DECLARANDO LOS COMPONENTES:

-Activity

-Service

-...

Mª Alejandra Tomás Fernández

Declarando componentes (I)

- La **tarea principal del Manifest** es informar a Android de los componentes usados.

- Un **ejemplo**:

```
<?xml  version="1.0" encoding="utf-8"?>
   <manifest... >
      <application  android:icon="@drawable/app_icon.png" ... >
          <activity
              android:name="com.ejemplos.project.EjemploActivity"
                      android:label="@string/ejemplo_label" ... >
          </activity>
            ...
       </application>
   </manifest>
```

Nombre cualificado de la clase

Declarando componentes (II)

- **No importa** que estén **declarados e implementados en el código Java**;
- **si un componente no está especificado en el manifest, no será visible para Android y no podrá ser utilizado**.
  - **Excepción**: los **broadcast receivers** pueden ser declarados en el código dinámicamente (como objetos **BroadcastReceiver)** y registrados en el sistema mediante **registerReceiver().**

Veremos en más detalle el manifest.

¿DÓNDE SE DEFINEN? (I)

- Los **componentes** (*activity, service, …)* se definen dentro de la **etiqueta <application>.**
- Habrá **atributos** que se puedan modificar independientemente **para cada componente**, como: *icon, label, permission, …*
- Otros **atributos** establecen **valores para la aplicación** en su conjunto y no se puede anular por un componente.

Describir los componentes de la aplicación (I): Atributos de la etiqueta <application> (I)

- android:**allowBackup**="…"
  - Hoy en día las **acciones Backup&Restore** son **muy usadas** debido a *actualizaciones de OS*, o *fallos múltiples* y *diversos* (**al enviar tu móvil a arreglar aunque sea una tecla, le hacen un bck&rst en casi todas las compañías…).**
  - Normalmente, **las aplicaciones descargadas y sus datos no son guardadas con el backup.**
  - Con las últimas **api's (17)**, en el **AndroidManifest.xml** de nuestros proyectos se añade una nueva línea por defecto:

    android:allowBackup="true" (true por defecto)

  - Esto, **habilita a la infraestructura de Android, al realizar un backup, a que guarde esta aplicación y sus datos.**

Describir los componentes de la aplicación (II): Atributos de la etiqueta <application> (V)

- ***Consecuencias de permitir el Backup***:

  - ***Positiva***:

    + El usuario final puede **hacer un backup de nuestras aplicaciones** y no las pierda.

  - ***Negativa***:

    – un **error de seguridad**: con el **USB debugging** activado, se puede sacar el backup fuera del dispositivo **y se puede leer los datos de backup**.

Declarando:

ACTIVITIES

Mª Alejandra Tomás Fernández

Describir los componentes de la aplicación (III): <activity>

- *Ejemplo de uso:*

```
<application android:icon="@drawable/icon" android:label="@string/app_name"
    <activity android:name=".SrSombreroActivity"
              android:label="Sr. Sombrero"
              android:screenOrientation="portrait"
              android:configChanges="keyboard|keyboardHidden|orientation">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

</application>
```

Describir los componentes de la aplicación (IV): Atributos de la etiqueta <activity> (I)

- android:**name**="…"
  - "**name**" especifica el nombre de la clase activity.
  - Este nombre se puede poner de ***dos formas***:
    1. **Relativo al atributo "package"** que especificamos en el elemento <manifest>.

       En este caso, **sólo se pone el nombre de la clase precedido por un "."**

       Ejemplo:

       Pincha aquí

    2. **Con la ruta completa, independientemente de si se especificó o no el paquete en el atributo "package"** del elemento <manifest>.

