

www.centic.es

Curso Android Edición 2011



CENTRO TECNOLÓGICO DE LAS TECNOLOGÍAS DE
LA INFORMACIÓN Y LAS COMUNICACIONES



centic

Android Manifest

Contenido.

- Nombre de los paquetes de la aplicación. Es usado como identificador de la aplicación.
- Describe todos los componentes de la aplicación. Actividades, Servicios, Intents, Proveedores de contenido, etc..
- Determina que acciones puede llevar a cabo la aplicación.
- Lista los permisos que la aplicación necesita para acceder a APIs protegidas.
- Declara los permisos que otras aplicaciones deben tener para acceder a funcionalidades de nuestra aplicación.
- Recopila las clases de Instrumentación necesarias para realizar el testing.
- Especifica el SDK mínimo para el que está destinada la aplicación.
- Muestra las librerías de las que hace uso la aplicación.



Android Manifest

Estructura.

<manifest>

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



Android Manifest

Estructura.

```
<application>
```

```
    <activity>
```

```
        <intent-filter>
```

```
            <action />
```

```
            <category />
```

```
            <data />
```

```
        </intent-filter>
```

```
        <meta-data />
```

```
    </activity>
```

```
    <activity-alias>
```

```
        <intent-filter> . . . </intent-filter>
```

```
        <meta-data />
```

```
    </activity-alias>
```

www.centic.es

Android Manifest

Estructura.

```
<service>  
    <intent-filter> . . . </intent-filter>  
    <meta-data />  
</service>  
  
<receiver>  
    <intent-filter> . . . </intent-filter>  
    <meta-data />  
</receiver>  
  
<provider>  
    <grant-uri-permission />  
    <meta-data />  
</provider>
```



Android Manifest

Estructura.

```
<provider>  
    <grant-uri-permission />  
    <meta-data />  
</provider>  
  
<uses-library />
```



Android Manifest

Estructura.

```
<application android:allowTaskReparenting=["true" | "false"]
    android:backupAgent="string"
    android:debuggable=["true" | "false"]
    android:description="string resource"
    android:enabled=["true" | "false"]
    android:hasCode=["true" | "false"]
    android:hardwareAccelerated=["true" | "false"]
    android:icon="drawable resource"
    android:killAfterRestore=["true" | "false"]
    android:label="string resource"
    android:logo="drawable resource"
    android:manageSpaceActivity="string"
    android:name="string"
    android:permission="string"
    android:persistent=["true" | "false"]
    android:process="string"
    android:restoreAnyVersion=["true" | "false"]
    android:taskAffinity="string"
    android:theme="resource or theme" >

    . . .
</application>
```



Android Manifest

Estructura.

```
<activity android:allowTaskReparenting=["true" | "false"]
    android:alwaysRetainTaskState=["true" | "false"]
    android:clearTaskOnLaunch=["true" | "false"]
    android:configChanges=["mcc", "mnc", "locale",
        "touchscreen", "keyboard", "keyboardHidden",
        "navigation", "orientation", "screenLayout",
        "fontScale", "uiMode"]
```

```
    android:enabled=["true" | "false"]
    android:excludeFromRecents=["true" | "false"]
    android:exported=["true" | "false"]
    android:finishOnTaskLaunch=["true" | "false"]
    android:hardwareAccelerated=["true" | "false"]
    android:icon="drawable resource"
    android:label="string resource"
    android:launchMode=["multiple" | "singleTop" |
        "singleTask" | "singleInstance"]
```



Android Manifest

Estructura.

```
android:multiprocess=["true" | "false"]
android:name="string"
android:noHistory=["true" | "false"]
android:permission="string"
android:process="string"
android:screenOrientation=["unspecified" | "user" | "behind" |
                             "landscape" | "portrait" |
                             "sensor" | "nosensor"]

android:stateNotNeeded=["true" | "false"]
android:taskAffinity="string"
android:theme="resource or theme"
android:windowSoftInputMode=["stateUnspecified",
                              "stateUnchanged", "stateHidden",
                              "stateAlwaysHidden", "stateVisible",
                              "stateAlwaysVisible", "adjustUnspecified",
                              "adjustResize", "adjustPan"] >
```



Android Manifest

Estructura.

```
<intent-filter android:icon="drawable resource"  
               android:label="string resource"  
               android:priority="integer" >
```

```
<action android:name="string" />
```

```
<category android:name="string" />
```

```
</intent-filter>
```



