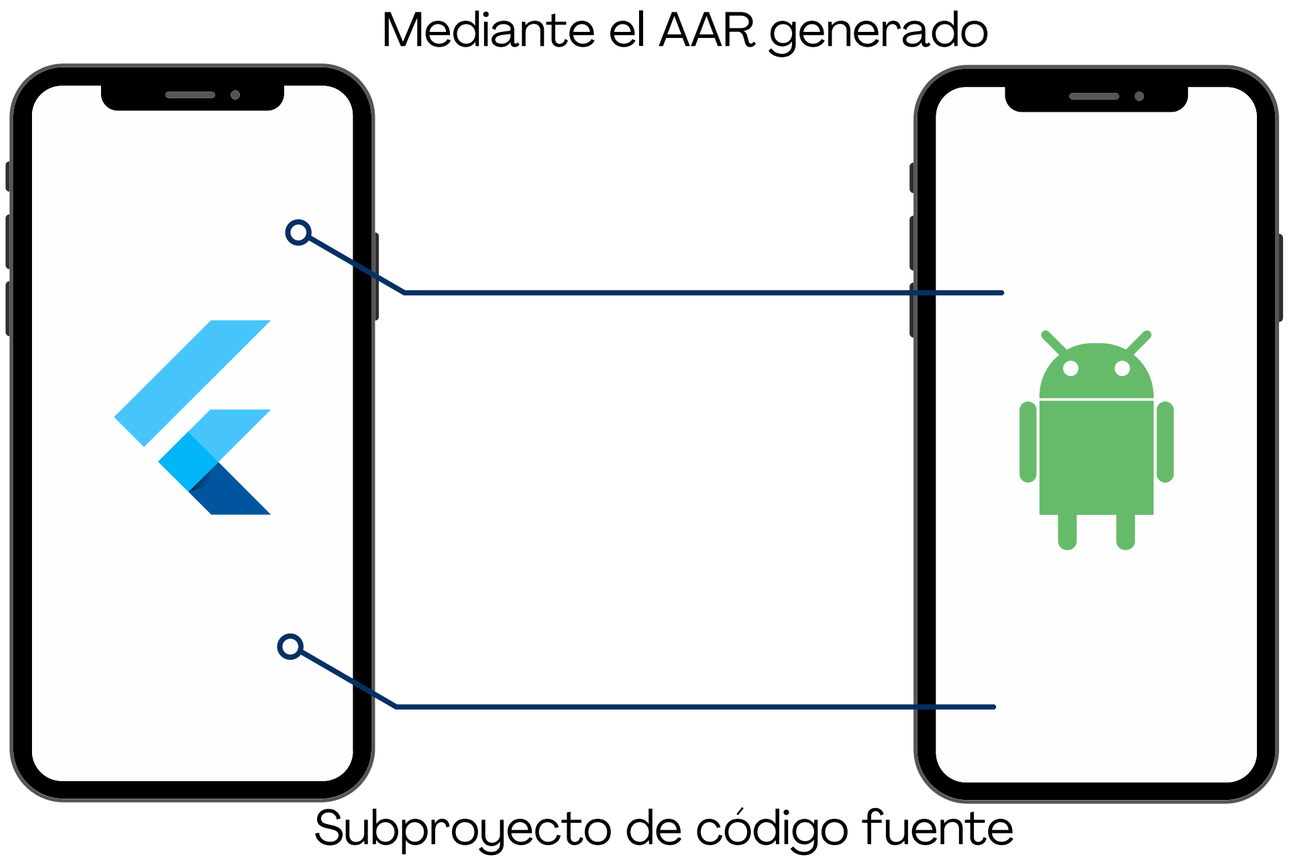
# Integrar un módulo Flutter en un proyecto Android

**Camilo A. Díez,**
Flutter Developer

Globant

# Tipos de integración




Flutter puede embeberse en una aplicación Android existente de forma fragmentada, como un subproyecto Gradle de código fuente o como AARs.

# Pros y Contras de integrar un AAR

## PROS

- Fácil de integrar, sin muchas modificaciones o configuraciones

## CONTRAS

- No se pueden hacer modificaciones en caliente al módulo. Se debe regenerar si es requerido.

# Pros y Contras de integrar el código Fuente

## PROS

- Puedes modificar el código del módulo de Flutter en caliente (Se debe recompilar el proyecto de Android para ver los cambios.

## CONTRAS

- Se requiere instalación del SDK de Flutter
- Configuraciones adicionales comparadas con el otro método.

.

