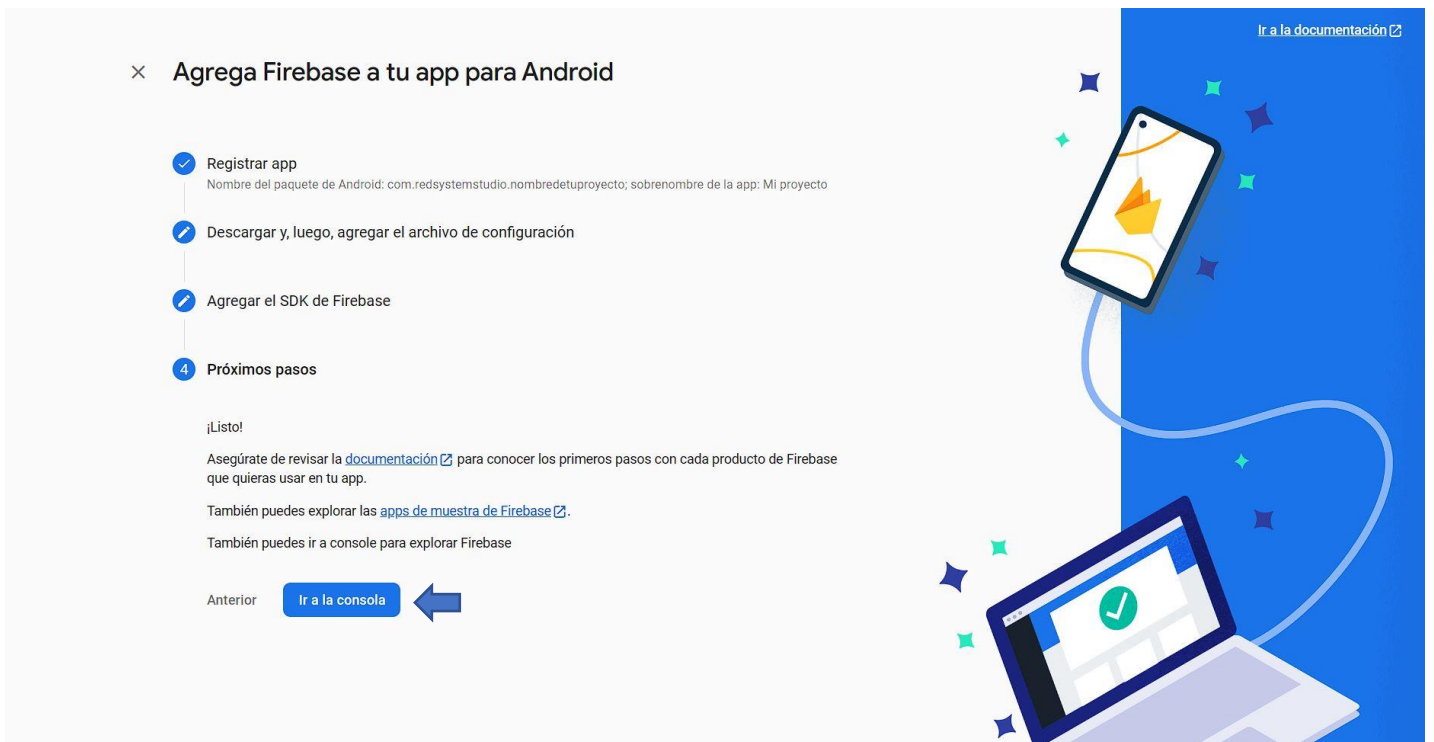
Si usas la BoM de Firebase para Android, tu app siempre utilizará versiones compatibles de la biblioteca de Firebase.  
[Más información](#)

3. Después de agregar el complemento y los SDK deseados, sincroniza tu proyecto de Android con archivos de Gradle.

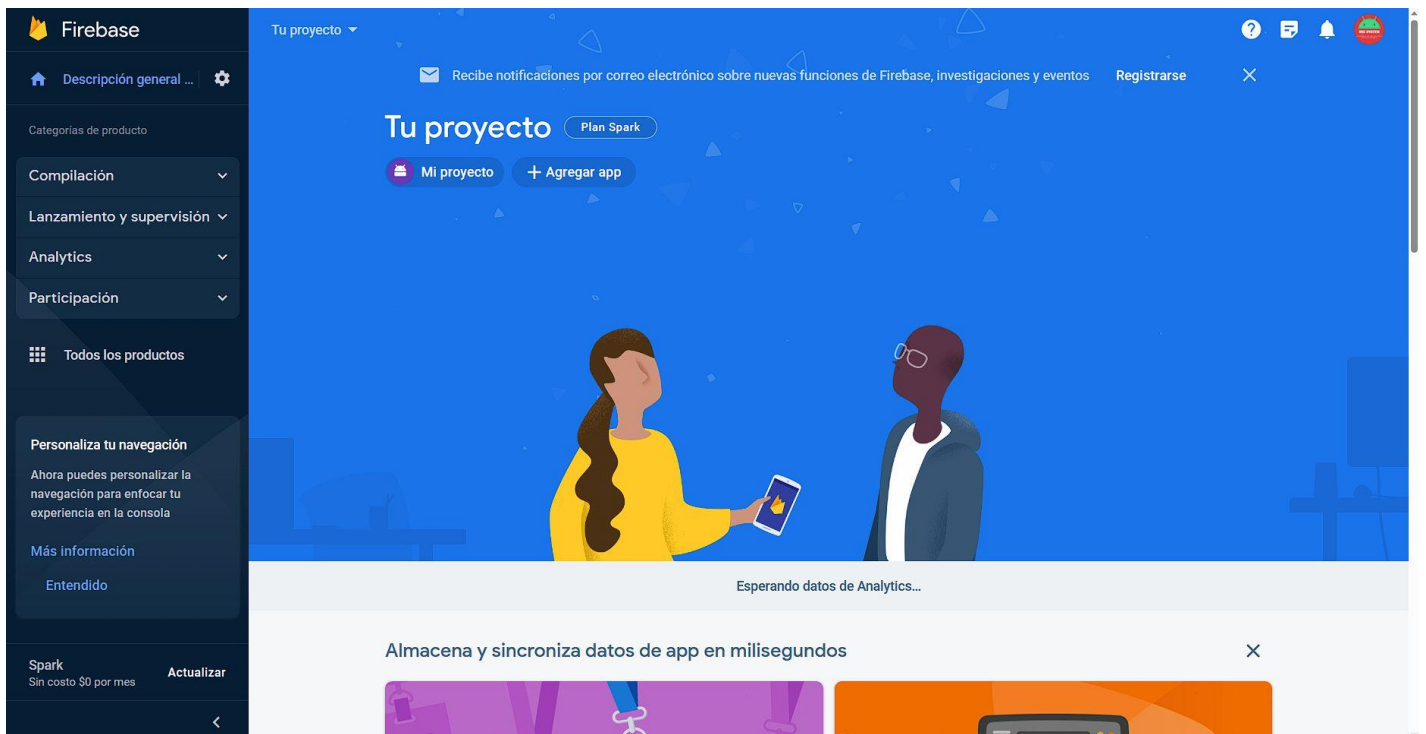
Anterior [Siguiente](#)