Android Manifest

Estructura.

```
<activity-alias android:enabled=["true" | "false"]
  android:exported=["true" | "false"]
  android:icon="drawable resource"
  android:label="string resource"
  android:name="string"
  android:permission="string"
  android:targetActivity="string" >
  . . .
</activity-alias>
```



Android Manifest

Estructura.

```
<service android:enabled=["true" | "false"]  
         android:exported=["true" | "false"]  
         android:icon="drawable resource"  
         android:label="string resource"  
         android:name="string"  
         android:permission="string"  
         android:process="string" >  
    . . .  
</service>
```



Android Manifest

Estructura.

```
<intent-filter android:icon="drawable resource"  
               android:label="string resource"  
               android:priority="integer" >
```

```
<action android:name="string" />
```

```
<category android:name="string" />
```

```
</intent-filter>
```



Android Manifest

Estructura.

```
<receiver android:enabled=["true" | "false"]  
          android:exported=["true" | "false"]  
          android:icon="drawable resource"  
          android:label="string resource"  
          android:name="string"  
          android:permission="string"  
          android:process="string" >  
    .  
    .  
    .  
</receiver>
```



Android Manifest

Estructura.

```
<intent-filter android:icon="drawable resource"  
               android:label="string resource"  
               android:priority="integer" >
```

```
<action android:name="string" />
```

```
<category android:name="string" />
```

```
</intent-filter>
```



Android Manifest

Estructura.

```
<provider android:authorities="list"
  android:enabled=["true" | "false"]
  android:exported=["true" | "false"]
  android:grantUriPermissions=["true" | "false"]
  android:icon="drawable resource"
  android:initOrder="integer"
  android:label="string resource"
  android:multiprocess=["true" | "false"]
  android:name="string"
  android:permission="string"
  android:process="string"
  android:readPermission="string"
  android:syncable=["true" | "false"]
  android:writePermission="string" >
  . . .
</provider>
```



Android Manifest

Estructura.

```
<meta-data android:name="string"
            android:resource="resource specification"
            android:value="string" />
```

```
<meta-data android:name="zoo" android:value="@string/kangaroo" />
```

```
<meta-data android:name="zoo" android:resource="@string/kangaroo" />
```

```
<grant-uri-permission android:path="string"
                     android:pathPattern="string"
                     android:pathPrefix="string" />
```



Android Manifest

Estructura.

www.centic.es

```
<uses-configuration android:reqFiveWayNav=["true" | "false"]
    android:reqHardKeyboard=["true" | "false"]
    android:reqKeyboardType=["undefined" | "nokeys" | "qwerty" |
        "twelvekey"]
    android:reqNavigation=["undefined" | "nonav" | "dpad" |
        "trackball" | "wheel"]
    android:reqTouchScreen=["undefined" | "notouch" | "stylus" |
        "finger"] />
```

```
<uses-configuration android:reqFiveWayNav="true" android:reqTouchScreen="finger"
    android:reqKeyboardType="qwerty" />
<uses-configuration android:reqFiveWayNav="true" android:reqTouchScreen="finger"
    android:reqKeyboardType="twelvekey" />
```



Android Manifest

Estructura.

www.centic.es

```
<uses-feature android:name="string"  
              android:required=["true" | "false"]  
              android:glEsVersion="integer" />
```

```
<uses-feature android:name="android.hardware.bluetooth" />  
<uses-feature android:name="android.hardware.camera" />
```



Android Manifest

Estructura.

```
<supports-screens android:resizeable=["true" | "false"]
                  android:smallScreens=["true" | "false"]
                  android:normalScreens=["true" | "false"]
                  android:largeScreens=["true" | "false"]
                  android:xlargeScreens=["true" | "false"]
                  android:anyDensity=["true" | "false"] />
```

```
<compatible-screens>
    <screen android:screenSize=["small" | "normal" | "large" | "xlarge"]
            android:screenDensity=["ldpi" | "mdpi" | "hdpi" | "xhdpi"] />
    ...
</compatible-screens>
```



Android Manifest

Estructura.

```
<uses-library android:name="string"  
              android:required=["true" | "false"] />
```

```
<uses-permission android:name="string" />
```

```
android.permission.CALL_EMERGENCY_NUMBERS  
android.permission.READ_OWNER_DATA  
android.permission.SET_WALLPAPER  
android.permission.DEVICE_POWER
```



