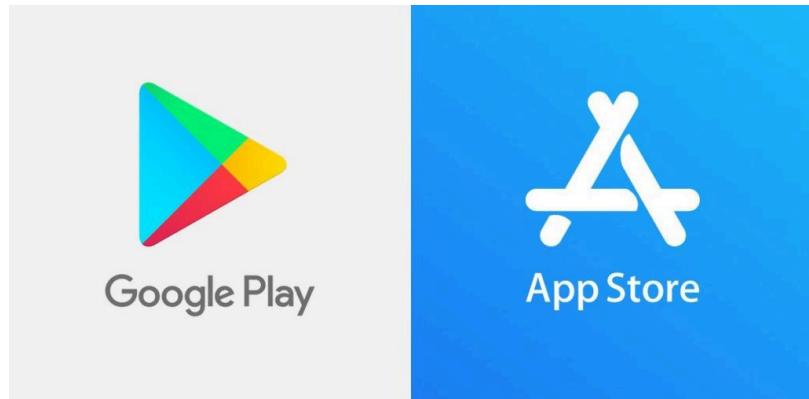


# Flutter

Desplegar aplicaciones en Play Store y App Store.



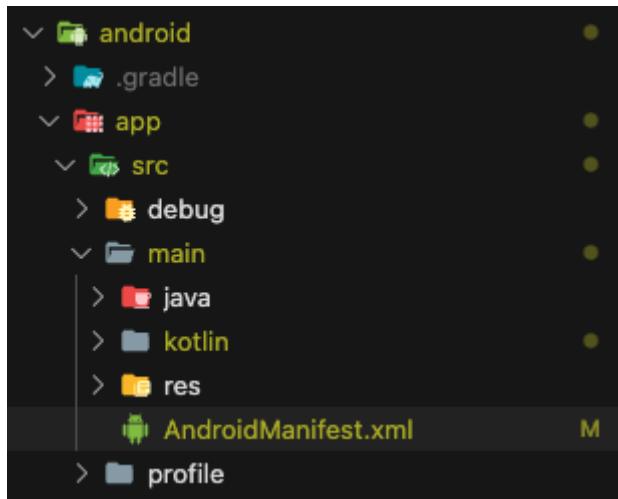
<b>Preparar el despliegue</b>	<b>3</b>
Cambiar el nombre de la Aplicación	3
Android	3
iOS	3
Cambiar el ID de la Aplicación	4
Android	4
iOS	5
Cambiar el icono de la Aplicación	6
Android	6
iOS	8
Cambiar Splash Screen	11
Android	11
iOS	13
Cambiar la versión de la Aplicación	14
Android	14
iOS	15
<b>Firmar Aplicaciones</b>	<b>15</b>
Android	15
Crear keystore	15
Referenciar llave desde la aplicación	15
Configurar la firma en el gradle	16
Generar la versión de producción de la aplicación	17
<b>Publicar Aplicación</b>	<b>18</b>
Publicar en Play Store	18
Prueba Interna	20
Termina de configurar tu aplicación	23
Prueba Cerrada	24
Producción	24
Publicar en App Store	27
Archives	30

# Preparar el despliegue

## Cambiar el nombre de la Aplicación

### Android

Abrimos el AndroidManifest.xml en la siguiente dirección



Dentro de este archivo se puede cambiar información relacionada a los permisos necesarios para el funcionamiento de la aplicación, pero lo que debemos buscar es el **android:label** y modificamos el texto de acuerdo al nombre de nuestra aplicación.

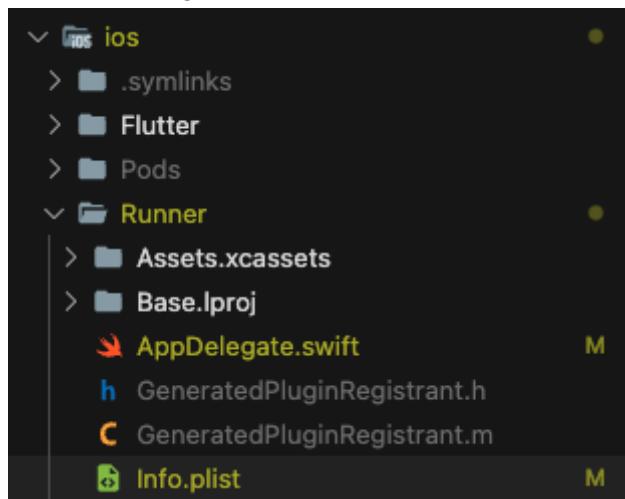
```

<application
    android:label="EELA Jogging"

```

### iOS

Abrimos el archivo Info.plist en la siguiente ubicación



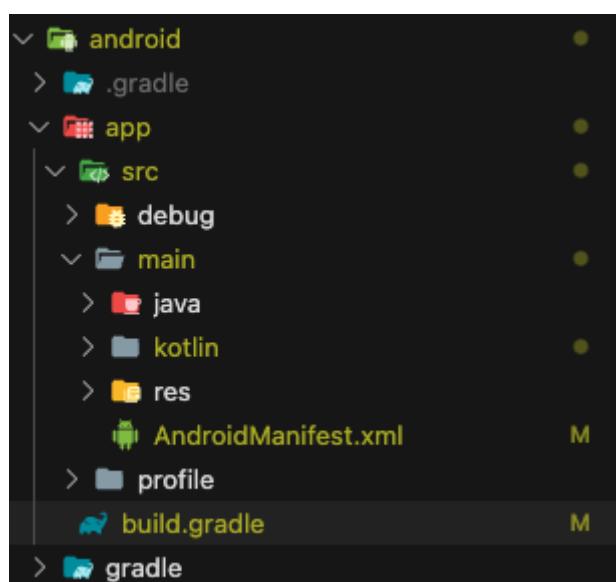
Buscamos la llave **CFBundleDisplayName** y agregamos el nombre de nuestra aplicación.

```
<key>CFBundleDisplayName</key>
<string>EELA Jogging</string>
```

## Cambiar el ID de la Aplicación

### Android

Buscamos y abrimos el archivo **build.gradle**, que se encuentra dentro de la carpeta **Android > app**.

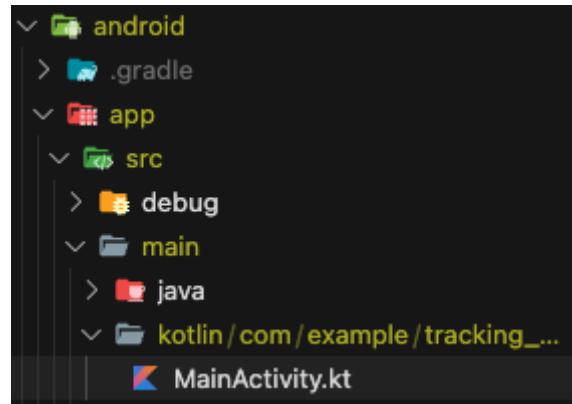


Buscamos **applicationId** y **namespace** y cambiamos el App ID en ambos lugares.

```
android {
    namespace "com.eela.education.jogging.app"
```

```
defaultConfig {
    Create Jira Issue
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
    applicationId "com.eela.education.jogging.app"
```

También en el archivo **MainActivity**, se encuentra el mismo ID, ambos deben de ser iguales, por lo tanto ahí también lo cambiamos



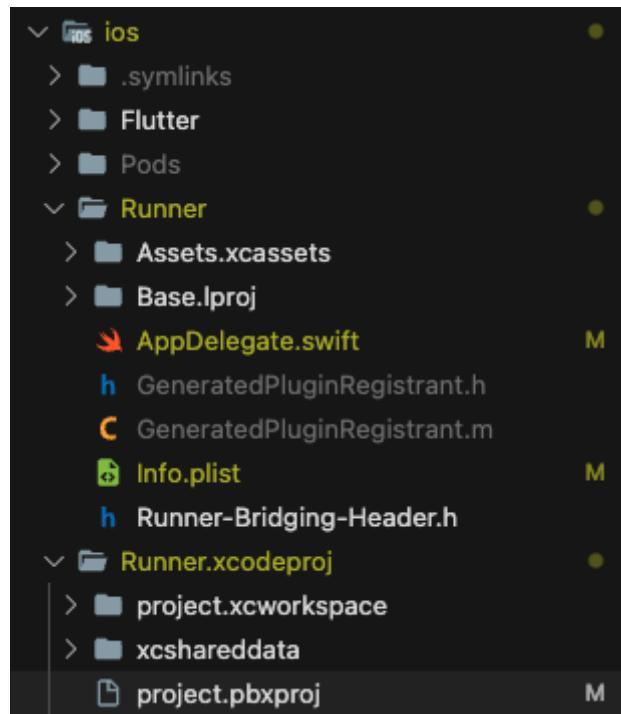
```

1 package com.eela.education.jogging.app
2
3 import io.flutter.embedding.android.FlutterActivity
4
5 class MainActivity: FlutterActivity() {
6 }
7

```

## iOS

Buscamos el archivo **project.pbxproj**



Buscamos todos los label **PRODUCT\_BUNDLE\_IDENTIFIER** y reemplazamos por nuestro App Id.

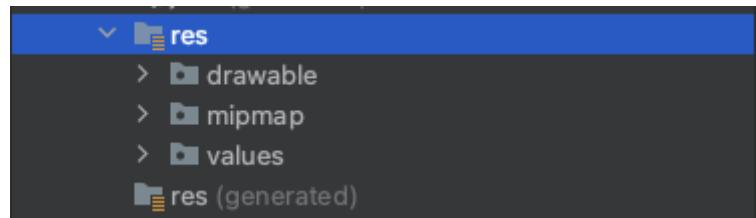


## Cambiar el icono de la Aplicación

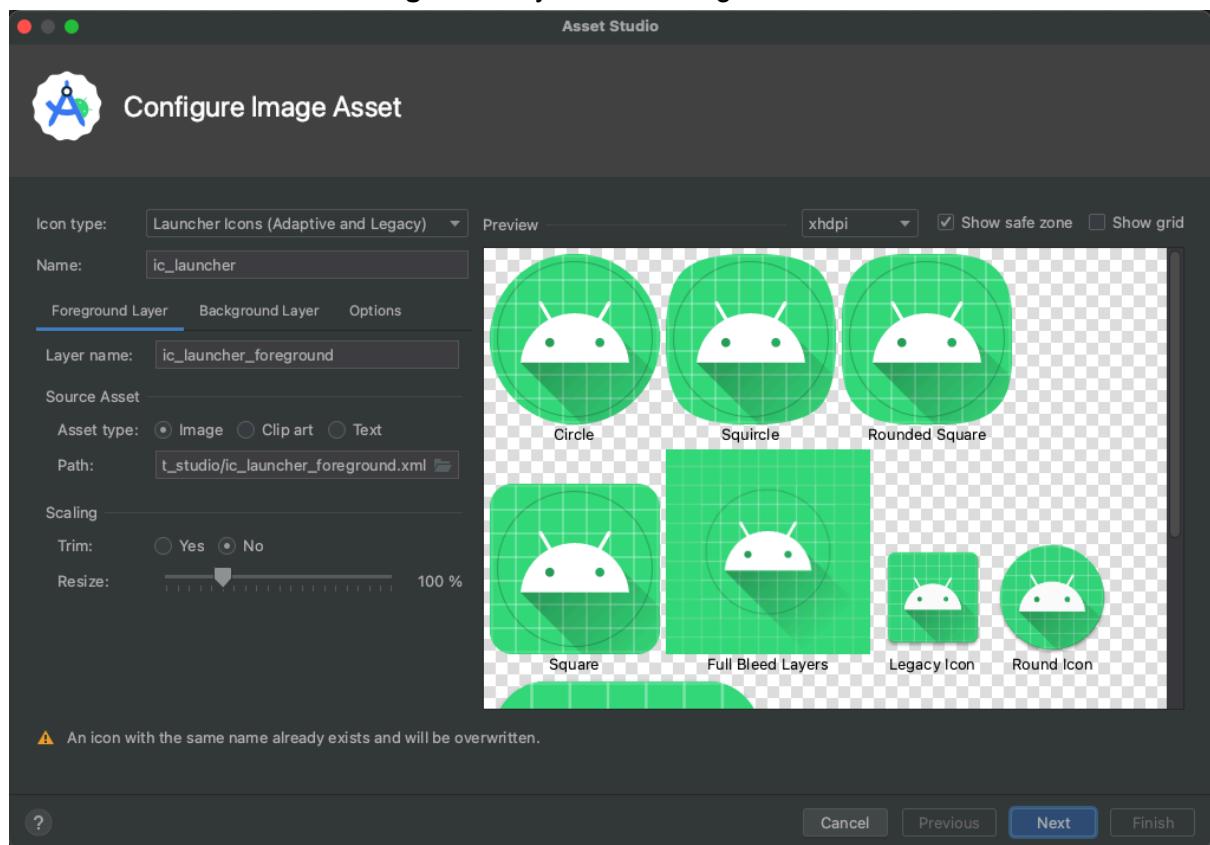
En esta ocasión vamos a crear manualmente los iconos. También se puede usar paquetes que lo realizan de forma automática como por ejemplo: [flutter\\_launcher\\_icons](#)