       En este caso, se pone el **nombre de la clase precedido del nombre del paquete** que contiene la clase

       Ejemplo:

       Pincha aquí

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" android:debuggable="true">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" android:debuggable="true">
        <activity
            android:name="com.ejemplos.miprimeraaplicacion.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Describir los componentes de la aplicación (V): Atributos de la etiqueta <activity> (II)

- android:**label**="…"
  - "**label**" especifica el **string que va a aparecer en la barra de título** de la activity.
  - La **activty puede no tener barra de título.**
  - **Si no hay barra de título, el string no se ve.**
  - Si definimos esta **activity como punto de entrada** para nuestra **aplicación**, entonces:
    - También se mostrará este **texto en el lanzador** de nuestra aplicación en el teléfono.
    - **Si se omite este atributo, el texto que se vea en** el **lanzador** de nuestra aplicación en el teléfono, será el del **atributo "label" de la etiqueta "application".**

Describir los componentes de la aplicación (VI): Atributos de la etiqueta <activity> (III)

- android:**screenOrientation**="…"
  - "**screenOrientation**" especifica la **orientación que se va a mostrar en la pantalla del dispositivo** cuando se ejecute esta activity.
  - Puede tomar todos estos **valores**: (*ver*)
  - **Los más usados al principio son:**
    - "**portrait**" → Vertical
    - "**landscape**" → Horizontal
  - ***Ejemplo***:

```
android:screenOrientation="portrait"
```

Describir los componentes de la aplicación (VII): Atributos de la etiqueta <activity>: screenOrientation

- Si no se pone este atributo, la activity tomará la orientación según el sensor del dispositivo: **EL DISPOSITIVO ESCOGE LA ORIENTACIÓN.**
- ***PROBLEMA***:
  - Los dispositivos Android que poseen un **acelerómetro** pueden **determinar cuando están en posición vertical** (portrait) **u horizontal** (landscape).
  - La **acción por defecto** es que la **aplicación rote la vista** de acuerdo a su orientación.
  - **Android está configurado para que cada vez que la vista cambia de un tipo a otro, se destruye una Activity y se crea una nueva *(se reinicia),* por lo que puede perderse el estado actual de la actividad.**
  - Esto implica que **si el dispositivo cambia de orientación**, nuestra **activity** será **destruida y restaurada.**
  - **Esto en juegos no puede permitirse.**
  - Por tanto, **siempre debería especificarse este atributo, al menos con "portrait" o "landscape".**

Capacidades de los componentes y cómo se declaran

Mª Alejandra Tomás Fernández

Intent-filters (I)

- El sistema identifica **qué componentes pueden responder a un intent** comprobando los intent-filters en los AndroidManifest.xml del resto de componentes y aplicaciones.

- Cuando se declara un componente, **opcionalmente** se puede incluir una etiqueta <intent-filter> **en donde se especificarán las capacidades del componente**, para que Android pueda decidir si utilizarlo para responder a determinados Intents lanzados por otras aplicaciones.

Intent filters (II)

- Un **ejemplo** sería una aplicación de e-mail:
  - Esta aplicación (*que no es nuestra, sino que ya existe*) tiene una actividad para redactar un nuevo email.
  - Este **actividad** de esa aplicación puede haber declarado un **intent-filter** en su manifest que dice que **responde a Intents** de tipo "**send**" (enviar).
  - Una aplicación nuestra puede crear un **Intent con una acción "send"** (ACTION_SEND)
  - Cuando invocamos el intent con startActivity, se utilizará la actividad de crear e-mails de la otra aplicación.

Intent Filters (3)

- Otro ejemplo: hacer una búsqueda en la web
  - Lanzamos un intent con acción ACTION_WEB_SEARCH.
  - Android busca qué aplicaciones son las que reciben este intent buscando entre los intent filters de sus componentes.
  - Se abre una de esas aplicaciones, normalmente el navegador de Android. Si hay más de una aparece un menú para que el usuario escoja.

Ejemplo de la sintaxis

Un *ejemplo de intent filter* que registra a la activity con capacidad de abrir un hipotético navegador:

```
<activity android:name=".BrowserActivitiy"
          android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="http"/>
    </intent-filter>
</activity>
```

Ejemplo de la sintaxis

Otro ejemplo de intent filter que registra a la activity con capacidad de enviar por email sólo texto plano que se le pase desde el intent:

```
<activity android:name=".ActivityTest"
          android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>
```