Android Manifest

Estructura.

```
<uses-sdk android:minSdkVersion="integer"  
          android:targetSdkVersion="integer"  
          android:maxSdkVersion="integer" />
```

Platform Version	API Level
Android 3.1	12
Android 3.0	11
Android 2.3.4	10
Android 2.3.3	
Android 2.3	9
Android 2.2	8
Android 2.1	7
Android 2.0.1	6
Android 2.0	5
Android 1.6	4
Android 1.5	3
Android 1.1	2
Android 1.0	1



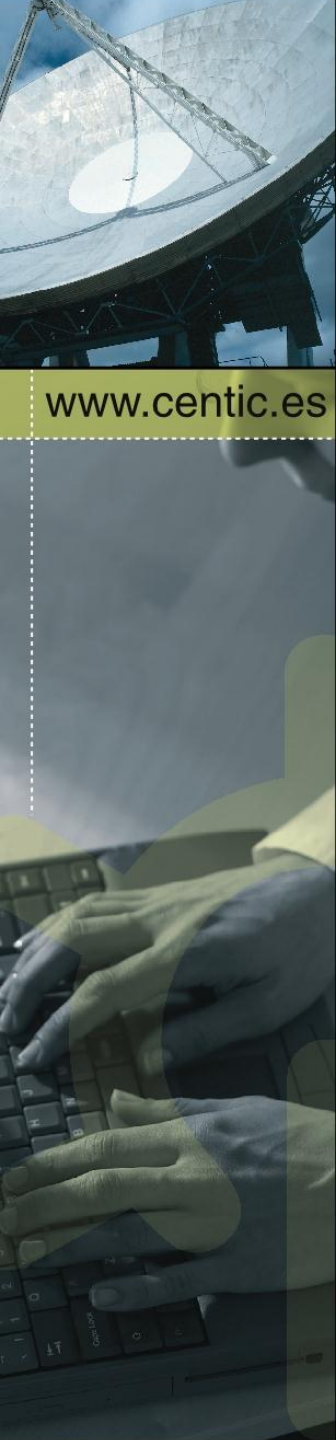
Application

Funcionalidad.

- Es la primera clase que se crea al lanzar la aplicación.
- Su vida está ligada a la ejecución de la aplicación.
- Accesible desde cualquier Activity de la aplicación.

Eventos.

- onCreate()
- onLowMemory()
- onTerminate()
- onConfigurationChanged(Configuration newConfig)



Application

Descripción.

- Es una pantalla con la que el usuario puede interactuar.
- Cada Activity proporciona su propia interfaz de usuario.
- Llamamos aplicación al conjunto de Activities.
- Una vez terminadas se almacenan en la «Pila del sistema»

Activities

Descripción.

- Cada Activity tiene su propio ciclo de vida, independiente del resto de Activities de la aplicación.
- Normalmente debe existir una Aplicación «MAIN», donde inicie la aplicación.
 - Excepciones:
 - Aplicaciones sin GUI
 - Servicios
 - Widgets
 - Otros

Activities

Crear una Activity.

- Clase heredada de Activity
- Implementar los metodos involucrados en su ciclo de vida
- Implementar onCreate()
- Unir la Activity a una vista con el método setContentView()



Activities

Implementar GUI.

- Implementar la jerarquía de «Views»
 - Desde XML
 - Desde Código
- Unir la GUI con setContentView()

Declarar la Activity

```
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" />
    ...
  </application ... >
  ...
</manifest >
```



Activities

Asignar filtros a la Activity.

- ACTION
 - android.intent.action.MAIN
 - Personalizado
- CATEGORY
 - android.intent.category.LAUNCHER
 - android.intent.category.DEFAULT

Ejemplo.

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```



Activities

Lanzar la Activity.

