

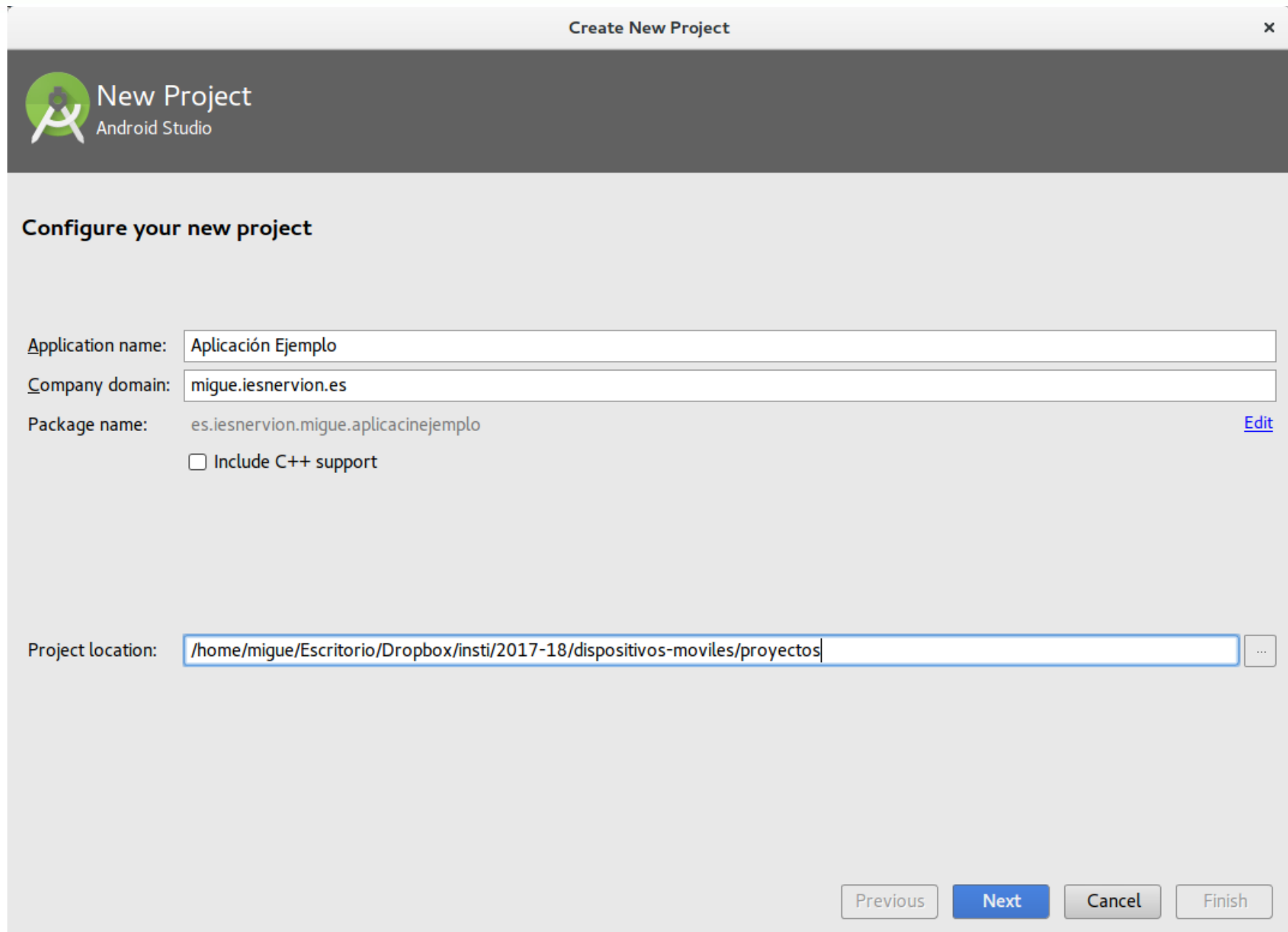
Programación multimedia y dispositivos móviles