### Android

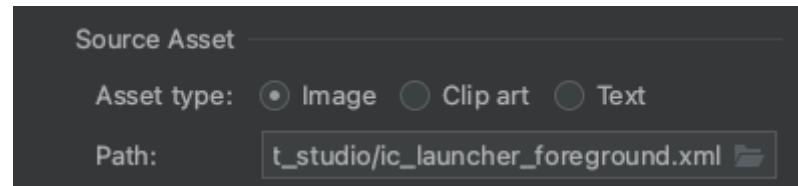
Abrimos Android Studio, en la barra lateral izquierda ubicamos la carpeta de **Android > app > res**.



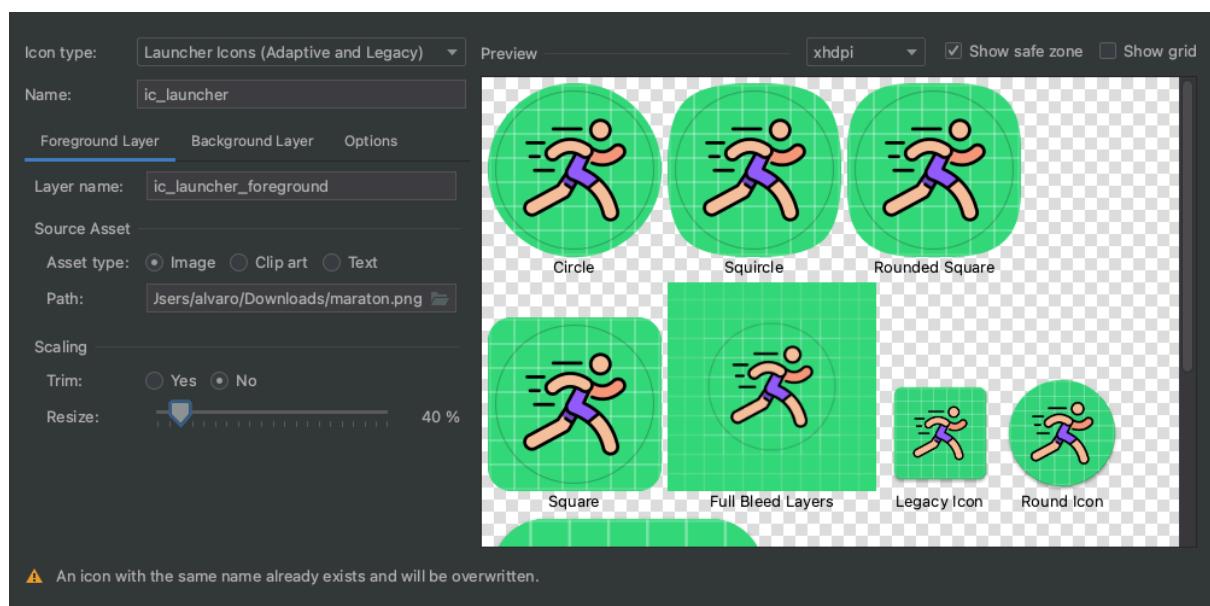
**Click derecho > Nuevo > Image Asset** y se abre la siguiente ventana:



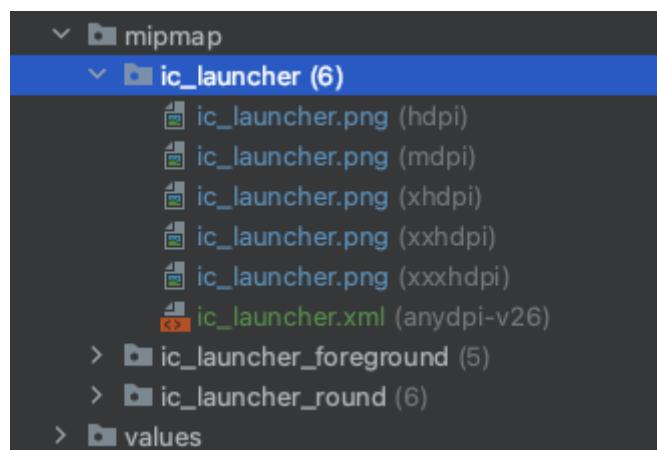
Damos Click en la opción **Path** y elegimos la ubicación de nuestro ícono.



Podemos ajustar el tamaño y cambiar el color de background.



Damos click en **Siguiente** y al finalizar tendremos todos los iconos para los diferentes tamaños de pantalla, algo como esto:



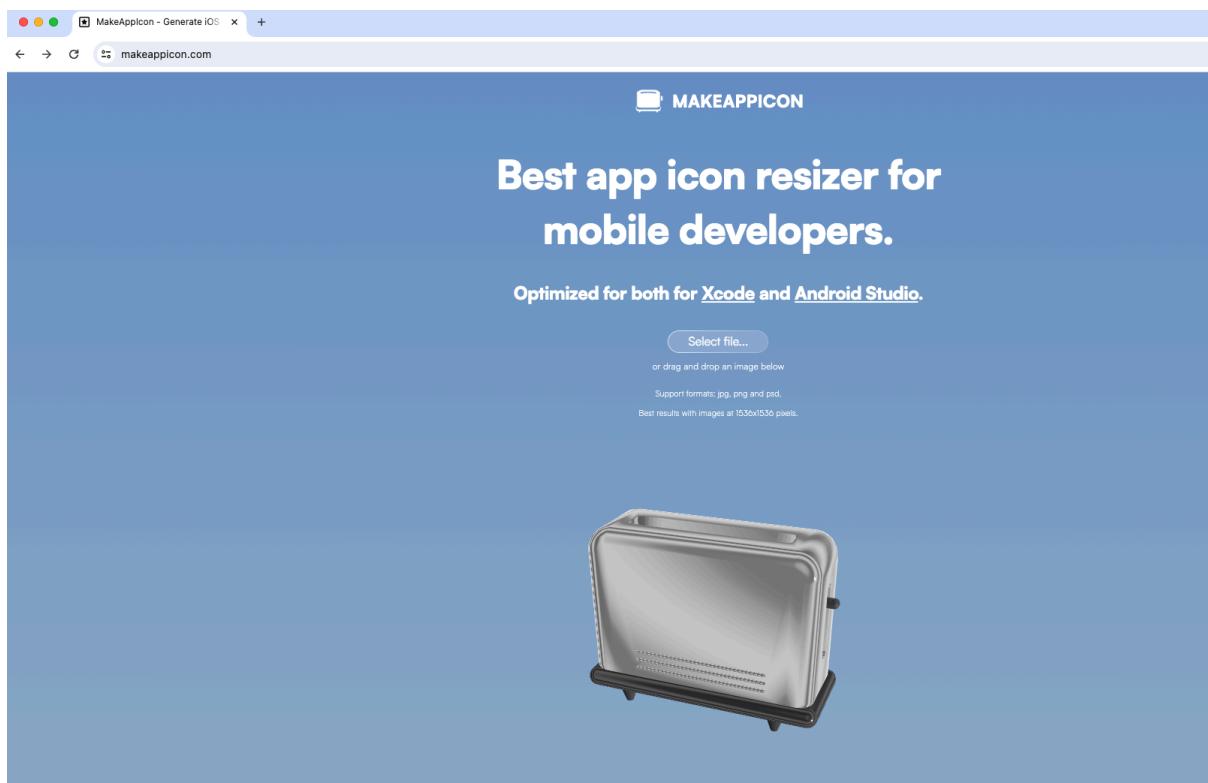
Si lanzamos nuevamente la aplicación, podremos ver el nuevo ícono de nuestra aplicación:



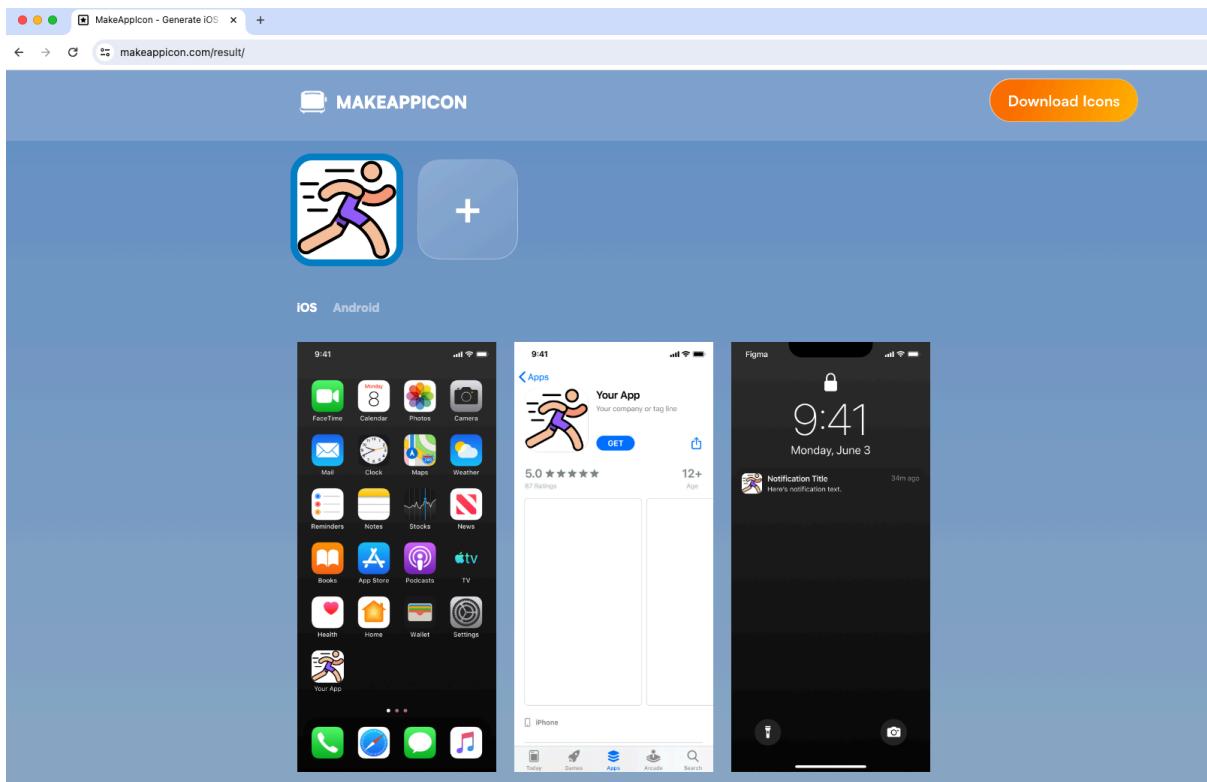
EELA Jo...