- Sin Respuesta

```
Intent intent = new Intent(this, SignInActivity.class);  
startActivity(intent);
```

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_EMAIL, recipientArray);  
startActivity(intent);
```

Activities

Lanzar la Activity.

- Con Respuesta

```
private void pickContact() {  
    // Create an intent to "pick" a contact, as defined by the content provider URI  
    Intent intent = new Intent(Intent.ACTION_PICK, Contacts.CONTENT_URI);  
    startActivityForResult(intent, PICK_CONTACT_REQUEST);  
}  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    // If the request went well (OK) and the request was PICK_CONTACT_REQUEST  
    if (resultCode == Activity.RESULT_OK && requestCode == PICK_CONTACT_REQUEST) {  
        // Perform a query to the contact's content provider for the contact's name  
        Cursor cursor = getContentResolver().query(data.getData(),  
            new String[] {Contacts.DISPLAY_NAME}, null, null, null);  
        if (cursor.moveToFirst()) { // True if the cursor is not empty  
            int columnIndex = cursor.getColumnIndex(Contacts.DISPLAY_NAME);  
            String name = cursor.getString(columnIndex);  
            // Do something with the selected contact's name...  
        }  
    }  
}
```



Activities

Terminar una Activity

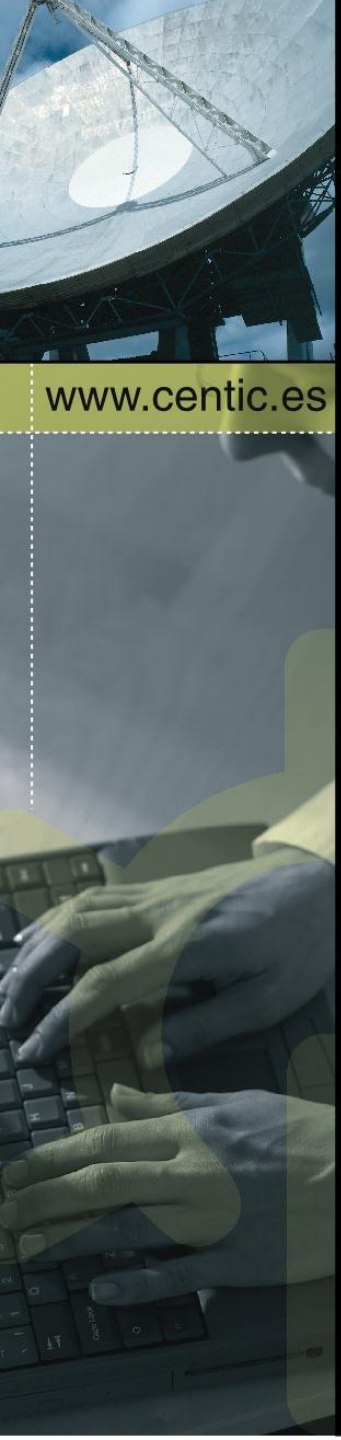
- Se encarga el SO
- En los casos de `startActivityForResult()`
 - `setResult()`
 - `RESULT_OK`
 - `RESULT_CANCELLED`
 - `Finish()`
 - `finishActivity()`

Activities

Ciclo de vida de una Activity. Estados.

- Resumed
 - La activity está en primer plano y con el foco
- Paused
 - No está en primer plano, es parcialmente visible y el foco lo tiene otra Activity
- Stopped
 - No es visible, sigue en memoria hasta que el sistema requiera de su espacio

Una actividad pausada o detenida puede ser destruida por el sistema en cualquier momento, si este requiere de sus recursos



Activities

Ciclo de vida de una Activity. Eventos.

- Son la transición entre sus diferentes estados.