2

Empezando a desarrollar en Android

IES Nervión
Miguel A. Casado Alías

Creación de un proyecto en Android Studio (I)



The screenshot shows the 'Create New Project' dialog in Android Studio. The window title is 'Create New Project'. The header bar is dark gray with the Android Studio logo and the text 'New Project' and 'Android Studio'. The main area is light gray and contains the heading 'Configure your new project'. Below this, there are four input fields: 'Application name' with the value 'Aplicación Ejemplo', 'Company domain' with the value 'migue.iesnervion.es', 'Package name' with the value 'es.iesnervion.migue.aplicacinejemplo' and an 'Edit' link to its right, and 'Project location' with a long file path and a browse button (three dots). There is also an unchecked checkbox for 'Include C++ support'. At the bottom right, there are four buttons: 'Previous' (disabled), 'Next' (active/blue), 'Cancel' (disabled), and 'Finish' (disabled).

Create New Project

New Project
Android Studio

Configure your new project

Application name: Aplicación Ejemplo

Company domain: migue.iesnervion.es

Package name: es.iesnervion.migue.aplicacinejemplo [Edit](#)


☐ Include C++ support

Project location: /home/migue/Escritorio/Dropbox/insti/2017-18/dispositivos-moviles/proyectos

Previous Next Cancel Finish

Creación de un proyecto en Android Studio (II)

Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

API 14: Android 4.0 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.
By targeting API 14 and later, your app will run on approximately **100,0%** of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK

API 21: Android 5.0 (Lollipop)