## iOS

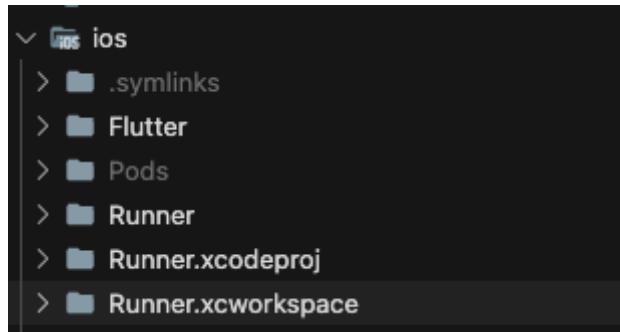
Para iOS vamos a usar la siguiente página: <https://makeappicon.com/>



Arrastramos nuestro ícono a la página. Y elegimos la opción de iOS y luego el botón **Download Icons**. Descomprimimos la carpeta que se descarga.



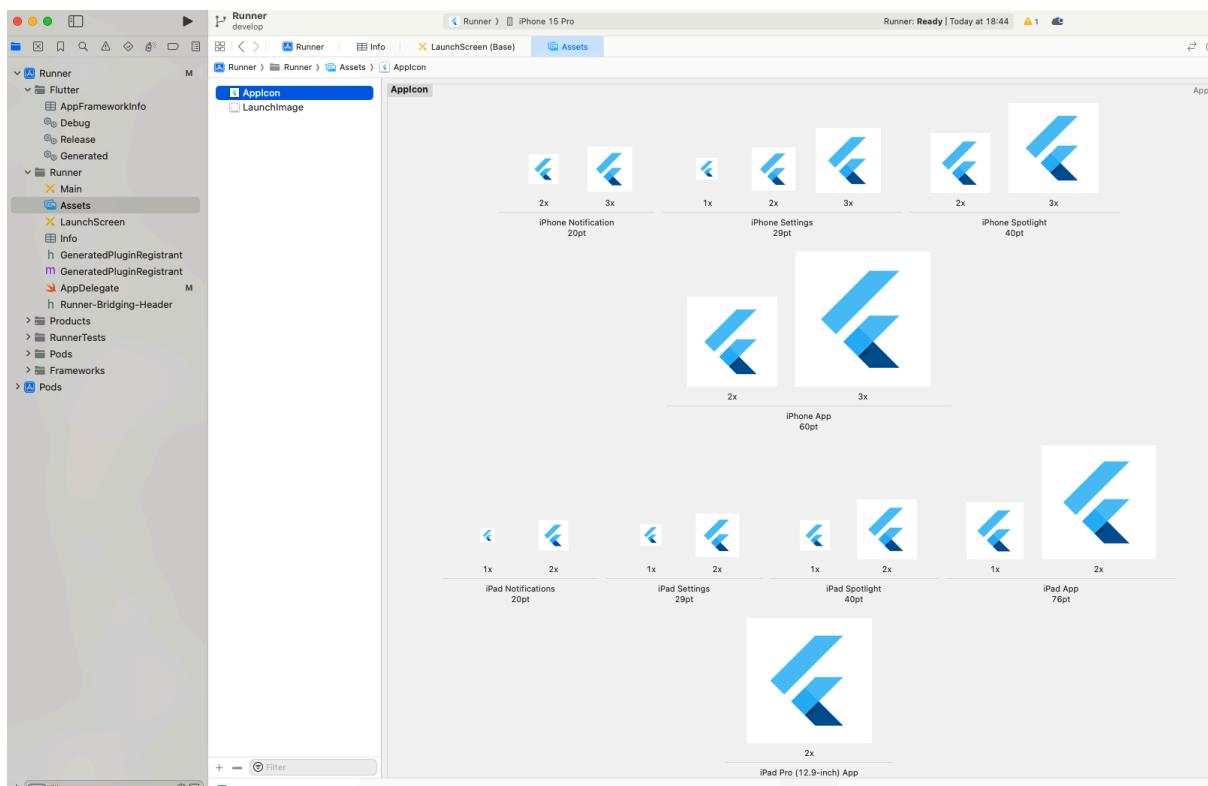
En nuestro proyecto, ubicamos la carpeta **iOS > Runner.xcworkspace > Click Derecho > Show in Finder**.



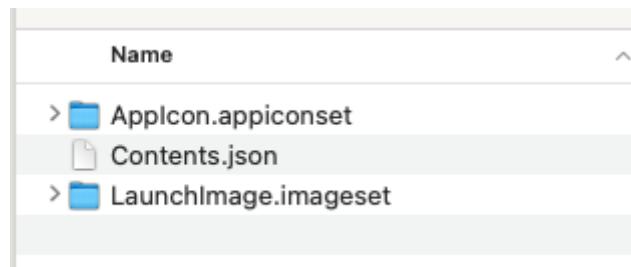
Y abrimos el archivo con Xcode.



En la parte izquierda ubicamos **Runner > Runner > Assets > AppIcon > Click Derecho > Show in Finder**.



Reemplazamos las imágenes de la carpeta **AppIcon.appiconset** con las que descargamos la página **MakeAppIcon**.



Tendremos algo como esto:



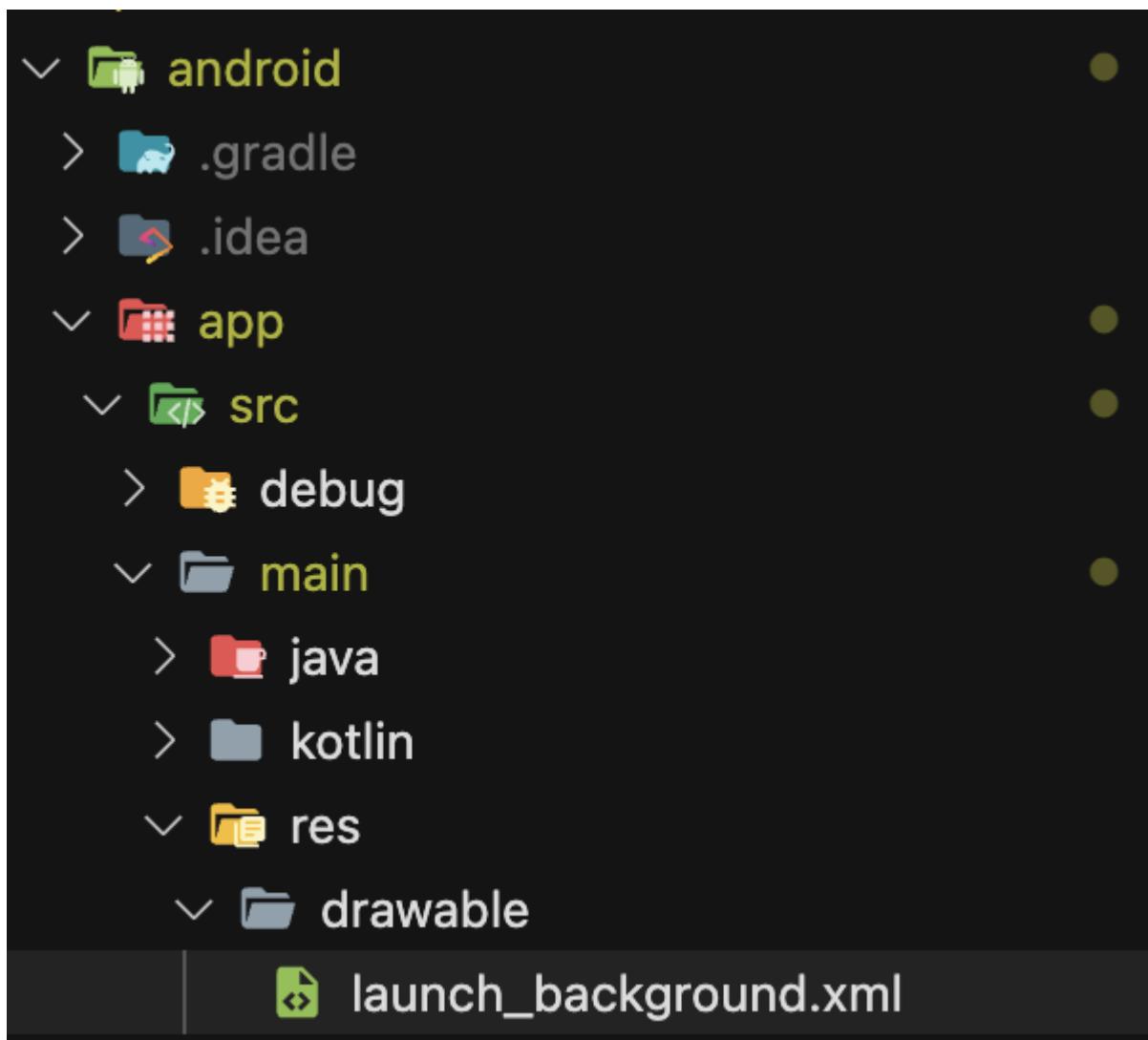
En caso de que tengamos una advertencia en alguna imagen, simplemente le damos click derecho y la eliminamos, ya que es una imagen que no se está usando. Si lanzamos nuevamente la aplicación en iOS, vamos a tener nuestro icono.