```
public class ExampleActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // The activity is being created.  
    }  
    @Override  
    protected void onStart() {  
        super.onStart();  
        // The activity is about to become visible.  
    }  
    @Override  
    protected void onResume() {  
        super.onResume();  
        // The activity has become visible (it is now "resumed").  
    }  
}
```


Activities

Ciclo de vida de una Activity. Eventos.

- Siempre hay que ejecutar el método de la superclase primero.

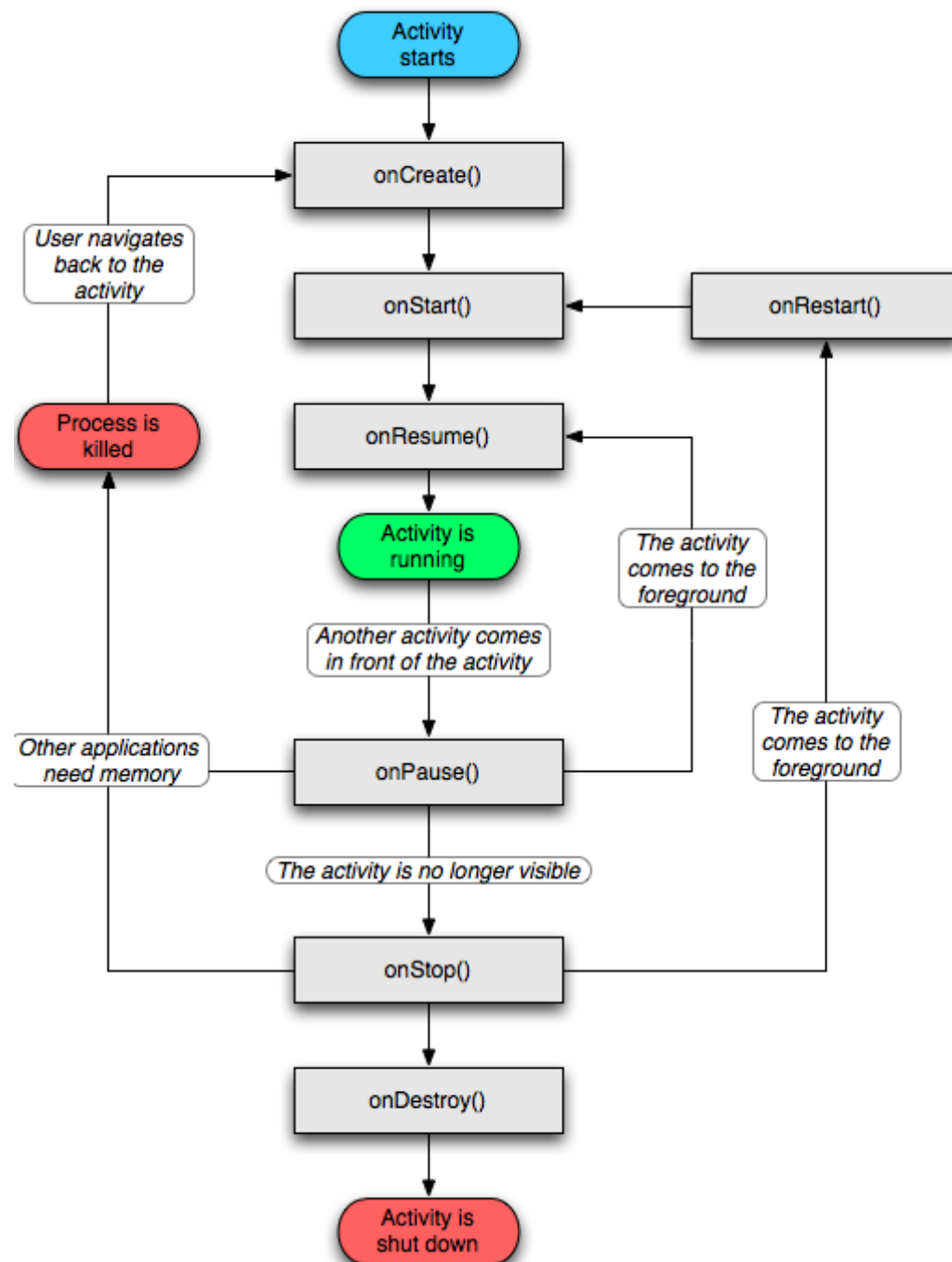
```
@Override
protected void onPause() {
    super.onPause();
    // Another activity is taking focus (this activity is about to be "paused").
}
@Override
protected void onStop() {
    super.onStop();
    // The activity is no longer visible (it is now "stopped")
}
@Override
protected void onDestroy() {
    super.onDestroy();
    // The activity is about to be destroyed.
}
}
```


Activities

Ciclo de vida de una Activity. Eventos.

- Ciclo de vida completo.
 - Desde onCreate hasta onDestroy()
 - Reservas realizadas en onCreate deben ser liberadas en onDestroy()
- Ciclo de vida visible
 - Desde onStart() hasta onStop()
 - Puede ser llamado multiples veces durante la ejecución de una Activity
- Ciclo de vida en primer plano.
 - Desde onResume() hasta onPause()
 - Es el que ocurre con mas frecuencia
 - Dialogos
 - Cambios de Activity
 - Menús





Method	Description	Killable after?	Next
<u>onCreate()</u>	Called when the activity is first created. This is where you should do all of your normal static set up — create views, bind data to lists, and so on. This method is passed a Bundle object containing the activity's previous state, if that state was captured (see <u>Saving Activity State</u> , later). Always followed by <code>onStart()</code> .	No	<code>onStart()</code>
<u>onRestart()</u>	Called after the activity has been stopped, just prior to it being started again. Always followed by <code>onStart()</code>	No	<code>onStart()</code>
<u>onStart()</u>	Called just before the activity becomes visible to the user. Followed by <code>onResume()</code> if the activity comes to the foreground, or <code>onStop()</code> if it becomes hidden.	No	<code>onResume()</code> or <code>onStop()</code>
<u>onResume()</u>	Called just before the activity starts interacting with the user. At this point the activity is at the top of the activity stack, with user input going to it. Always followed by <code>onPause()</code> .	No	<code>onPause()</code>