☐ Android Auto

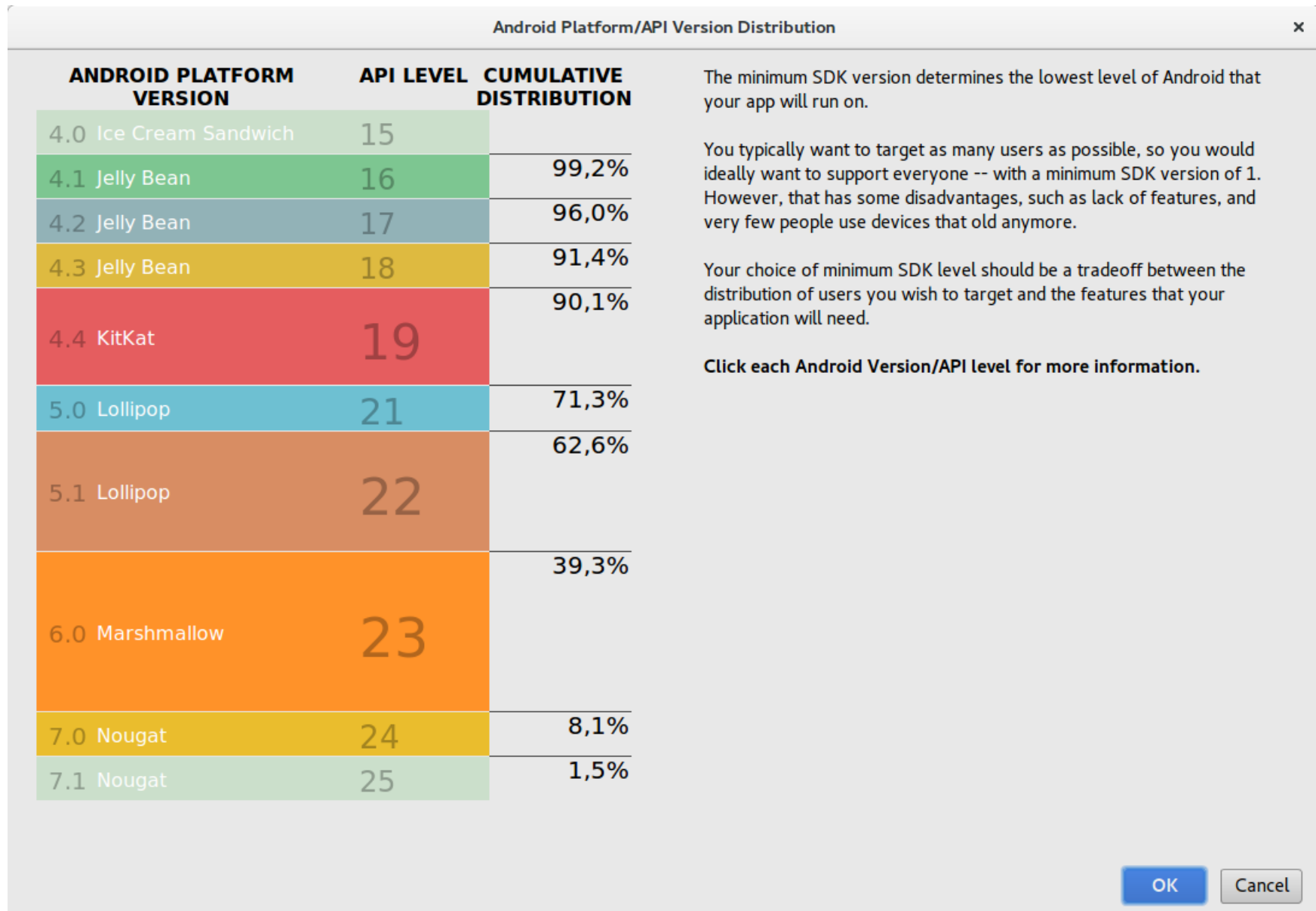
Previous

Next

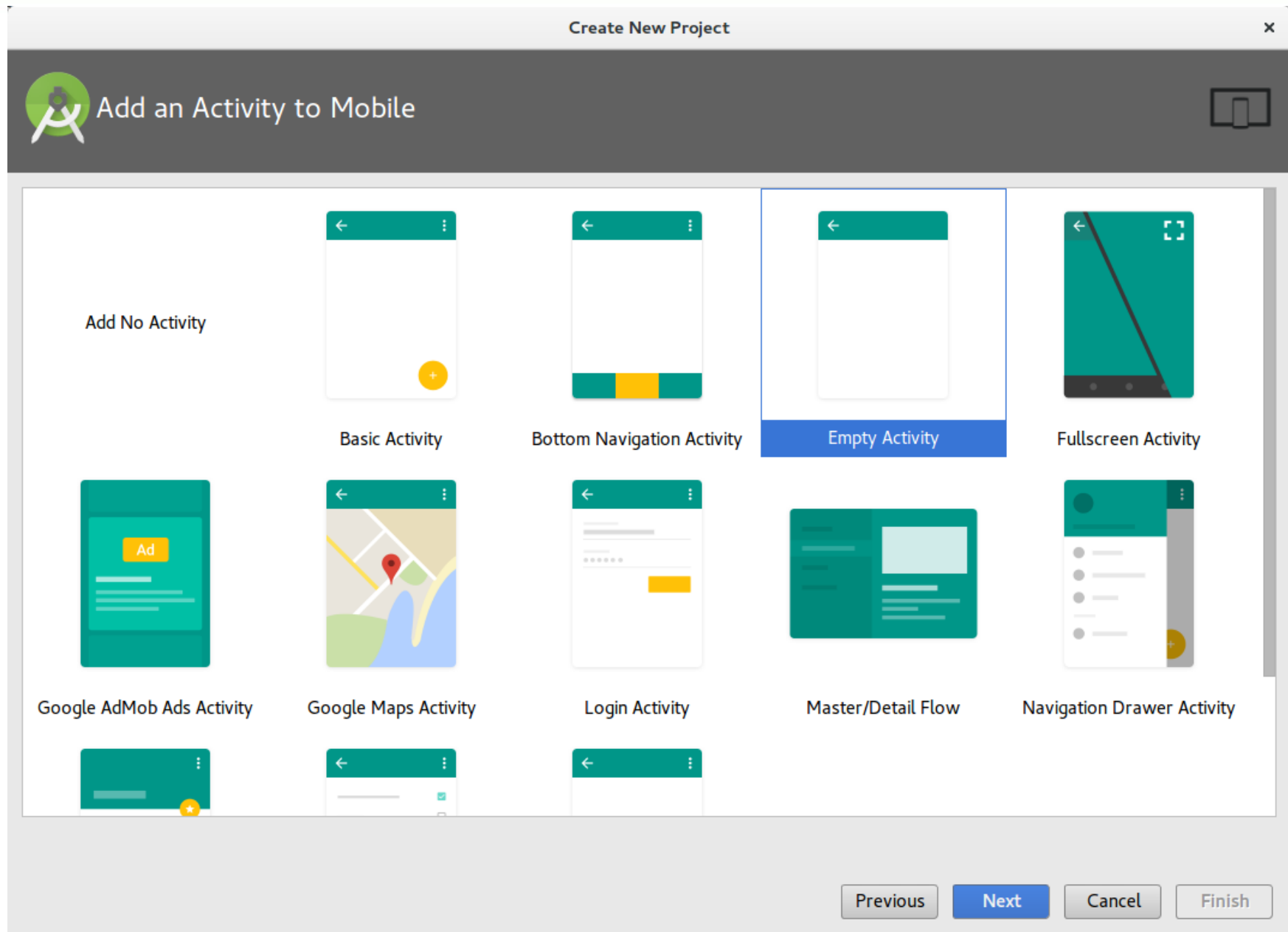