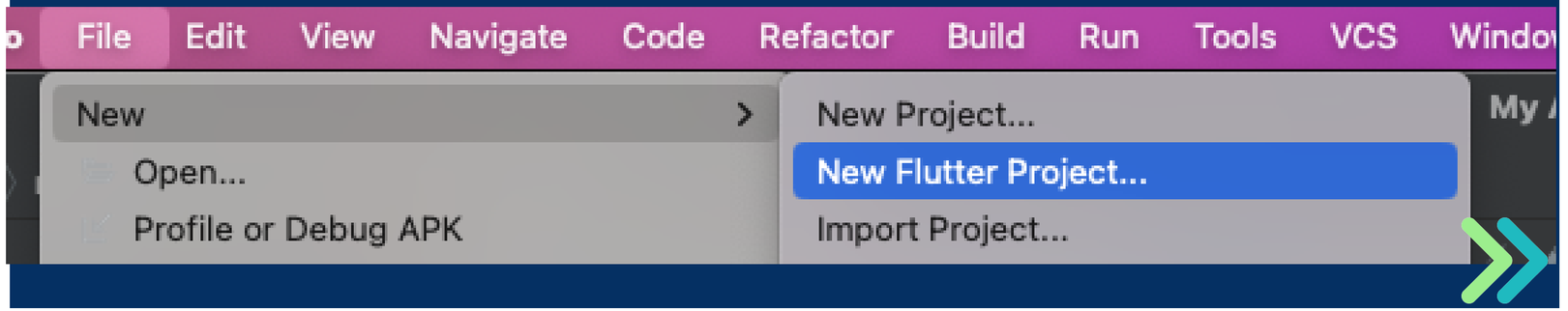
# Primer paso

## Crear módulo de Flutter

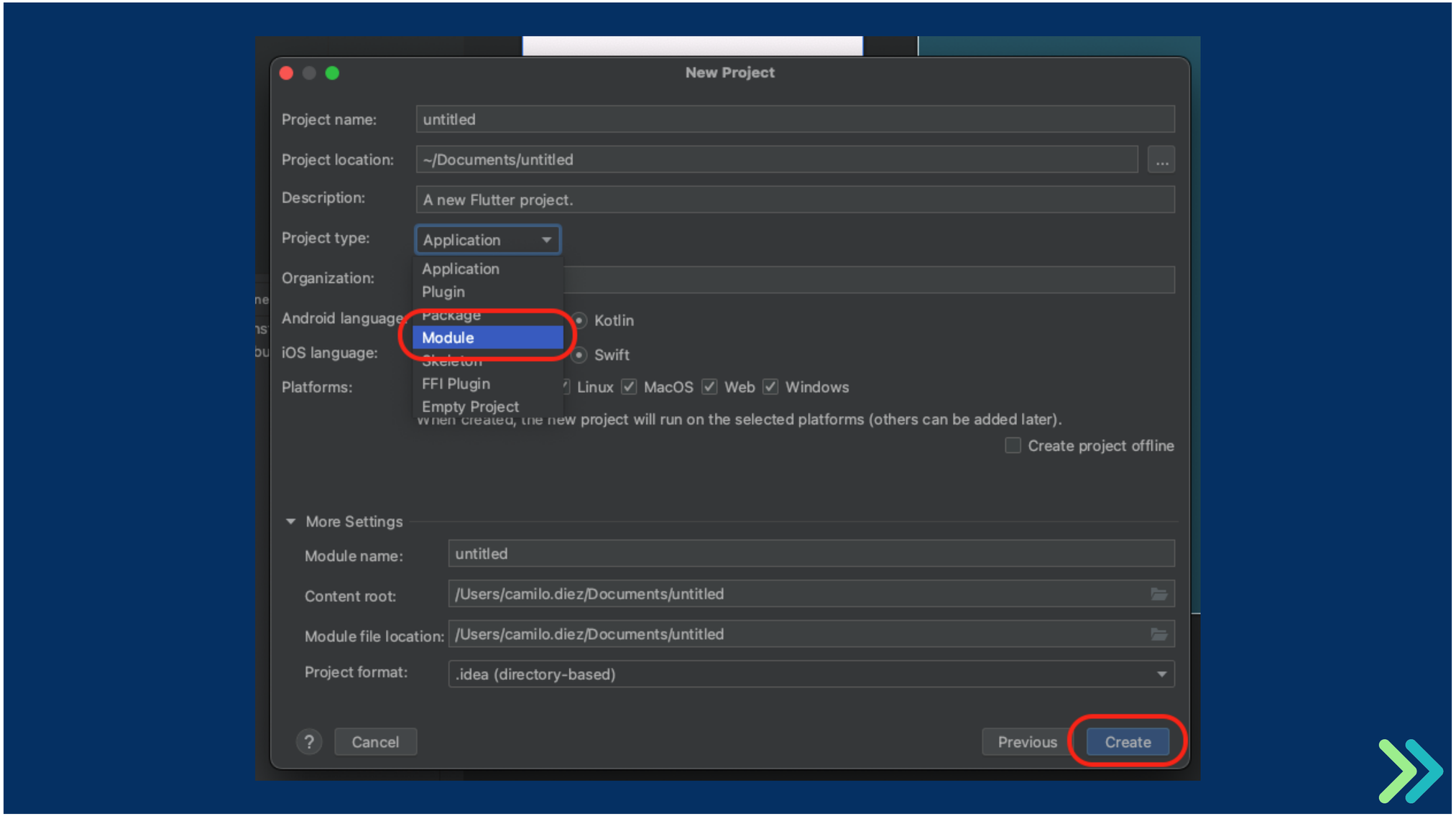
Mediante consola

```
cd some/path/
$ flutter create -t module --org com.example flutter_module
```

