

Samsung **TECH INSTITUTE**

Curso UPM / Samsung
DESARROLLO DE APPS
PARA ANDROID Y WEB

¡Desarrolla tu futuro!



SAMSUNG

POLITÉCNICA
"Ingeniamos el futuro"

DESARROLLO DE
APLICACIONES EN ANDROID



SAMSUNG

Abraham Gutiérrez Rodríguez

Samsung **TECH INSTITUTE**

Android: Recursos Proyectos Android

- Estructura:
 - src: Contiene los archivos con el código auxiliar de la aplicación.
 - bin: Directorio de salida de la compilación. Aquí es donde se puede encontrar el archivo .apk y otros recursos compilados.
 - jni: Contiene fuentes de código nativo desarrollados utilizando el NDK Android.
 - gen: Contiene los archivos Java generados por el ADT, como el archivo R.java y los interfaces creados a partir de archivos AIDL.
 - assets: Inicialmente está vacía. Se puede utilizar para almacenar archivos de contenido raw. Los archivos que se guardan aquí se compilan en el archivo .apk tal y como aparecen, el nombre de archivo original se preserva.
 - Es una buena ubicación para las texturas y los datos del juego.
 - res: Contiene recursos de la aplicación, como los archivos gráficos, los archivos de diseño, y los valores de texto.



Android: Recursos Proyectos Android

- Estructura:
 - libs: Contiene las bibliotecas necesarias para el proyecto.
 - AndroidManifest.xml: Es el archivo de control que describe la naturaleza de la aplicación y cada uno de sus componentes.
 - Por ejemplo, describe lo siguiente: características de las actividades, servicios, receptores de intención, y los proveedores de contenidos; los permisos que se solicitan; bibliotecas externas necesarias; qué características de los dispositivos se requieren, qué niveles API se requieren o son soportadas; y otros.
 - project.properties: Este archivo contiene elementos de la configuración del proyecto, como el destino de generación. Este archivo es parte integral del proyecto, por lo que se debe mantener en un sistema de control de revisiones
 - local.properties: Propiedades específicas del equipo personalizables para el sistema de construcción. Si utiliza Ant para construir el proyecto, contiene la ruta de acceso a la instalación del SDK. Debido a que el contenido del archivo es específico de la instalación local del SDK, los local.properties no deben ser mantenidos en un sistema de control de revisiones. Si utiliza Eclipse, no se utiliza este archivo.
 - ant.properties: Propiedades personalizables para el sistema de construcción. Puede editar este archivo para anular la configuración de generación predeterminados utilizados por Ant y también proporcionar la ubicación del almacén de claves y alias de clave de forma que las herramientas de construcción pueden firmar la solicitud en la construcción en modo de lanzamiento. Este archivo es parte integral del proyecto, por lo que se debe mantener en un sistema de control de revisiones. Si utiliza Eclipse, no se utiliza este archivo.
 - build.xml: El fichero de construcción Ant para el proyecto. Esto sólo es aplicable a los proyectos que utilizan Ant.



Android: Recursos Proporcionando Recursos

- **Externalizar recursos:**
 - Siempre hay que externalizar los recursos de la aplicación, como imágenes y cadenas de texto del código, de modo que se puedan mantener de forma independiente.
 - También hay que proporcionar recursos alternativos para configuraciones de dispositivo específicos, agrupándolos en directorios de recursos con nombres especiales. En tiempo de ejecución, Android utiliza el recurso adecuado en función de la configuración actual.
 - Por ejemplo, es posible que se desee proporcionar un diseño de interfaz de usuario diferente en función del tamaño de la pantalla o dependiendo de la configuración de idioma.
 - Una vez externalizados los recursos de la aplicación, se puede acceder a ellos mediante identificadores de recursos ID's que se generan en la clase R del proyecto.



Android: Recursos Proporcionando Recursos

- **Agrupando recursos:**
 - **res:** Contiene recursos de la aplicación, como los archivos gráficos, los archivos de diseño, y los valores de texto.
 - **anim:** Para los archivos XML que se compilan en objetos de animación.
 - **color:** Para los archivos XML que describen colores.
 - **drawable:** Para los archivos de mapa de bits (PNG, JPEG, GIF o), archivos de imagen 9-Patch, y los archivos XML que describen formas Drawable u objetos Drawable que contienen varios estados (normal, presionado, o el foco).
 - **layout:** Archivos XML que se compilan en presentaciones de pantalla (o parte de una pantalla).
 - **menu:** Para los archivos XML que definen los menús de la aplicación.
 - **raw:** Para los archivos de assets (activos) con contenido arbitrario. Guardar archivos de activos aquí en lugar de en el directorio assets/ sólo se diferencia en la forma en que se accede a ellos. Estos archivos son procesados por AAPT y deben ser referenciados desde la aplicación utilizando un identificador de recursos en la clase R.
 - Por ejemplo, este es un buen lugar para que los medios de comunicación, tales como MP3 u Ogg.
 - **values:** Para los archivos XML que se compilan para diferentes tipos de recursos. A diferencia de otros recursos en el directorio res/, los recursos definidos en archivos XML en esta carpeta no se referencian por el nombre del archivo. En cambio, forman parte de la clase R.
 - **xml:** Para los archivos XML que configuran diversos componentes de la aplicación.