Cancel

Finish

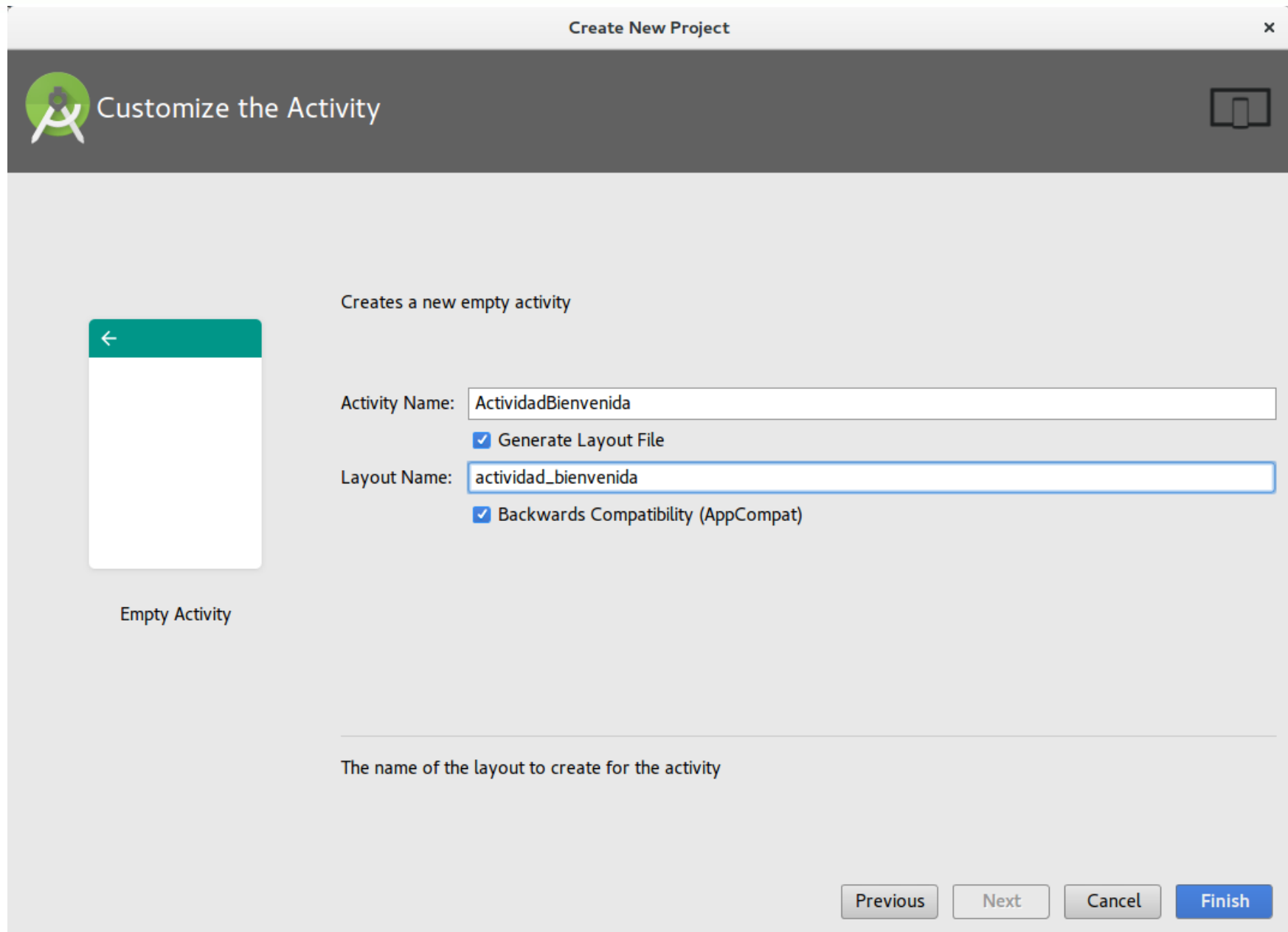
Creación de un proyecto en Android Studio (III)