Con Android Studio



# Primer paso: Crear módulo de Flutter



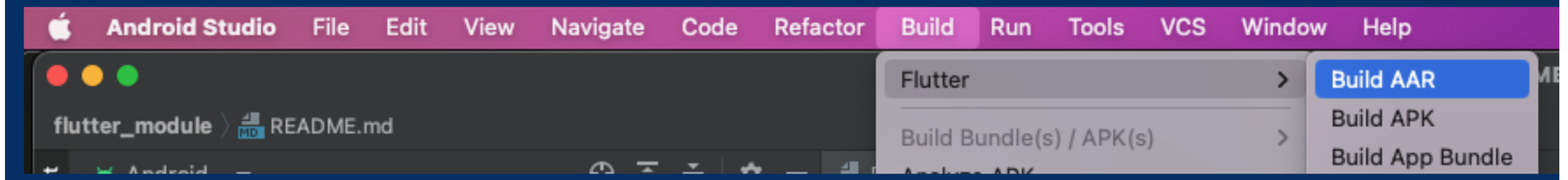
# Segundo paso

## Crear artefacto AAR módulo de Flutter

Mediante consola

```
cd some/path/flutter_module
$ flutter build aar
```

Con Android Studio



# Tercer paso

## Consumir el módulo

**OPCIÒN 1**

**Integrando dependencia AAR:** Agregamos la carpeta contenedora del artefacto como repositorio maven y en las dependencias.