Android: Recursos Conceptos previos

- Recursos de diseño y gráficos:

	Low density (120), <i>ldpi</i>	Medium density (160), <i>mdpi</i>	High density (240), <i>hdpi</i>	Extra high density (320), <i>xhdpi</i>
Small screen	QVGA (240x320)		480x640	
Normal screen	WQVGA400 (240x400) WQVGA (240x432)	HVGA (320x480)	WVGA (480x800) WVGA854 (480x854) 600x1024	640x960
Large screen	WVGA800 (480x800) WVGA854 (480x854)	WVGA800 (480x800) WVGA854 (480x854) 600x1024		
Extra large screen	1024x600	WXGA (1280x800) 1024x768 1280x768	1536x1152 1920x1152 1920x1200	2048x1536 2560x1536 2560x1600



Android: Recursos Proporcionando Recursos

- Recursos de diseño y gráficos:

- `/res/layout`

- `/res/layout` – diseños para pantallas de tamaño normal o por defecto
- `/res/layout-small` - diseños para pantallas de tamaño pequeño
- `/res/layout-large` - diseños para pantallas de tamaño grande
- `/res/layout-xlarge` - diseños para pantallas de tamaño extra-grande
- `/res/layout-xlarge-land` - diseños para pantallas de tamaño extra-grande y orientación apaisada
- `/res/layout-sw600dp` – diseños para tabletas de 7" (600dp de ancho como mínimo)
- `/res/layout-sw720dp` diseños para tabletas de 10" (720dp de ancho como mínimo)
- `/res/layout-w600dp` – diseños multipanel (cualquier pantalla con 600dp de ancho como mínimo)
- `/res/layout-h720dp` – diseños multipanel (cualquier pantalla con 720dp de alto como mínimo)



Android: Recursos Proporcionando Recursos

- Recursos de diseño y gráficos:
 - /res/drawable
 - /res/drawable-mdpi - bitmap de densidad media
 - /res/drawable-hdpi - bitmap de densidad alta
 - /res/drawable-xhdpi - bitmap de densidad extra-alta
 - /res/drawable-nodpi - bitmap sin preescalado
 - ...
 - /res/drawable-small-land
 - /res/drawable-ja/



Android: Recursos Proporcionando Recursos

- Recursos de localización:
 - Una aplicación Android se puede ejecutar en muchos dispositivos y en muchas regiones. Para llegar a más usuarios, la aplicación debe manejar texto, archivos de audio, números, moneda y gráficos de manera apropiada a los lugares donde se utilizará la aplicación.
 - Los recursos son cadenas de texto, diseños, sonidos, gráficos y cualquier otro dato estático que necesita la aplicación.
 - Una aplicación puede incluir múltiples conjuntos de recursos, cada uno personalizado para una configuración de dispositivo diferente.
 - Cuando un usuario ejecuta la aplicación, Android selecciona automáticamente y carga los recursos que mejor se adapten al dispositivo.



Android: Recursos Proporcionando Recursos

- Recursos de localización:
 - Es una buena práctica utilizar el marco de recursos Android para separar los aspectos localizados de la aplicación en la medida de lo posible de la funcionalidad básica proporcionada por Java.
 - Para crear recursos localizados, se colocan los archivos en subdirectorios con nombres especiales dentro del directorio `/res` del proyecto.
 - Por ejemplo directorios adicionales `values` que incluyen un guion y el código ISO del idioma al final del nombre del directorio.



Android: Recursos Accediendo a los Recursos

- Identificadores de recursos:
 - Una vez que proporcionados los recursos a la aplicación, se puede haciendo referencia a ellos mediante su ID de recurso. Todos los identificadores de recursos están definidos en la clase `R` del proyecto, que la herramienta `AAPT` genera automáticamente.
 - Cuando se compila la aplicación, `AAPT` genera la clase `R`, que contiene identificadores de recursos para todos los recursos disponibles en el directorio `res/`.
 - Para cada tipo de recurso, hay una subclase de `R` (por ejemplo, `R.drawable` para todos los recursos de tipo dibujable), y para cada recurso de ese tipo, hay un número entero estático (por ejemplo, `R.drawable.icon`). Este entero es el identificador de recursos que puede utilizar para recuperar el recurso.
 - Aunque la clase `R` es donde se especifican los ID de los recursos de, nunca hay que editar esta clase en busca del identificador de un recurso. Un identificador de recursos siempre se compone de:
 - El **tipo de recurso**: cada recurso se agrupa en un "tipo", tales como string, drawable, o layout.
 - El **nombre del recurso**: que puede ser el nombre del archivo, con exclusión de la extensión; o el valor del atributo `android:name` fijado en el documento XML que lo contiene, si el recurso es un valor sencillo (como una cadena).