Creación de un proyecto en Android Studio (IV)



Creación de un proyecto en Android Studio (V)



The screenshot shows the 'Create New Project' dialog in Android Studio, specifically the 'Customize the Activity' screen. The dialog has a title bar 'Create New Project' and a close button. Below the title bar is a header with the Android Studio logo and the text 'Customize the Activity'. On the left side, there is a preview of an 'Empty Activity' with a back arrow. The main area contains the following options:

- Creates a new empty activity
- Activity Name:
- ☒ Generate Layout File
- Layout Name:
- ☒ Backwards Compatibility (AppCompat)

At the bottom, there is a section titled 'The name of the layout to create for the activity' which is currently empty. At the very bottom, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

Principales componentes de una aplicación

- Activities
- Layouts y Views
- Intents
- Services
- Content Providers
- Broadcast Receivers
- Application Context

Activities

- Podríamos decir que una actividad es una pantalla con todos sus elementos y acciones asociadas
- Las aplicaciones suelen tener varias actividades
- Al igual que un sitio web está compuesto de múltiples páginas webs, se puede pensar en una aplicación como que está compuesta de varias actividades
- El usuario “navega” de actividad en actividad
- Se puede “saltar” de una actividad de una aplicación a una actividad de otra aplicación
- Suele haber una actividad principal, que es la primera que se muestra al entrar en la aplicación

Layouts y Views

- Las vistas son todos los elementos como botones, listas, cuadros de texto, etc...
- Los layouts agrupan las vistas para organizarlas
- Un layout puede contener otros layouts