## Cambiar Splash Screen

### Android

Abrir el archivo **lunch\_background.xml**



Realizamos la siguiente configuración

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- Modify this file to customize your launch splash screen -->
3  <layer-list xmlns:android="http://schemas.android.com/apk/res/android">
4      <item android:drawable="@android:color/white" />
5
6      <!-- You can insert your own image assets here -->
7      <item>
8          <bitmap
9              android:gravity="center"
10             android:src="@mipmap/ic_launcher" />
11     </item>
12 </layer-list>
13

```

Y en el **launch\_background.xml** (V21)

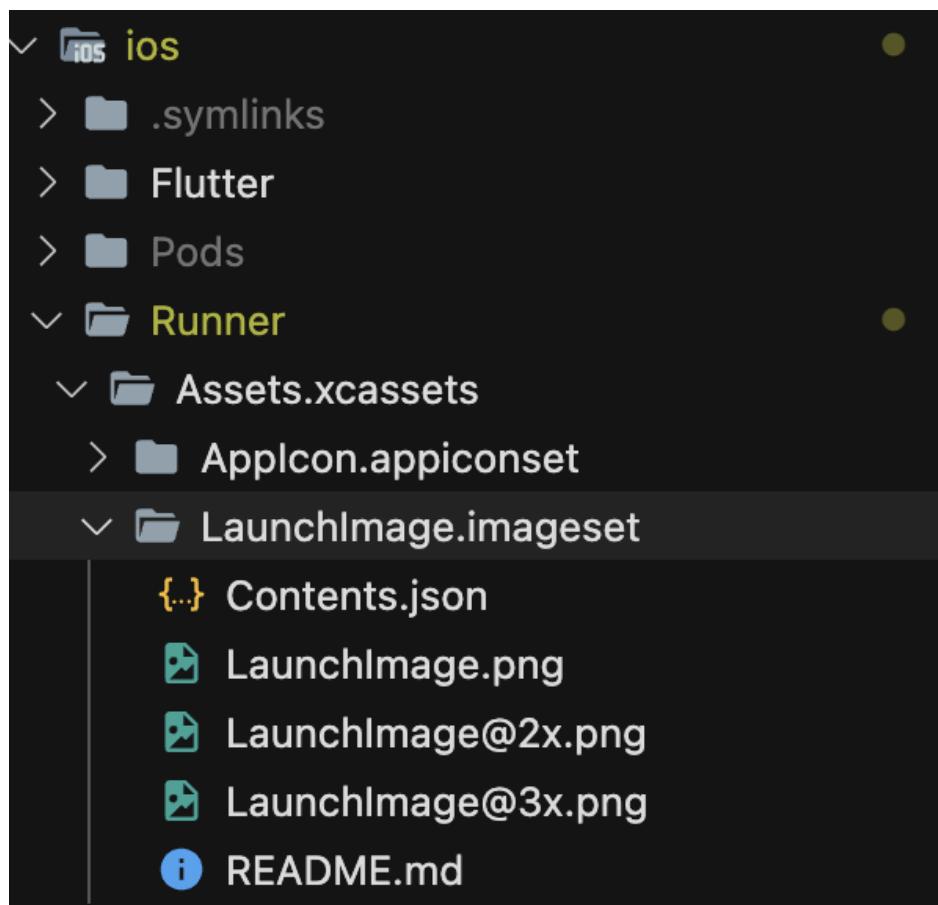
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- Modify this file to customize your launch splash screen -->
3  <layer-list xmlns:android="http://schemas.android.com/apk/res/android">
4    <item android:drawable="?android:colorBackground" />
5
6    <!-- You can insert your own image assets here -->
7    <item>
8      <bitmap
9        android:gravity="center"
10       android:src="@mipmap/ic_launcher" />
11    </item>
12  </layer-list>
13

```

## iOS

Reemplazar estas 3 imágenes por las imágenes que deseas usar en un splash screen, dentro de la carpeta **LaunchImage.imageset** con las imágenes que descargamos de **MakeAppIcon**





## Cambiar la versión de la Aplicación

Esto se puede hacer fácilmente desde el archivo **pubspec.yaml**

```

1 name: tracking_app
2 description: A new Flutter project.
3 # The following line prevents the package from being accidentally published to
4 # pub.dev using `flutter pub publish`. This is preferred for private packages.
5 publish_to: 'none' # Remove this line if you wish to publish to pub.dev
6
7 # The following defines the version and build number for your application.
8 # A version number is three numbers separated by dots, like 1.2.43
9 # followed by an optional build number separated by a +.
10 # Both the version and the builder number may be overridden in flutter
11 # build by specifying --build-name and --build-number, respectively.
12 # In Android, build-name is used as versionName while build-number used as versionCode.
13 # Read more about Android versioning at https://developer.android.com/studio/publish/versioning
14 # In iOS, build-name is used as CFBundleShortVersionString while build-number is used as CFBundleVersion.
15 # Read more about iOS versioning at
16 # https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
17 # In Windows, build-name is used as the major, minor, and patch parts
18 # of the product and file versions while build-number is used as the build suffix.
19 version: 1.0.0+2

```

## Android

<https://developer.android.com/studio/publish/versioning>

En Android, el número **1.0.0** corresponde al **versionName**, es decir es el nombre que se va a ver públicamente en la tienda.

Y el **+2** corresponde al **versionCode**, que le sirve a Play Store para identificar la versión de la aplicación.

*Importante: Cada vez que se quiera desplegar una nueva versión de tu app a la tienda, DEBES incrementar el **versionCode** al menos en 1. El **versionName** puede ser el mismo. por ejemplo, si ahora quiero subir una actualización, tendría que usar el siguiente código:*

```
version: 1.0.0+3
```

## iOS

<https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html>

En iOS, **1.0.0** es usado como **CFBundleShortVersionString** mientras que el número **+2** se usa como **CFBundleVersion**.

## Firmar Aplicaciones

### Android

#### Crear keystore

En macOS o Linux, usar el siguiente comando:

```
keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA \
-keysize 2048 -validity 10000 -alias upload
```

En Windows:

```
keytool -genkey -v -keystore %userprofile%\upload-keystore.jks ^
-storetype JKS -keyalg RSA -keysize 2048 -validity 10000 ^
-alias upload
```

- Este comando pedirá una contraseña y repetir la contraseña, (guarden esta clave, por que cada vez que subamos una actualización de la aplicación la vamos a necesitar)
- Luego pide información personal, no es tan importante, pero deben de colocar sus datos.
- Al finalizar, hay que escribir **yes**
- Confirmar con una nueva contraseña (puede ser la misma de antes) y al finalizar, les dirá donde quedará guardado tu key.jks

#### Referenciar llave desde la aplicación

Primero debemos crear el archivo: **key.properties** en la dirección **<app dir>/android/key.properties**

*Si estamos usando un sistema de control de versiones, este archivo NO DEBE de estar incluido en el repositorio, así que hay que agregarlo al .gitignore:*

Debemos llenar la siguiente información, con los datos y contraseñas que creamos en el paso anterior.

```
storePassword=<password-from-previous-step>
keyPassword=<password-from-previous-step>
keyAlias=upload
storeFile=<keystore-file-location>
```

```
1 storePassword=<password-from-previous-step>
2 keyPassword=<password-from-previous-step>
3 keyAlias=upload
4 storeFile=<keystore-file-location>
5
```

## Configurar la firma en el gradle

Agregar la información del archivo **properties** antes del bloque de **android**

```
def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}

android {
    ...
}

def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()) {
    keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}
```

Buscar el bloque **buildTypes** y **signingConfigs/ release** y reemplazar con la siguiente información:

```
signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
        storePassword keystoreProperties['storePassword']
    }
}
```

```

}
buildTypes {
    release {
        signingConfig signingConfigs.release
    }
}

```

```

signingConfigs {
    release {
        keyAlias keystoreProperties['keyAlias']
        keyPassword keystoreProperties['keyPassword']
        storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
        storePassword keystoreProperties['storePassword']
    }
}

buildTypes {
    release {
        signingConfig signingConfigs.release
    }
}

```

## Generar la versión de producción de la aplicación

Importante: Play Store ahora solo permite subir aplicaciones en formato **App bundle**

Ejecutar el siguiente comando en la raíz de nuestro proyecto:

```
flutter build appbundle
```

```

● → tracking_app git:(develop) ✘ flutter build appbundle --release
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 2392
when building your app.
Running Gradle task 'bundleRelease'...                                         5.1s
✓ Built build/app/outputs/bundle/release/app-release.aab (21.1MB).
○ → tracking_app git:(develop) ✘

```

Podemos encontrar el archivo en la siguiente ubicación:

[project]/build/app/outputs/bundle/release/app.aab.

Con esto estamos listos para publicar nuestra aplicación en Play Store.

# Publicar Aplicación

## Publicar en Play Store

Buscar “play console” en Google y abrir la primera opción o ingresar aquí

<https://play.google.com/intl/es/console/about/>

play console

Cerca de 1.520.000.000 resultados (0,20 segundos)

**Patrocinado**

play.google.com  
https://play.google.com › console › features

**Google Play Console - Get Started Today**

Easily update your app with new features, bug fixes, & enhancements, ensuring a smooth UX. Manage your app effectively. From publishing & distribution to performance monitoring. KPI de Play Console. Análisis de desempeño app. Estrategia de retención.

**Lanzar app alta calidad**

Prueba la estabilidad y desempeño. Recopila las primeras reseñas.

**Monetizar con facilidad**

Aprende tácticas de monetización. Crea el flujo de ingresos del app.

**Webinars de Play Console**

Conservar usuarios ¡Sigue pendiente de las novedades!

**Publicar con confianza**

Publica, administra y distribuye. Lanza tu app exitosamente.

**Google Play**  
https://play.google.com › intl › console › about

**Google Play Console**

Obtén ingresos en más de 190 mercados con acceso a compradores, herramientas de optimización, asistencia fiscal y de cumplimiento, y una experiencia de compra ...

[Google Play Commerce](#) · [Fichas de Play Store...](#) · [Google Play Points](#) · [Suscríbeme](#)

Elegimos la opción de la derecha que dice Ir a la Play Console e ingresamos con nuestra cuenta.



## Lo último para potenciar tu crecimiento en Google Play

Conoce cómo podemos ayudarte a ampliar tu público y aumentar su participación; crear experiencias de alta calidad, y mejorar la monetización y la seguridad en Google Play.

[Mirar ahora](#)



Si es la primera vez que vamos a subir una aplicación, nos aparece una opción para crear nuestra primera aplicación. O sino seleccionamos **Todas las apps > Crear app**.

### Crear tu primera aplicación

Las publicaciones que crees aparecerán en esta página. Para empezar, crea tu primera aplicación.

[Crear aplicación](#)

[Más información](#)



Llenamos la información requerida:

## Crear aplicación

### Detalles de la aplicación

Nombre de la aplicación

Este es el nombre que tendrá tu aplicación en Google Play

0/30

Idioma predeterminado

Inglés (Estados Unidos) – en-US

Aplicación o juego

Puedes cambiar esta opción más tarde en Configuración de la tienda

- Aplicación
- Juego

Gratis o de pago

Puedes editar esta información más tarde en la página de aplicación de pago

- Gratis
- De pago

### Declaraciones

Políticas del Programa para Desarrolladores

- Confirma que la aplicación cumple las Políticas del Programa para Desarrolladores  
La aplicación cumple las [Políticas del Programa para Desarrolladores](#). Consulta estos [consejos sobre cómo crear descripciones de aplicaciones de conformidad con las políticas](#) para conocer los motivos habituales por los que se suspenden las aplicaciones y poder evitar la suspensión. Si tu aplicación o tu ficha de Play Store [cumple los requisitos para enviar información por adelantado](#) al equipo de revisión de aplicaciones de Google Play, [ponte en contacto con nosotros](#) antes de realizar la publicación.

Las políticas de despliegue de aplicaciones se actualizaron en Noviembre de 2023..

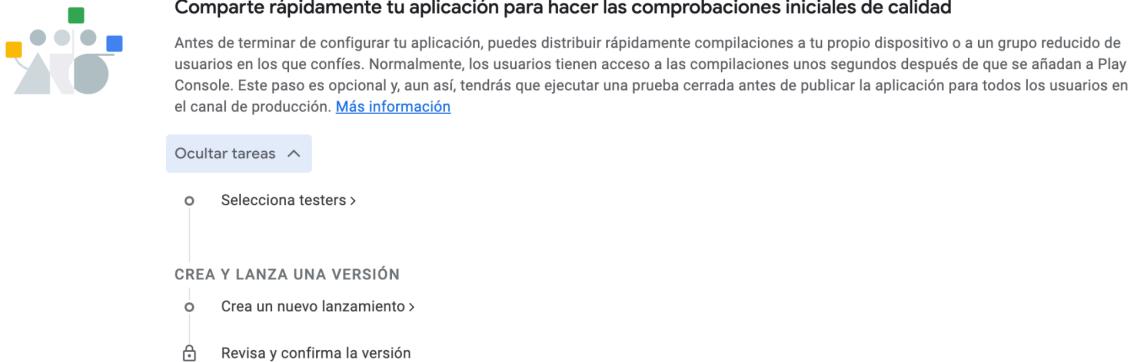
Pueden leer más información aqui

<https://support.google.com/googleplay/android-developer/answer/14151465>

### Prueba Interna

De acuerdo a estas nuevas políticas, primero debemos testear la aplicación de forma interna, antes de lanzarla a producción. Debemos seguir los siguientes pasos:

## Prueba interna



En la parte superior derecha, seleccionamos la opción **Crear nueva versión**

Prueba interna

CREA Y LANZA UNA VERSIÓN

○ Crea un nuevo lanzamiento >

□ Revisa y confirma la versión

Versión Testers

Generamos el AppBundle con el siguiente comando **flutter build appbundle --release**

```
● → tracking_app git:(develop) ✘ flutter build appbundle --release
Font asset "MaterialIcons-Regular.otf" was tree-shaken, reducing it from 1645184 to 2392 bytes (99.9% re-
p.
Running Gradle task 'bundleRelease'...                                                 11.9s
✓ Built build/app/outputs/bundle/release/app-release.aab (21.1MB).
```

Subimos el App bundle.