Android: Recursos Accediendo a los Recursos

- Modos de acceso a los recursos:
 - Hay dos formas de acceder a un recurso:
 - **Desde el código:** Utilizando un número entero estático definido en una subclase de la clase `R`, como una sintaxis:

```
[<package_name>].R.<resource_type>.<resource_name>
```

Hay muchas APIs de Android que pueden acceder a los recursos cuando se proporciona un identificador de recurso en este formato.
 - **Desde XML:** Utilizando una sintaxis especial de XML que también se corresponde con el ID de recurso definido en la clase `R`, como una sintaxis:

```
@[<package_name>:]<resource_type>/<resource_name>
```

Puede utilizarse esta sintaxis en cualquier lugar de un documento XML en donde se espera un valor de un recurso.
 - `R.string.hola` vs `@string/hola` dónde `string` es el tipo de recurso y `hola` es el nombre del recurso.



Android: Recursos *AndroidManifest.xml*

- Cada aplicación debe tener un archivo de manifiesto `AndroidManifest.xml` (precisamente con ese nombre) en su directorio raíz. El archivo de manifiesto presenta información esencial acerca de una aplicación para el sistema Android, la información que el sistema debe tener antes de que pueda ejecutar cualquiera de código de la aplicación.
- Entre otras cosas, el manifiesto hace lo siguiente:
 - Da nombre al paquete Java para la aplicación. El nombre del paquete sirve como identificador único para la aplicación.
 - En él se describen los componentes de la aplicación: las actividades, los servicios, los receptores de difusión y proveedores de contenido. Se nombran las clases que implementan cada uno de los componentes y publica sus capacidades (por ejemplo, qué mensajes se deben manejar). Estas declaraciones permiten que el sistema Android sepa cuáles son los componentes y bajo qué condiciones pueden ser lanzados .

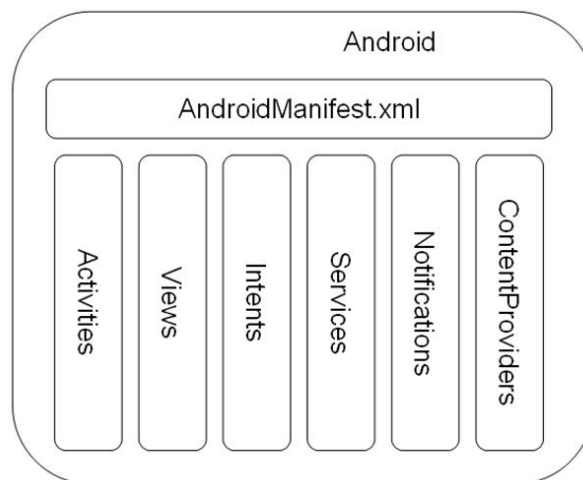


Android: Recursos *AndroidManifest.xml*

- Entre otras cosas, el manifiesto hace lo siguiente:
 - Determina qué procesos serán los anfitriones de los componentes de la aplicación.
 - Declara que los permisos debe tener la aplicación para poder acceder a las partes protegidas de la API y para interactuar con otras aplicaciones.
 - También declara qué permisos requieren otros componentes para poder interactuar con los componentes de la aplicación .
 - Enumera las clases de instrumentación que proporcionan perfiles y otras información mientras la aplicación se está ejecutando. Estas declaraciones están presentes en sólo el manifiesto mientras la aplicación está siendo desarrollada y probada; se quitan antes de la publicación de la solicitud.
 - Declara el nivel mínimo de la API de Android que requiere la aplicación.
 - Enumera las bibliotecas con las que debe ser enlazada la aplicación.



Android: Recursos *AndroidManifest.xml*



Android: Recursos

- Estructura del fichero de manifiesto:

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

<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

    </application>

</manifest>
```

1

2



Android: Recursos *AndroidManifest.xml*

- Estructura del fichero de manifiesto:

```
<service>
    <intent-filter> . . . </intent-filter>
    <meta-data/>
</service>

<receiver>
    <intent-filter> . . . </intent-filter>
    <meta-data />
</receiver>

<provider>
    <grant-uri-permission />
    <meta-data />
    <path-permission />
</provider>

<uses-library />

</application>

</manifest>
```

3

4



Android: Recursos *AndroidManifest.xml*

- Estructura del fichero de manifiesto:

- | | |
|--------------------------|------------------------|
| – <action> | – <permission> |
| – <activity> | – <permission-group> |
| – <activity-alias> | – <permission-tree> |
| – <application> | – <provider> |
| – <category> | – <receiver> |
| – <data> | – <service> |
| – <grant-uri-permission> | – <supports-screens> |
| – <instrumentation> | – <uses-configuration> |
| – <intent-filter> | – <uses-feature> |
| – <manifest> | – <uses-library> |
| – <meta-data> | – <uses-permission> |
| | – <uses-sdk> |



Android: Recursos *AndroidManifest.xml*

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.jsupport"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".CheckConnectionActivity"
            android:label="@string/title_activity_check_connection" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