Intents

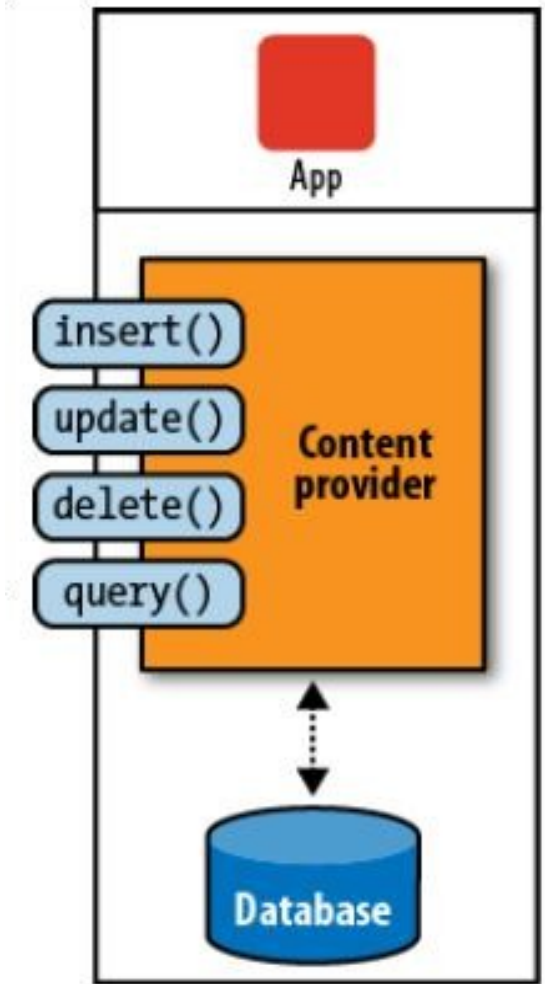
- No se deben traducir como “intentos”, sino más bien como un propósito, como la intención de hacer algo
- Son mensajes mediante los cuales las aplicaciones intercambian información con otras aplicaciones o con el sistema. Ejemplos:
 - El sistema notifica que se ha insertado una tarjeta SD
 - El sistema notifica que hay cobertura Wi-Fi
 - Una aplicación solicita abrir una página web, en tal caso lanza un intent, y todas las aplicaciones capaces de abrir una página web (por ejemplo el navegador Firefox) se prestan para completar dicha acción
- Una aplicación lanza un intent cuando solicita algo del sistema o de otra aplicación

Services

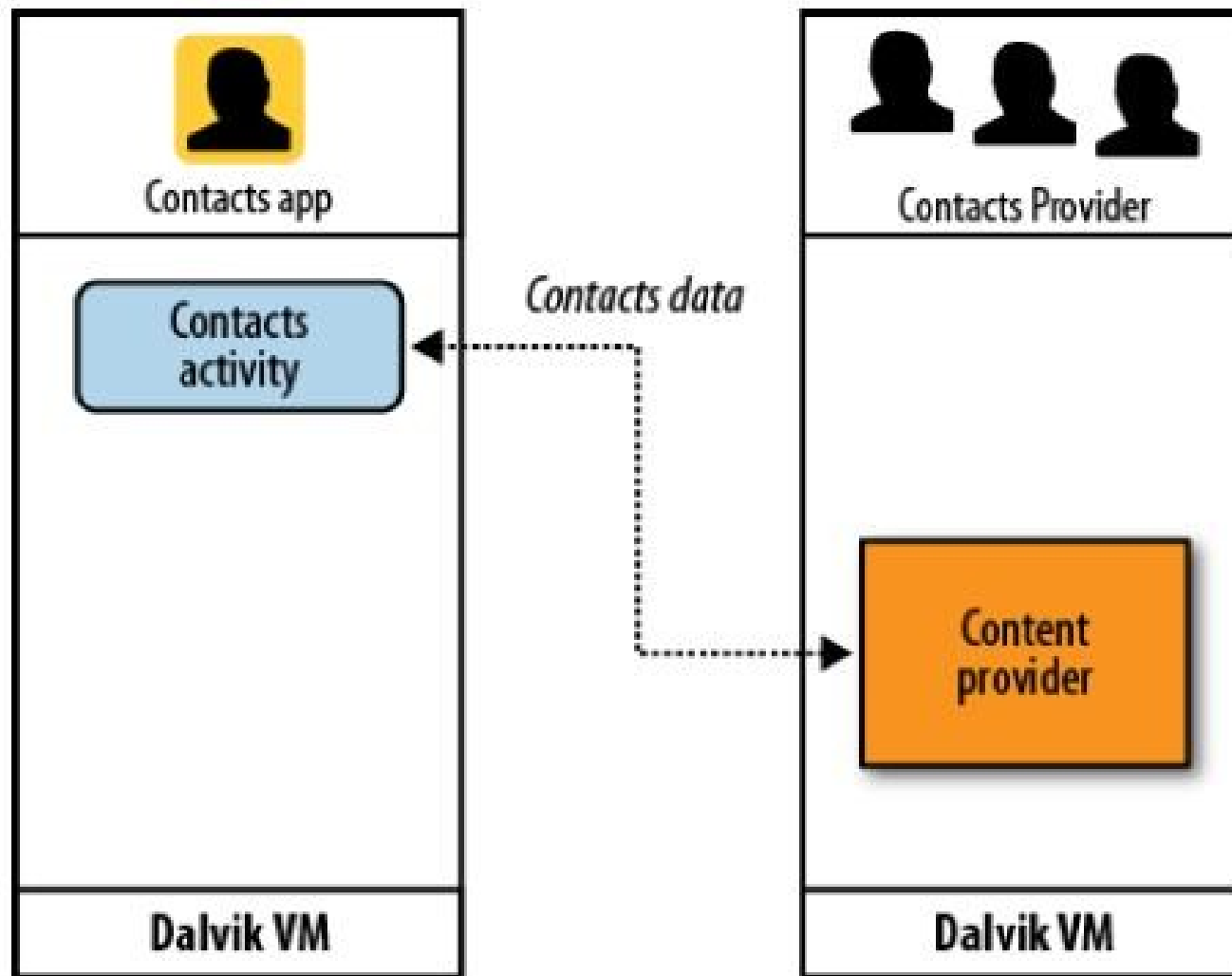
- Se ejecutan en segundo plano
- No tiene interfaz de usuario
- Son útiles para llevar a cabo acciones durante un buen tiempo independientemente de lo que se esté mostrando en la pantalla. Por ejemplo:
 - Queremos descargar un fichero de gran tamaño
 - Queremos hacer sonar música y que ésta no cese cuando estemos usando otras aplicaciones