## Crear versión de prueba interna

Las versiones de pruebas abiertas están disponibles para un máximo de 100 testers que tú eliges

**1** Crear versión — **2** Revisar y confirmar

[Descartar versión](#)

### Integridad de la aplicación

Versiones firmadas por Google Play

Los servicios de Integridad y de firma de Google Play te ayudan a asegurarte de que la experiencia de uso de tus aplicaciones y juegos se corresponda con lo que tienes en mente

[Consigue protección de integridad](#) [Cambiar clave de firma](#)

### App bundles

Arrastra aquí los app bundles que quieras subir

[Subir](#) [Añadir de la biblioteca](#)

### Detalles de la versión

Nombre de la versión *	
------------------------	--

0/50

Solo se usará para que puedas identificar esta versión y no se mostrará a los usuarios en Google Play. El nombre sugerido se basa en el primer app bundle o APK de esta versión, pero puedes cambiarlo.

Luego de subir la aplicación, se llenan los detalles de la versión, donde podemos agregar notas de la versión, aquí es común poner que actualizaciones o funcionalidades tiene la app en cada versión que subamos. Cuando esté lista, damos click en **Siguiente y Guardar**

### Crear versión de prueba interna

Tipo de archivo	Versión	Niveles de API	SDK objetivo	Diseños de pantalla	ABIs	Funciones obligatorias
App bundle	1 (1.0.0)	20 y posterior	33	4	3	2

### Detalles de la versión

Nombre de la versión *	1 (1.0.0)
------------------------	-----------

9/50

Solo se usará para que puedas identificar esta versión y no se mostrará a los usuarios en Google Play. El nombre sugerido se basa en el primer app bundle o APK de esta versión, pero puedes cambiarlo.

Notas de la versión

[Copiar de una versión anterior](#)

```
<es-419>
Introduce o pega aquí las notas de la versión en es-419
</es-419>
```

Has incluido notas de la versión en 0 idiomas

Informa a los usuarios del contenido de tu versión. Introduce notas de la versión dentro de las etiquetas de cada idioma.

Podemos agregar testers con el correo electrónico y compartir el link:

Versiones Testers

### Testers

Hasta 100 testers pueden participar en tus pruebas internas. Puedes elegir más de 100, pero solo los primeros 100 podrán unirse.

**Testers** 

Create an email list to add users that have access to this testing program.

[Crear una lista de correo electrónico](#)

Dirección de correo electrónico o URL para enviar comentarios  Informa a los testers de cómo pueden enviarte comentarios. 0/512

**Cómo se unen los testers a la prueba**

<a href="#">Unirse en la Web</a>	<a href="#">Los testers pueden unirse a tu prueba en la Web</a>
<a href="#">Copiar enlace</a>	

## Termina de configurar tu aplicación

Ahora debemos seguir los siguientes pasos:

### Termina de configurar tu aplicación



Proporciona información de la aplicación y crea una ficha de Play Store

Explicanos qué contenido tiene tu aplicación y gestiona cómo se organiza y presenta en Google Play

Ocultar tareas ^

#### EXPLÍCANOS QUÉ TIPO DE CONTENIDO TIENE TU APLICACIÓN

- Establece la política de privacidad >
- Acceso a las aplicaciones >
- Anuncios >
- Clasificación de contenido >
- Audiencia objetivo >
- Aplicaciones de noticias >
- Aplicaciones de rastreo de contactos y de estado de COVID-19 >
- Seguridad de los datos >
- Aplicaciones gubernamentales >
- Funciones financieras >

#### GESTIONA CÓMO SE ORGANIZA Y SE MUESTRA TU APLICACIÓN

- Selecciona una categoría de la aplicación y proporciona datos de contacto >
- Configura la Ficha de Play Store >

Para generar una página con las Políticas de Privacidad Gratis, puden usar esta plataforma  
<https://app.flycricket.com/>

## Prueba Cerrada

Seguir los siguientes pasos:

### Prueba cerrada



#### Identifica problemas en tu aplicación, recibe comentarios y obtén acceso a producción

Las pruebas cerradas te permiten compartir tu aplicación con un amplio grupo de usuarios que puedes controlar. De esta forma, puedes identificar problemas, recibir comentarios y asegurarte de que tu aplicación esté lista antes de lanzarla. Debes ejecutar una prueba cerrada antes de solicitar que se publique tu aplicación para todo el mundo en el canal de producción. [Más información](#)

Ocultar tareas ^

#### CONFIGURA EL CANAL DE PRUEBAS CERRADO

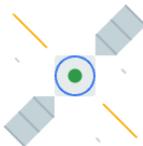
- Selecciona países y regiones >
- Selecciona testers >

#### CREA Y LANZA UNA VERSIÓN

- Crea un nuevo lanzamiento >
- Revisa y confirma la versión
- Envía la versión a Google para que se revise

Una vez que completamos todos los requisitos, podemos lanzar la app a producción

## Producción



### Solicitar acceso a producción

El canal de producción es donde pones tu aplicación a disposición de miles de millones de usuarios en Google Play. Para poder solicitar acceso a producción, debes realizar una prueba cerrada que cumpla nuestros criterios. Al enviar tu solicitud, también deberás responder a algunas preguntas sobre la prueba cerrada. [Vista previa de las preguntas](#)

- Publicar una versión de prueba cerrada
- Tener al menos 20 testers que hayan aceptado participar en tu prueba cerrada (por el momento, han aceptado participar 0 testers)
- Ejecutar la prueba cerrada con 20 testers como mínimo y durante al menos 14 días

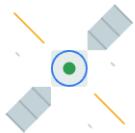
[Solicitar acceso a producción](#)

[Más información](#)

## Producción

Para poder lanzar nuestra app a producción y que esté disponible para cualquier usuario, debemos esperar a que se publique la versión de prueba cerrada, y tener al menos 20 testers durante al menos 14 días .

## Producción



### Solicitar acceso a producción

El canal de producción es donde pones tu aplicación a disposición de miles de millones de usuarios en Google Play. Para poder solicitar acceso a producción, debes realizar una prueba cerrada que cumpla nuestros criterios. Al enviar tu solicitud, también deberás responder a algunas preguntas sobre la prueba cerrada.

- ✓ Publicar una versión de prueba cerrada
- ✓ Tener al menos 20 testers que hayan aceptado participar en tu prueba cerrada
- ✓ Ejecutar la prueba cerrada con 20 testers como mínimo y durante al menos 14 días

[Solicitar acceso a producción](#)

[Más información](#)

Esto es algo que solo el titular de la cuenta puede hacerlo. En caso de que alguien más no haya agregado como miembro de su equipo. Si la cuenta es propia, podemos solicitar el acceso a producción sin problema.



### Solicitar acceso a producción

El canal de producción es donde pones tu aplicación a disposición de miles de millones de usuarios en Google Play. Para poder solicitar acceso a producción, debes realizar una prueba cerrada que cumpla nuestros criterios. Al enviar tu solicitud, también deberás responder a algunas preguntas sobre la prueba cerrada.

🕒 Has solicitado acceso a producción

Estamos revisando tu formulario de solicitud. Envaremos un correo al propietario de la cuenta con las novedades. Este proceso suele tardar 7 días o menos, pero en ocasiones puede llevar más tiempo.

Solicitud enviada el hoy, 9:38.

Cuando acepten los cambios, recibiremos una notificación en la bandeja de entrada de la Play Store.

## Bandeja de entrada

Todos tus mensajes de Google Play en un solo lugar. [Mostrar más](#)

[Recibe notificaciones por correo electrónico](#)

[Mensajes](#) [Mensajes archivados](#)

[EELA Jogging](#) X Añadir filtro X

ⓘ Se han publicado las actualizaciones recientes de tu aplicación

Los usuarios deberían ver estos cambios inmediatamente, pero a veces esto puede tardar más

Archivar

EELA Jogging | 13 feb 2024 2:10 ^

Y podremos ver nuestra aplicación en la tienda.

# EELA Jogging

EELA Institute

5+ Descargas

Para todos

[Instalar en más dispositivos](#)

[Compartir](#)

Esta aplicación está disponible para tu dispositivo



Registra estadísticas de tu entrenamiento



Sigue la ruta hasta un punto específico



Asistencia de la aplicación ▾

Más apps que podrías probar →



Spotify: música y podcasts  
Spotify AB  
4,7 ★



TikTok Lite  
TikTok Pte. Ltd.  
4,5 ★



Documentos de Google  
Google LLC  
4,2 ★



Microsoft Excel: Spreadsheets  
Microsoft Corporation

## Novedades

¡Presentamos la primera versión de EELA Jogging! Descubre las funciones destacadas:

-Integración de Google Maps

## Publicar en App Store

Para desplegar aplicaciones en la Apple AppStore, es necesario pagar 100 dólares ANUALES, a diferencia de Google PlayStore que el servicio es de 30 dólares de por vida.

Diferentes membresías de Apple aquí:

<https://developer.apple.com/support/compare-memberships/>

Enrolarse al programa de developer de Apple

<https://developer.apple.com/programs/>

**Importante: Sin la membresía no se puede continuar, es necesario también tener una computadora MAC, no es opcional.**

Una vez que tenemos la cuenta de desarrollador de Apple, el siguiente paso es crear la aplicación en App Store Connect. [Consulta aquí la documentación oficial.](#)

Accedemos con appId que creamos la cuenta de desarrollador en

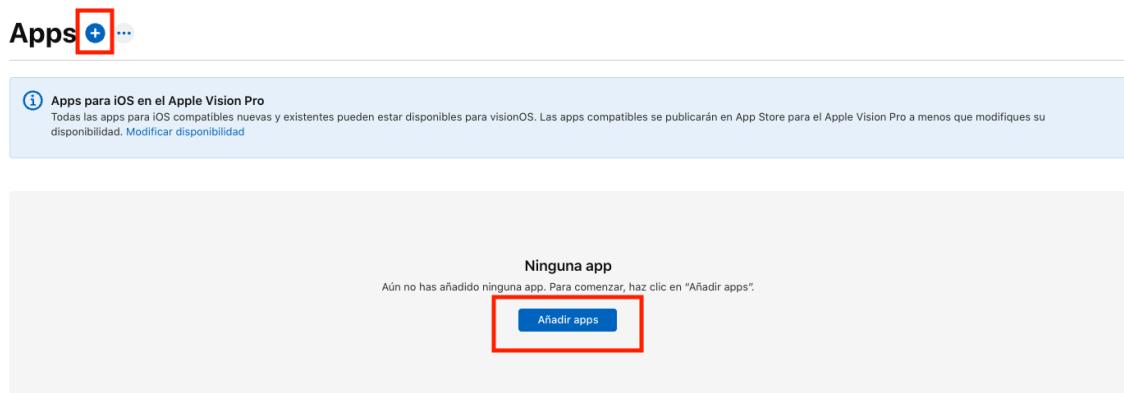
<https://appstoreconnect.apple.com/login>

Si estás trabajando en otra cuenta que no sea la tuya, por ejemplo para una cuenta de alguna organización, debes crear los certificados para las aplicaciones. [Revisa este video antes de continuar.](#)

Seleccionamos la opción **Mis apps**



Si no tenemos ninguna aplicación, podemos agregar una nueva en el botón **Añadir apps** o en el ícono (+) de la parte superior izquierda.