<u>onPause()</u>	Called when the system is about to start resuming another activity. This method is typically used to commit unsaved changes to persistent data, stop animations and other things that may be consuming CPU, and so on. It should do whatever it does very quickly, because the next activity will not be resumed until it returns. Followed either by <code>onResume()</code> if the activity returns back to the front, or by <code>onStop()</code> if it becomes invisible to the user.	Yes	<code>onResume()</code> or <code>onStop()</code>
<u>onStop()</u>	Called when the activity is no longer visible to the user. This may happen because it is being destroyed, or because another activity (either an existing one or a new one) has been resumed and is covering it. Followed either by <code>onRestart()</code> if the activity is coming back to interact with the user, or by <code>onDestroy()</code> if this activity is going away.	Yes	<code>onRestart()</code> or <code>onDestroy()</code>
<u>onDestroy()</u>	Called before the activity is destroyed. This is the final call that the activity will receive. It could be called either because the activity is finishing (someone called <code>finish()</code> on it), or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the <code>isFinishing()</code> method.	Yes	<i>nothing</i>

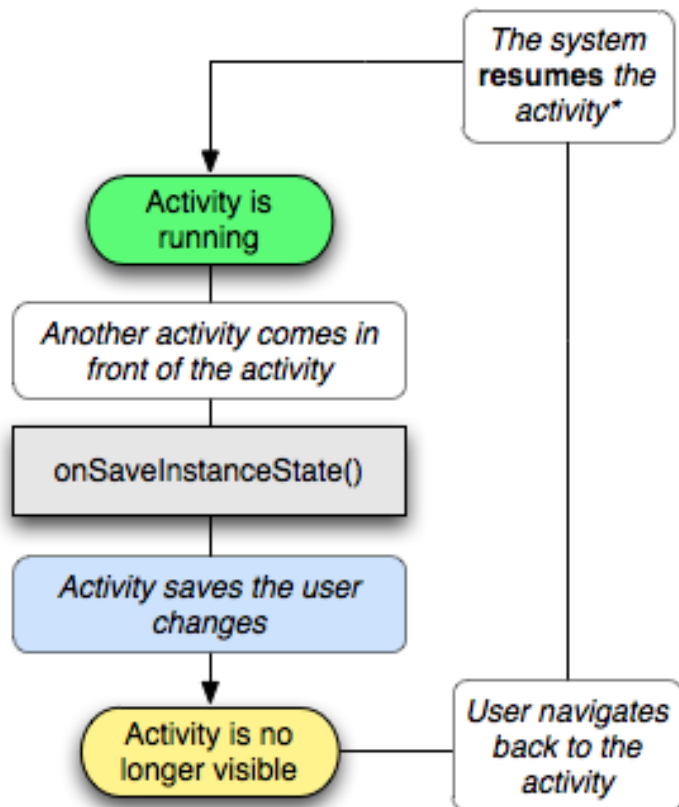


Activities

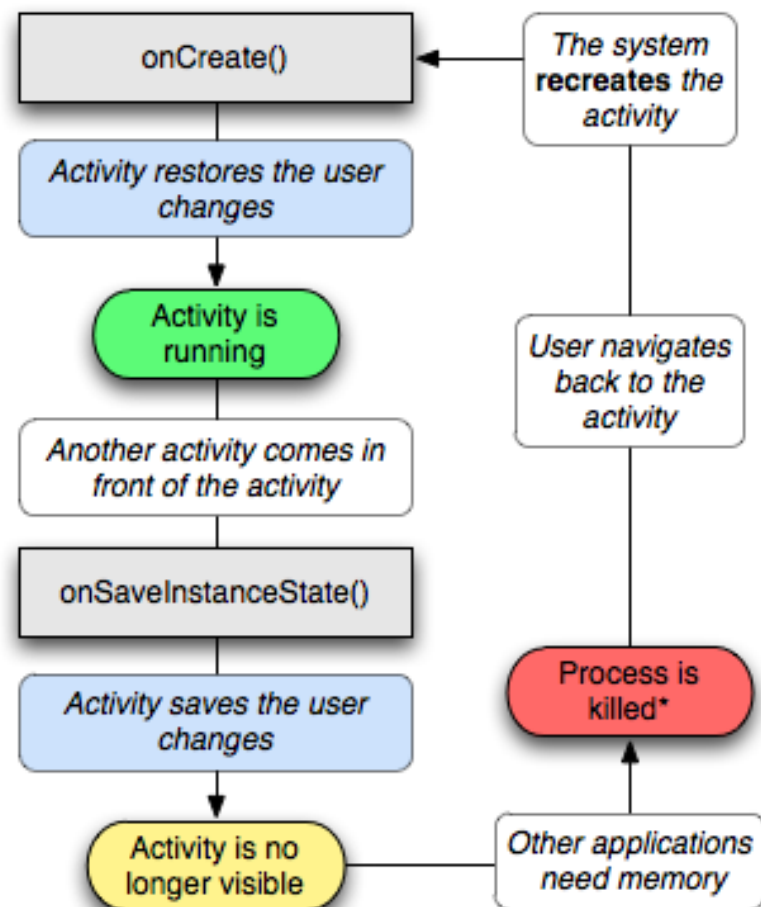
Ciclo de vida de una Activity. Guardando el estado.

- `onSaveInstanceState()`
 - Guardamos los cambios
- `onCreate()`
 - Restauramos los cambios
- `onRestoreInstanceState()`
 - En desuso
- `Bundle`
 - Almacén de datos
- El sistema no garantiza la llamada al método `onSaveInstanceState()`
- Los componentes guardan su estado, si estos tienen asignado un `android:id`, podemos cambiar este comportamiento con `android:saveEnabled=false`





* There's no need to restore state, because the activity is intact



* User changes are lost

Activities

Ciclo de vida de una Activity. Manejando cambios

- Cambios:
 - Orientación
 - Teclado
 - Idioma
 - Etc...
- Genera un `onDestroy()` y un `onCreate()`
- Usamos `onSaveInstanceState()`

Activities

Ciclo de vida de una Activity. Manejando los cambios.

- Dejamos el trabajo al SO:
 - Sobrecribir `onRetainNonConfigurationInstance()`