Content Providers

- Son interfaces para compartir datos entre aplicaciones
- Por defecto Android ejecuta cada aplicación en su propio “contenedor”, aislando a cada aplicación y sus datos de las demás aplicaciones
- Ejemplos:
 - “Contacts Provider” proporciona información sobre los contactos a las aplicaciones que la necesiten
 - “Settings Provider” comparte los ajustes del sistema con las aplicaciones



Content Providers (II)



Broadcast Receivers

- Código destinado a llevar a cabo una acción cuando ocurra un evento determinado
- El sistema lanza mensajes broadcast continuamente. Por ejemplo: cuando llega una llamada, cuando se recibe un SMS, cuando la batería está cargada, etc...
- Si quisiéramos hacer una aplicación que baje el brillo de la pantalla al mínimo cuando la batería esté apunto de agotarse, deberíamos hacerlo mediante un “broadcast receiver”

Application Context

- Actividades, Servicios, Proveedores de contenidos, etc... todos juntos forman una aplicación. Podríamos decir que todos ellos habitan en el mismo “**Contexto**”
- El contexto de la aplicación hace referencia al entorno de la misma y al proceso dentro del cual todos los componentes de la aplicación se ejecutan
- El contexto permite a los diversos “bloques de construcción” de la aplicación intercambiar información y recursos
- Las actividades y los servicios son subclases de contexto