[Ir a la documentación](#)

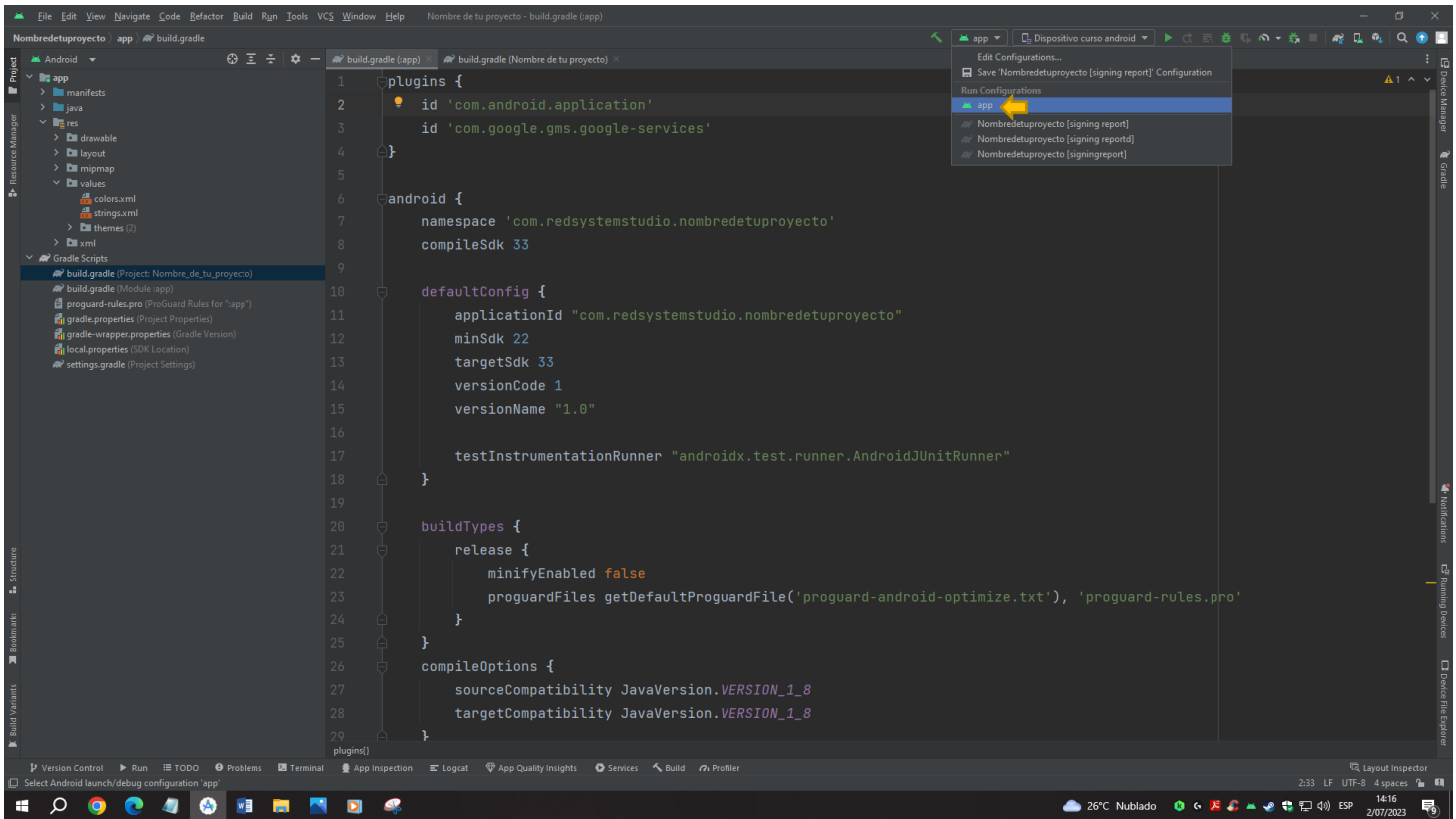
[30] Nos aparecerá la siguiente pantalla, presionamos en **Ir a la consola**



[31] De esta manera hemos integrado Firebase a nuestro proyecto



## [32] Finalmente regresamos a Android Studio y seleccionamos **app**



- Esta es la manera más reciente de integrar Firebase a Android Studio