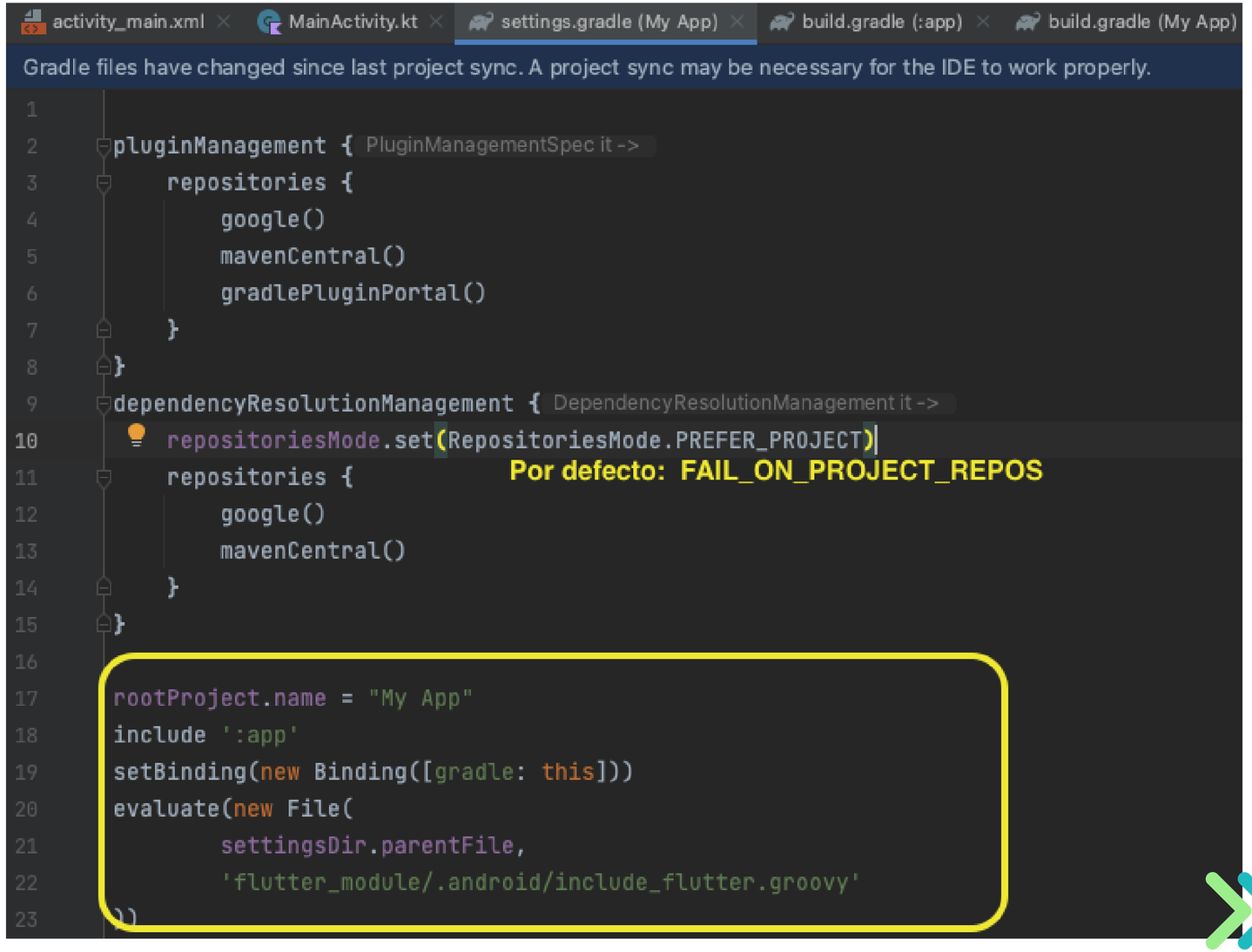
```
Running Gradle task 'assembleAarDebug'...                          35.7s
✓ Built build/host/outputs/repo.
Running Gradle task 'assembleAarProfile'...                        33.6s
✓ Built build/host/outputs/repo.
Running Gradle task 'assembleAarRelease'...                        32.7s
✓ Built build/host/outputs/repo.

Consuming the Module
  1. Open <host>/app/build.gradle
  2. Ensure you have the repositories configured, otherwise add them:

    String storageUrl = System.env.FLUTTER_STORAGE_BASE_URL ?: "https://storage.googleapis.com"
    repositories {
      maven {
          url '/Users/camilo.diez/Documents/flutter_module/build/host/outputs/repo'
      }
      maven {
          url "$storageUrl/download.flutter.io"
      }
    }

  3. Make the host app depend on the Flutter module:

  dependencies {
    debugImplementation 'com.example.flutter_module:flutter_debug:1.0'
    profileImplementation 'com.example.flutter_module:flutter_profile:1.0'
    releaseImplementation 'com.example.flutter_module:flutter_release:1.0'
  }

  4. Add the `profile` build type:

    android {
      buildTypes {
        profile {
          initWith debug
        }
      }
    }

To learn more, visit https://flutter.dev/go/build-aar
Process finished with exit code 0
```

# Tercer paso

## Consumir el módulo

**OPCIÒN 2**

**Integrar Código Fuente:**
Realizar el binding con el módulo creado en el primer paso.



# Tercer paso: Integración

**OPCIÒN 2**

**Integrar Código Fuente:**
Si los settings del proyecto fueron generados por una de las versiones más recientes de Gradle debemos agregar el acceso a los repositorios externos como Google o Maven y también se debe agregar la dependencia del modulo Flutter

## Agregar repositorioes en build.gradle (project)

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
id 'com.android.application' version '8.1.2' apply false
    id 'org.jetbrains.kotlin.android' version '1.8.10' apply false
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
```

## Agregar dependencia en build.gradle (module)

```
dependencies {
    ...
    implementation project(':flutter')
}
```

# Último paso:  Agregar Pantalla a la app de Android

## 1. Agregar Actividad al Manifest

```
<activity
    android:name="io.flutter.embedding.android.FlutterActivity"
    android:theme="@style/Theme.AndroidApp"
    android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
    android:hardwareAccelerated="true"
    android:windowSoftInputMode="adjustResize"
    />
```

## 2. Importar librería

```
import io.flutter.embedding.android.FlutterActivity
```

## 3. Iniciar FlutterActivity

```
private fun goToFlutterModule(){
    startActivity(
        FlutterActivity.createDefaultIntent( launchContext: this)
    )
}
```

# Demo de la integración con la opción 2 (código fuente).

# CONCLUSIONES

**Adaptabilidad**

Adaptabilidad alta de Flutter para proyectos nativos de Android.

**Integración**

Diferentes vías de integración.

**Simple**

Fácil de integrar, configurar e implementar.

**Recuerda:** Es Importante tener en cuenta la configuración de los repositorios externos si el gradle es generado mediante el SDK de Android

# Fin de la Presentación, Gracias por su atención!

**EXPOSITOR:**
**Camilo A. Díez,**
Flutter Developer
Globant

Repo del ejercicio