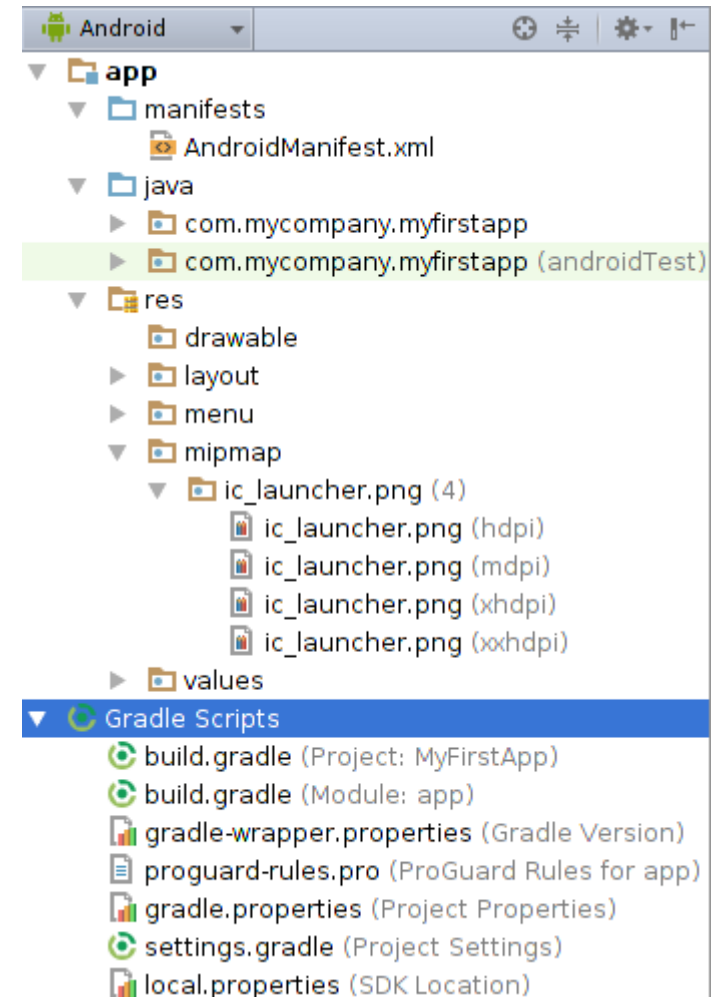
Explorando un proyecto Android

■ Vista ANDROID:

- No refleja la jerarquía real de archivos en disco
- Se organiza en módulos y tipos de archivos
- Se ocultan algunos ficheros y directorios

■ Vista PROJECT:

- Refleja la jerarquía real de archivos en disco



Explorando un proyecto Android (II)

- AndroidManifest.xml: describe la aplicación y sus componentes
- build/ : alberga la aplicación una vez compilada
- libs/ : librerías de terceros usadas por nuestra aplicación
- res/ : recursos como imágenes, layouts, etc...
- src/ : código fuente de nuestra aplicación
- assets/ : más recursos, no accesibles a través de clase R
- gen/ : código generado por las herramientas de desarrollo de Android

<https://developer.android.com/studio/projects/index.html#ProjectFiles>

La clase R.java

- La primera vez que se compila un proyecto se genera automáticamente R.class
- Contiene una serie de constantes enlazadas a los recursos disponibles en el directorio res/
- En nuestro código podemos hacer referencia a los recursos escribiendo algo como R.drawable.mi_imagen
- NO SE DEBE MODIFICAR

Layouts XML vs Layouts Java

- Layouts Java
 - Es mejor usarlos sólo cuando necesitamos instanciación dinámica de las vistas
 - Ejemplo: ver proyecto Skeleton/Now
- Layouts XML
 - Recomendado. Separamos visualización del contenido
 - Los creamos en res/layout
 - Los enlazamos con el código fuente java a través del método *setContentView* y valiéndonos de la clase R.java
 - Accedemos a los elementos mediante *findViewById*
 - Ejemplo: ver proyecto Layouts/NowRedux

@+id

- Necesitamos poner un identificador a aquellos elementos de nuestro Layout basado en XML a los que queramos acceder desde el código java
 - Ejemplo: **android:id="@+id/boton1"**
- La convención es anteceder el nombre con **@+id/** **la primera vez que aparezca** ese elemento en nuestro fichero XML
- Con el "+" indicamos al parser que estamos creando un nuevo identificador, y que debe ser añadido a nuestra clase R.java
- Tras esa primera vez, nos referiremos al identificador con **@id/nombre_del_identificador**
 - Ejemplo: **android:id="@id/boton1"**