Luego elegimos las plataformas en las que va a estar disponible nuestra aplicación, en este caso solo **iOS**, luego el nombre de la aplicación, el idioma principal, el ID del pack y el SKU (puede ser el mismo ID del pack)

## Nueva app

### Plataformas ?

iOS  macOS  tvOS  visionOS

### Nombre ?

EELA Jogging

18

### Idioma principal ?

Español (México)

▼

### ID de pack ?

Seleccionar

▼

Registra un nuevo ID de pack en [Certificados, identificadores y perfiles](#).

### SKU ?

### Acceso de usuarios ?

Acceso limitado  Acceso ilimitado

[Cancelar](#)

[Crear](#)

Con esto ya podemos empezar a llenar la información de nuestra aplicación:

Es obligatorio subir capturas de pantalla para los dispositivos que se muestran en pantalla. Aquí puedes saber a qué versión de iPhone corresponden los tamaños de pantalla  
<https://developer.apple.com/help/app-store-connect/reference/screenshot-specifications>

Algo importante es que su aplicación tiene algún tipo de inicio de sesión, debes proporcionar cuentas de prueba a los revisores, por que ellos no se van a registrar a su app para probarla. **Esa parte pueden agregarla en la sección Información para el equipo de revisión de apps.** A pesar de que la app de Jogging no tenía un login, si tenía un onboarding donde se pedía información. Entonces para evitar problemas, les agregue la

información que pueden ingresar para ingresar a la aplicación. Incluso si quieren ser mas claros, pueden agregar capturas de pantalla.

#### Información para el equipo de revisión de apps

Información de inicio de sesión ?  
Indica un nombre de usuario y una contraseña para que podamos iniciar sesión en tu app.  
Necesitaremos estos datos para poder completar la revisión de tu app.

Es necesario iniciar sesión

Alvaro Armijos	...	123456789
----------------	-----	-----------

Datos de contacto ?

Alvaro	...	Armijos
+593 989540235		alvaro.armijoss@gmail.com

Notas ?  
La legislación china exige que las apps que incluyan contenido religioso, noticias, libros o revistas tengan los permisos pertinentes para que puedan estar disponibles en App Store de China continental. Si corresponde, introduce la información sobre tus permisos a continuación. [Más información](#)

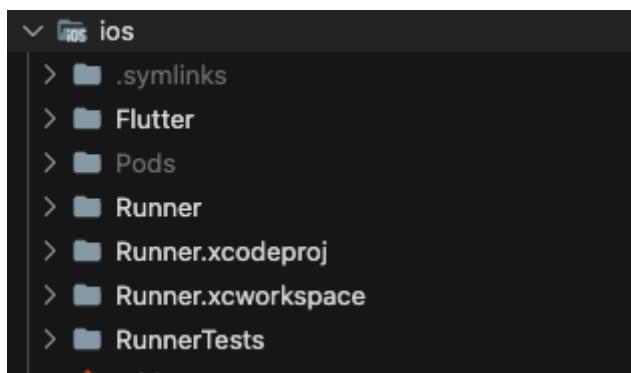
```
name: Alvaro Armijos
email: test@gmail.com
age: 27
weight: 75
```

Archivo adjunto ?

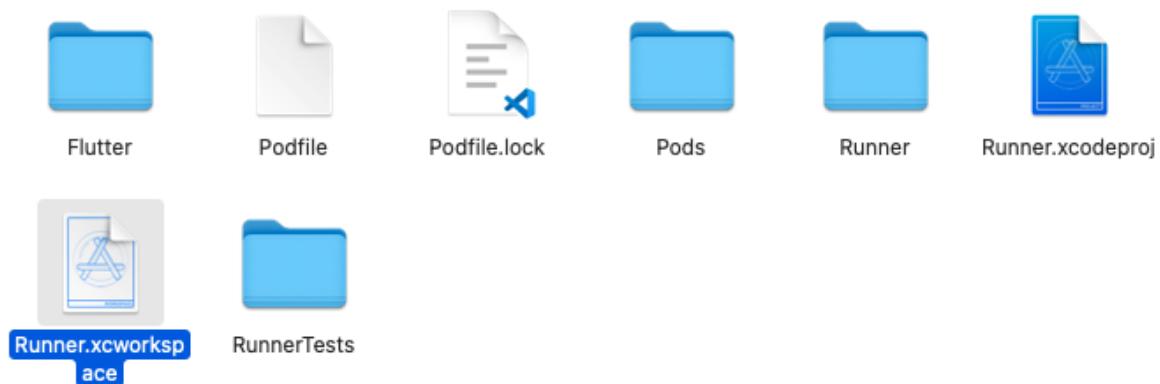
Seleccionar archivo (opcional)

## Archives

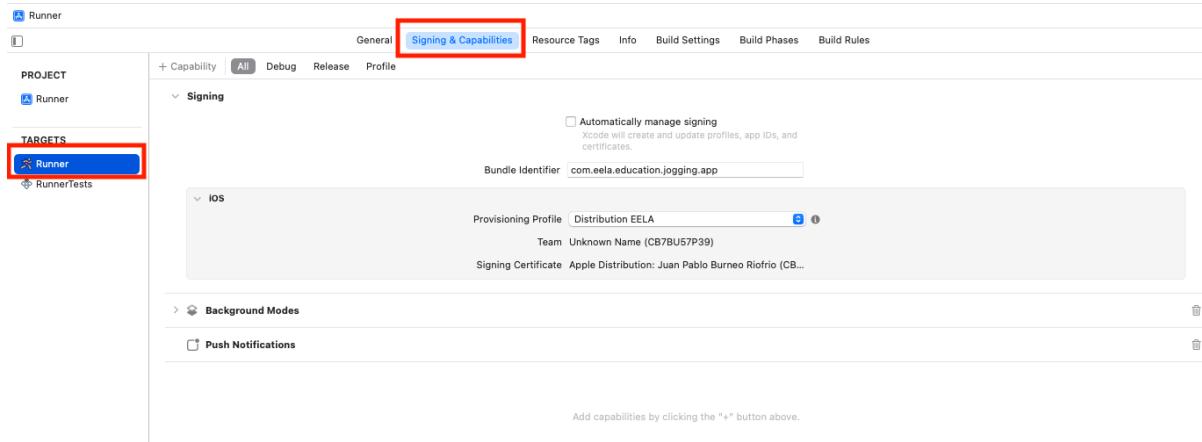
Primero localizamos la carpeta iOS en nuestro proyecto, y la abrimos en el explorador de archivos.



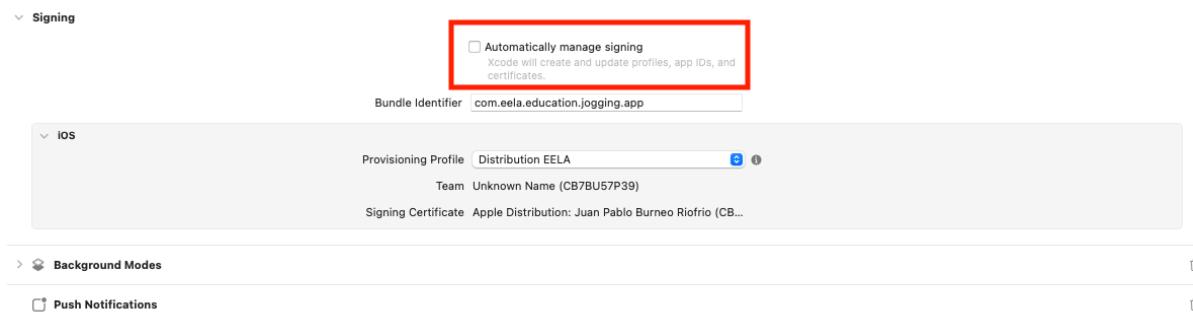
#### Abrimos el archivo Runner.xcworkspace



Primero vamos en la parte izquierda y seleccionamos **Runner**, luego en la parte superior, **Signing & Capabilities**



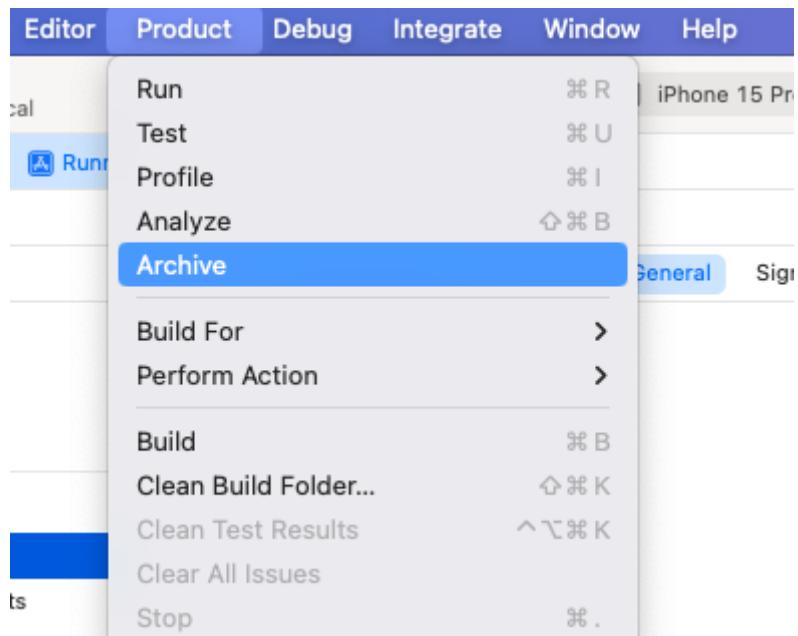
En caso de que la aplicación sea nuestra y estemos subiendo con nuestra cuenta de desarrollador, podemos seleccionar la opción **Automatically manage signing**. Y Xcode se va a encargar de todo



