

# Fichero “AndroidManifest.xml”

# Misión del manifest? (I)

- ▶ cada aplicación debe tenerlo obligatoriamente.
- ▶ Contiene información necesaria para el empaquetado de la información
- ▶ **Informa a Android** de características importantes de una aplicación.
- ▶ Sin el Manifest, los **componentes no son accesibles** por parte del sistema.

# Misión del manifest? (II)

- ▶ **La aplicación debe declarar en este archivo todos los componentes que use** (activities, broadcast receivers, services, etc.)

# ¿Qué hace el manifest? (I)

## 1. Identifica el paquete Java para la aplicación:

- 📄 El **nombre del paquete** sirve como **IDENTIFICADOR ÚNICO** para la **aplicación**.

## 2. Describe los componentes de la aplicación:

- 📄 Indicando las **clases** que implementan cada **componente**.
- 📄 Describiendo las **capacidades que tiene cada componente** (*por ejemplo: qué tipo de Intents acepta*). Esto sirve para que Android sepa qué componentes son y cuando deben ejecutarse y **quién los puede ejecutar**.

## ¿Qué hace el manifest? (II)

3. Declara los permisos que necesita la aplicación para acceder a partes protegidas de la API e interactuar con otras apps (*escribir en la tarjeta SD, acceder a la red, acceder a Internet,...*)
4. Declara los permisos que otras apps deben tener para que puedan interactuar con los componentes de nuestra aplicación.

## ¿Qué hace el manifest? (III)

5. Declara la **versión de nuestra aplicación** con la que aparecerá en el Google Play.
6. Declara características de hardware requeridos por la aplicación (*por ej.: multitouch, resoluciones de pantalla, cámaras,...*)
7. Declara librerías API que necesita la aplicación (*además de las de Android*), como la API de Google Maps u otras.
8. Describe las clases de instrumentación (*Instrumentation*) que proveen medidores de rendimiento e información extra cuando la aplicación está funcionando.

# ESTRUCTURA DEL MANIFEST

# Estructura del manifest (I)

- ▶ Ver en:

<https://developer.android.com/guide/topics/manifest/manifest-intro.html>

- ▶ Empieza con la línea:

`<?xml version="1.0" encoding="utf-8"?>`

- Por ser un XML.



# Estructura del manifest (II)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.alejandra.activityejercicio02"
    android:versionCode="1"
    android:versionName="1.0" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="Activity: Ejercicio 02"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.android.alejandra.activityejercicio02.MainActivity"
            android:label="Activity: Ejercicio 02" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".CicloVidaActivity"
            android:label="Ciclo de Vida de la Activity" >7
        </activity>
    </application>

</manifest>
```

Pertenecen al namespace  
"android". Versión de  
nuestra app.

Nodo raíz (**obligatorio**),  
*no se puede repetir*

Nodo application  
(**obligatorio**) *no se  
puede repetir*

Declarando  
**componente**  
**ACTIVITY**  
(MainActivity)

Declarando **componente**  
**ACTIVITY**  
(CicloVidaActivity)

# Estructura del manifest (III):

## Convenciones y reglas que se aplican

### ► Atributos:

- ☞ Casi todos son **opcionales**.
- ☞ Todos **empiezan con un prefijo “android:”**, excepto algunos atributos del elemento “<manifest>”.

### ► Múltiples valores:

- ☞ Si se puede **especificar más de un valor**, el elemento **se repite**, en lugar de enumerar varios valores dentro de un solo elemento.
- ☞ Por *ejemplo*, un “**intent-filter**” puede enumerar varias acciones (*actions*):

```
<intent-filter . . . >
    <action android:name="android.intent.action.EDIT" />
    <action android:name="android.intent.action.INSERT" />
    <action android:name="android.intent.action.DELETE" />
    . . .
</intent-filter>
```

# Estructura del manifest (IV): Convenciones y reglas que se aplican

## ► ¿Cómo referenciar recursos?

- Los **recursos** están almacenados en subdirectorios del directorio “/app/src/main/res”.



# Iconos y Etiquetas(I)

- ▶ Algunos elementos (*etiquetas*) del manifest tienen atributos: “**icon**” y “**label**”.
- ▶ Son iconos y textos que pueden ser mostrados al usuario.
- ▶ Por ejemplo: *<application>*, *<activity>*, *<service>*, ... los tienen
- ▶ Algunos pueden tener también un atributo “**description**” que puede albergar un **texto explicativo más largo** para mostrar al usuario.

✍ Ejemplo: el elemento **<permission>** tiene estos **tres atributos** por lo que cuando se le pregunta al usuario si desea conceder el permiso a una aplicación que lo ha solicitado, se le puede mostrar al usuario: un **icono** que representa el permiso, el **nombre** del permiso, y una **descripción** de lo que implica.

# Iconos y Etiquetas(II)

- ▶ Todos los **iconos y etiquetas** establecidos para un **ELEMENTO** (por ej `<application>`) son los iconos y etiquetas por defecto para todos sus subelementos.
- ▶ Igualmente, todos los **iconos y etiquetas** establecidos para un **COMPONENTE** (por ej `<activity>`) son los iconos y etiquetas por defecto para todos los `<intent-filter>` del componente.
- ▶ Si un elemento `<application>` tiene una “**label**” asignada, pero una **activity** y sus **intent-filter** no, la etiqueta de la aplicación, será considerada también como la etiqueta de dicha **activity** y sus **intent-filters**.

# Iconos y Etiquetas(III)

- ▶ El **icono** y la **etiqueta** fijados para un **<intent-filter>** se utilizan para representar un componente cada vez que el componente se presenta al usuario como resultado del cumplimiento de la función anunciada por el filtro.
- ▶ Por ejemplo:
  - ✂ Esto sucede, cuando varios componentes son capaces de responder al mismo intent.
  - ✂ En estos casos, Android muestra una lista de componentes al usuario para que éste escoja cuál quiere usar.
  - ✂ En esa lista aparecerá el icono y etiqueta asociado al **<intent-filter>**.

# Atributos de la etiqueta <manifest> (III)

- ▶ android:**versionCode**="..."
- ▶ android:**versionName**="..."

Pertenecen al namespace "android". Por eso van precedidos de "android:...."

- **versionCode** y **versionName** especifican la versión de nuestra aplicación de dos modos distintos.

- **versionCode** :

☞ tiene que ser un **número** que se tiene que ir incrementando cada vez que publiquemos una nueva versión de nuestra aplicación:

```
android:versionCode="1"
```

☞ Es usado por el Google Play para seguir nuestra aplicación.

- **versionName** es lo que se va a mostrar a los usuarios del Google Play. Por tanto, puede ser cualquier **String** que nos guste::

```
android:versionName="1.0"
```



# Atributos de la etiqueta <manifest> (IV)

▶ android:**installLocation**=“...”

- el atributo **installLocation** sólo está disponible si configuramos el “**target**” (*versiones de Android que admite nuestra aplicación*).
- Se usa para indicar dónde queremos que se instale nuestra aplicación.
- “**preferExternal**” indica al sistema que nuestra aplicación se instale en la tarjeta SD:

**android:installLocation=“preferExternal”**



# Estructura del manifest (V): *Resumen*

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.videotutoriales.videotutorial1"  
    android:versionCode="1"  
    android:versionName="1.0"  
    android:installLocation="preferExternal">  
  
</manifest>
```

- \*Componentes de la aplicación
- \* Permisos
- \* Perfiles de hardware
- \* Versiones de Android