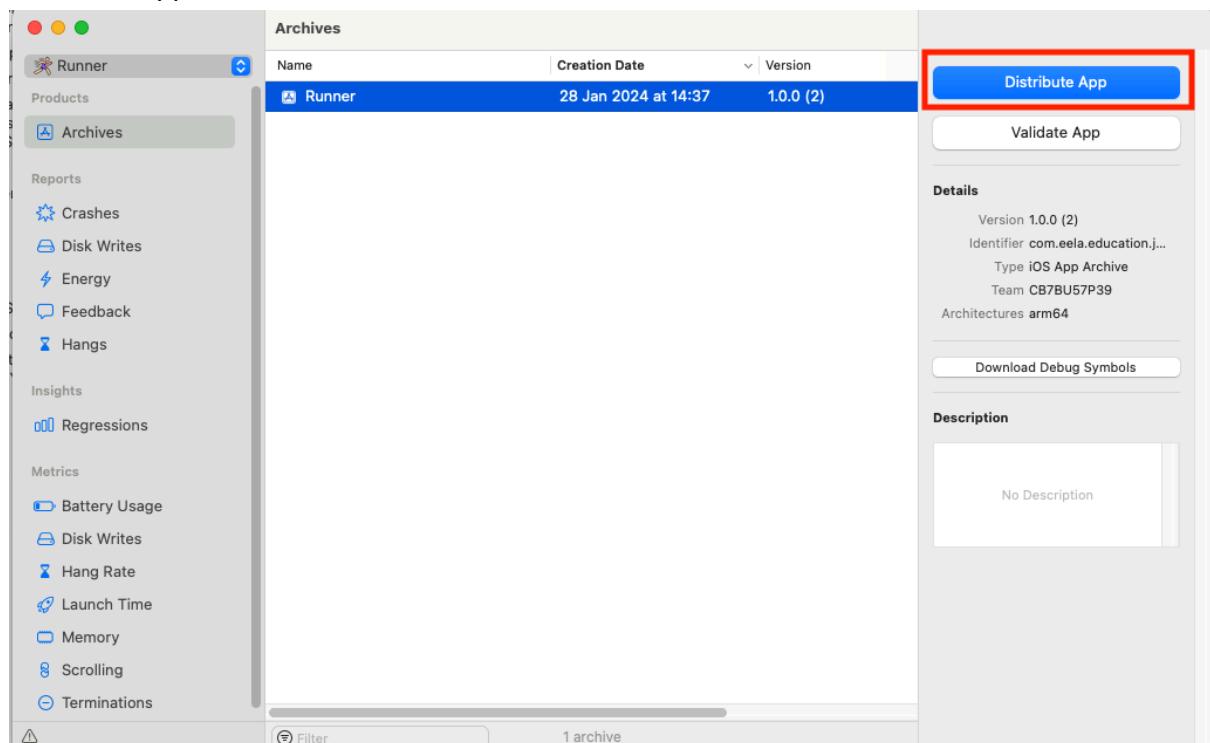
Pero si estamos subiendo la aplicación en la cuenta de otra persona o empresa, desmarcamos esta opción y seleccionamos el Provisional profile que generamos al inicio. [Aqui puedes consultar como generar este certificado](#)



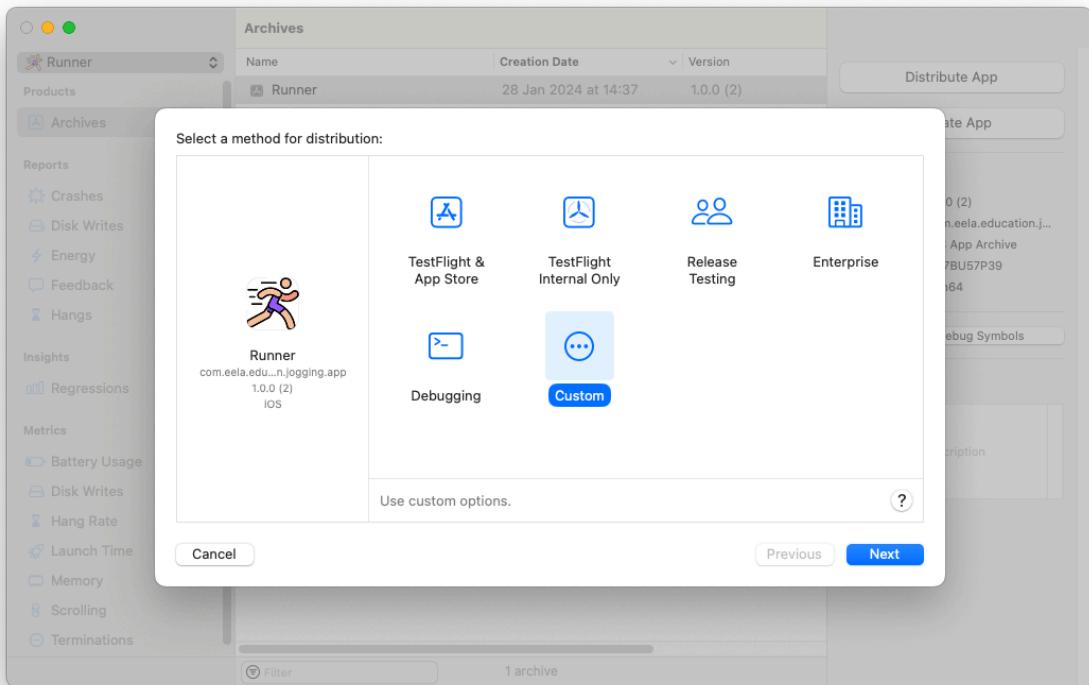
Ahora, en la barra de herramientas, elegimos la opción **Product**, y luego **Archive**. Esto va a empezar a generar nuestro archivo para subir a la tienda de Apple.



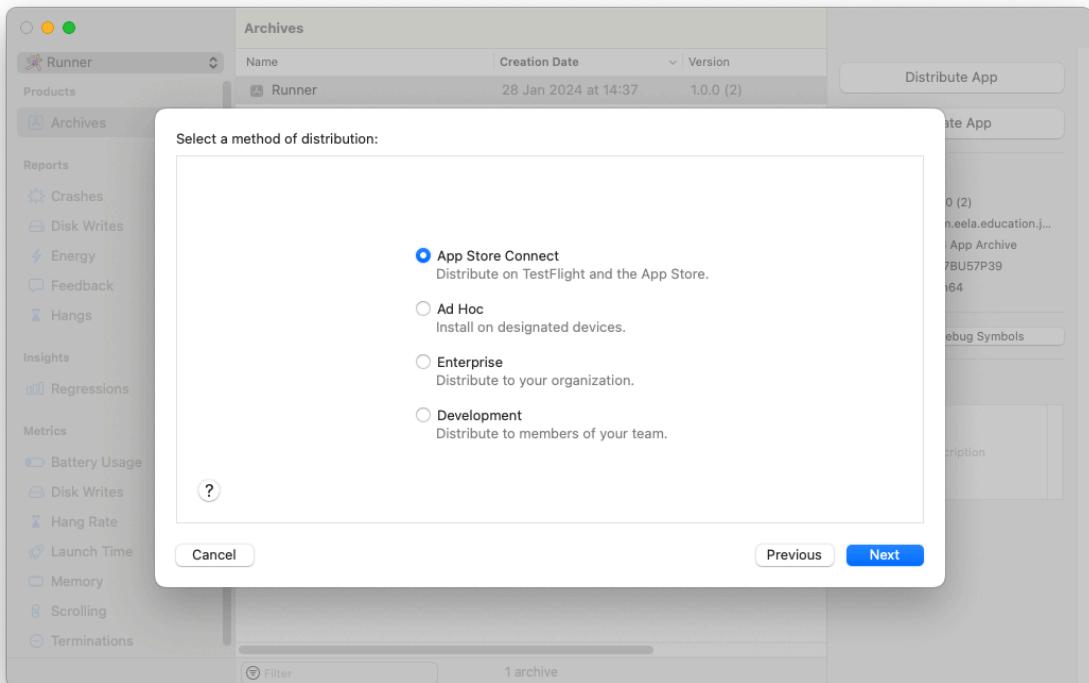
Cuando se termina de construir el archivo, nos muestra esta ventana. Elegimos la opcion Distribute App



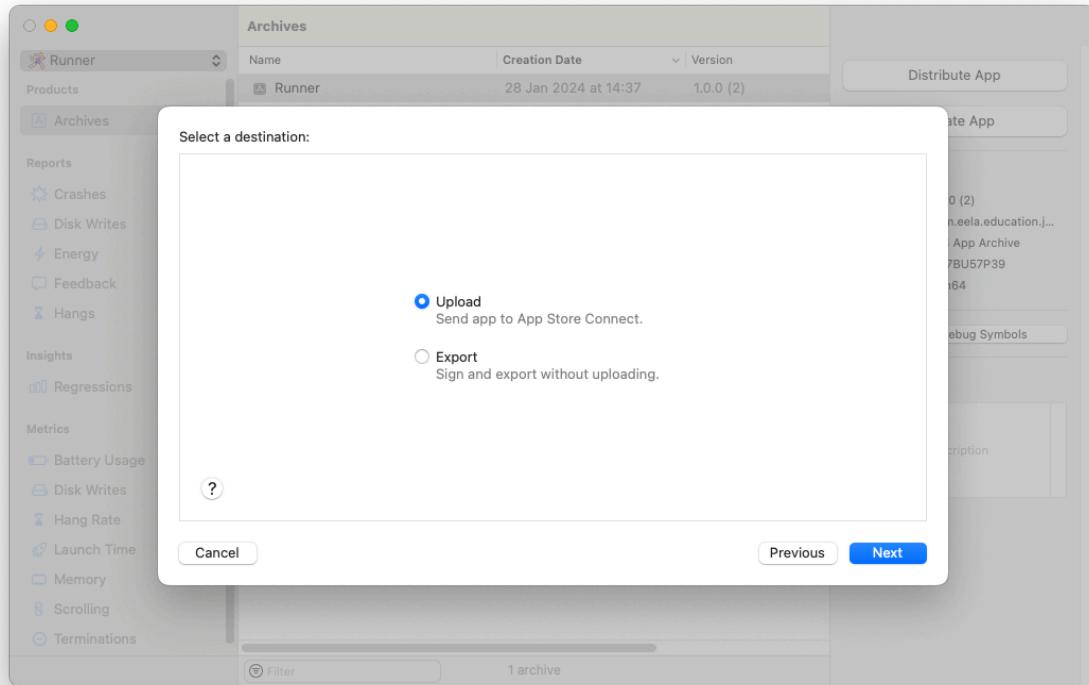
Seleccionamos el método de distribución, en este caso **Custom**



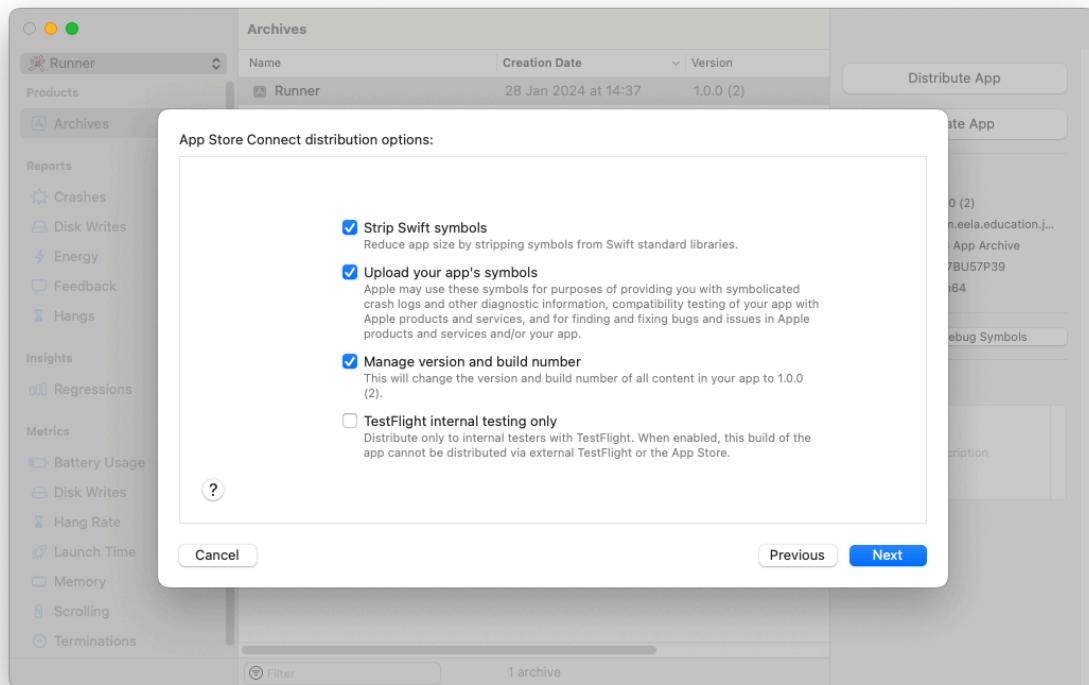
## Seleccionamos App Store Connect



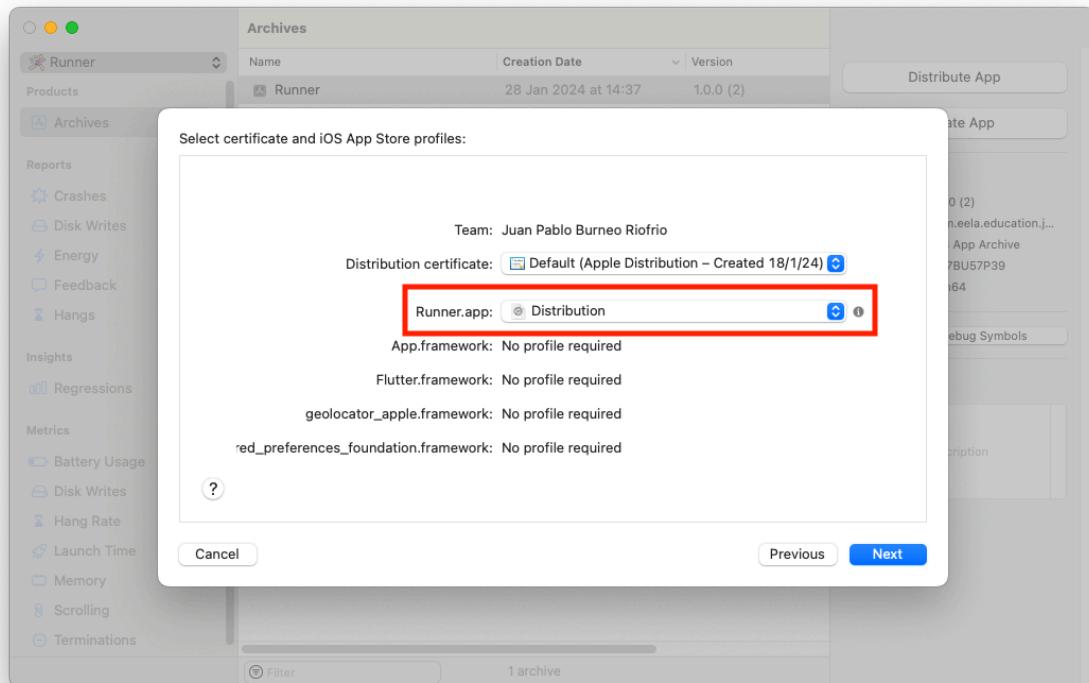
## Elegimos el destino, Upload



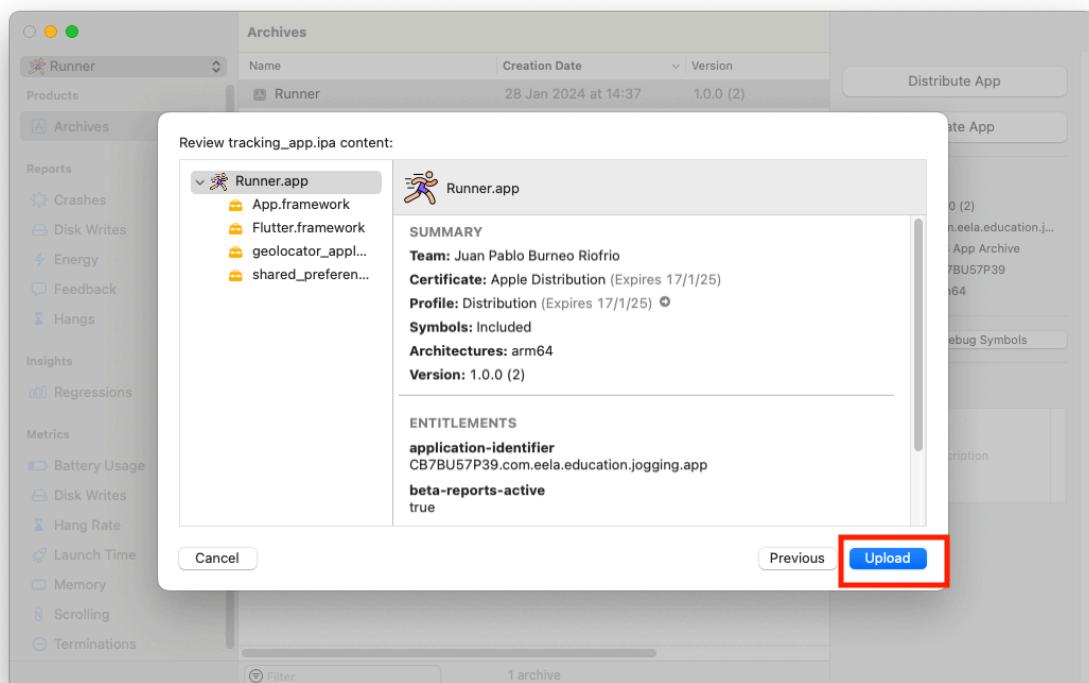
Marcamos las 3 primeras opciones



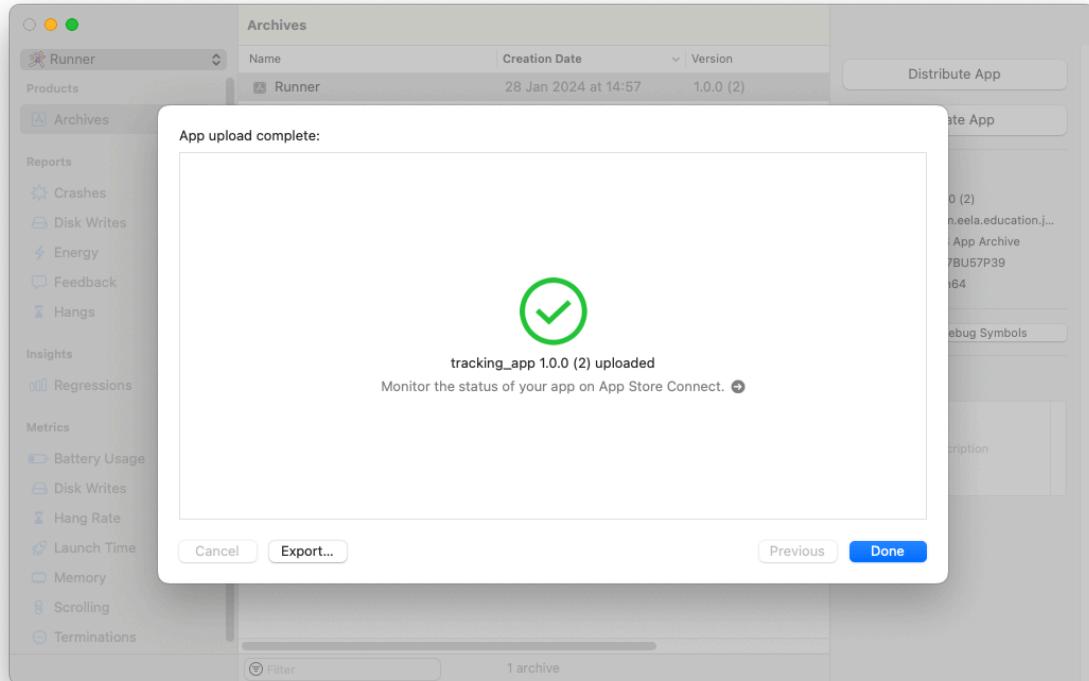
Ahora nos va a pedir elegir los certificados, en **Runner.app** elegimos el Provisional profile de distribución que generamos al inicio



Nos va a mostrar un resumen antes de subir la aplicación. Continuamos con **Upload**



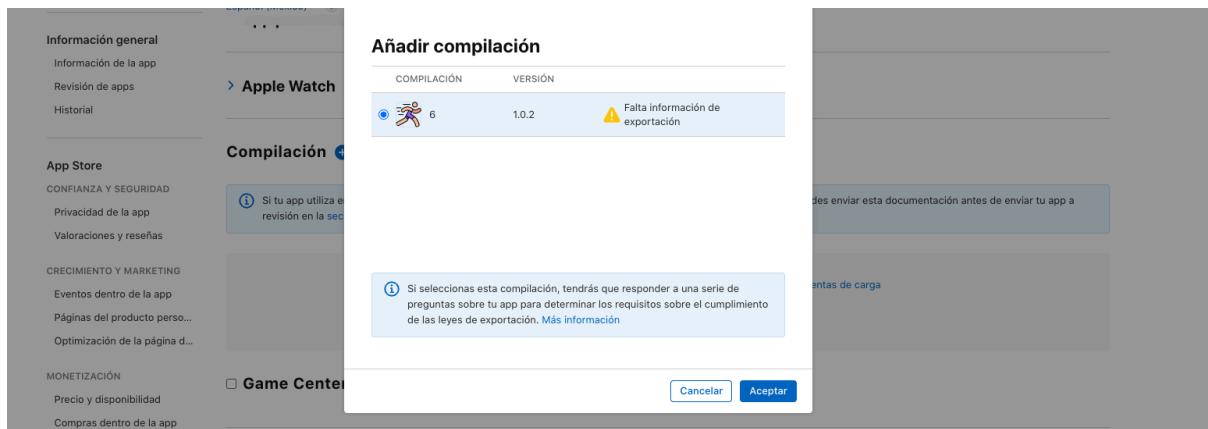
Si todo está bien, nos muestra la información que se subió con éxito. Pero todavía no está publicado. Vamos al siguiente paso.



En **App Store Connect**, elegimos la versión de nuestra app que empezamos a configurar y luego bajamos hasta la sección de Compilación y seleccionamos la opción **Añadir compilación**.

The screenshot shows the 'App para iOS' section of the App Store Connect interface. On the left, there is a sidebar with sections like 'Información general', 'App Store', 'CONFIANZA Y SEGURIDAD', and 'Privacidad de la app'. The main area is titled 'App para iOS Versión 1.0.2'. It shows two build versions: '1.0.2 En preparación para...' (highlighted with a red box) and '1.0.1 Listo para distribuci...' (green dot icon). Below these are buttons for 'Guardar' and 'Añadir a revisión'. Under the title, it says 'usuarios instalan la app.' and 'Español (México)'. A 'Composición' section follows, with a note about encryption documentation and a 'Añadir compilación' button highlighted with a red box. At the bottom, there is a note about loading builds with tools and another 'Añadir compilación' button.

Aquí ya vamos a ver la versión que subimos desde Xcode y la seleccionamos.



Una vez que ya tenemos el archivo de compilación y toda la información requerida por la App Store, podemos enviar nuestra aplicación a revisión. Para las Políticas de Privacidad, pueden usar esta plataforma que es la misma que utilizamos para la Play Store

<https://app.flycricket.com/>

El tiempo de revisión ahora está siendo muy rápido, si tenemos suerte en un día la revisan y si todo está bien recibiremos un correo de confirmación.

## App Store Connect

---

### Congratulations!



We're pleased to let you know that your app, EELA Jogging, has been approved for distribution.

Please note that it can take up to 24 hours for apps to become available on the App Store after release. If your contracts are not yet in effect, your app cannot be distributed. Check the status of your contracts in the [Agreements, Tax, and Banking](#) module in App Store Connect.

### Así es como se ve en la App Store

#### App Store Preview

This app is available only on the App Store for iPhone and iPad.



#### EELA Jogging (17+)

Juan Pablo Burneo Riofrio

Free

#### Screenshots iPhone iPad