```
@Override
public Object onRetainNonConfigurationInstance() {
    final MyDataObject data = collectMyLoadedData();
    return data;
}
```

- En el `onCreate` recuperamos los datos guardados.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final MyDataObject data = (MyDataObject) getLastNonConfigurationInstance();
    if (data == null) {
        data = loadMyData();
    }
    ...
}
```



Activities

Ciclo de vida de una Activity. Manejando los cambios.

- Gestión manual:
 - Lo especificamos en el Manifest.xml

```
<activity android:name=".MyActivity"
          android:configChanges="orientation|keyboardHidden"
          android:label="@string/app_name">
```

- Implementamos onConfigurationChanged.

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Checks the orientation of the screen
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }

    // Checks whether a hardware keyboard is available
    if (newConfig.hardwareKeyboardHidden == Configuration.HARDKEYBOARDHIDDEN_NO) {
        Toast.makeText(this, "keyboard visible", Toast.LENGTH_SHORT).show();
    } else if (newConfig.hardwareKeyboardHidden == Configuration.HARDKEYBOARDHIDDEN_YES) {
        Toast.makeText(this, "keyboard hidden", Toast.LENGTH_SHORT).show();
    }
}
```



Activities

Sincronizando el baile de Activities.

Proceso de llamar desde la Activity «A» a la Activity «B»

- «A» ejecuta `startActivity(B)`
- «A» ejecuta `onPause()`
- «B» ejecuta `onCreate()`, `onStart()` y `onResume()`
- Si «A» no es visible ejecuta `onStop()`
- Si el sistema requiere memoria «A» que es «Killable» ejecuta `onDestroy()`



Activities

Conjunto de Activities «Task» y Pila de Activities «Back Stack».

- Las Activities se construyen alrededor de ACCIONES
- Nuestras Activities pueden llamar a Activities de otras aplicaciones
- Las Activities ejecutadas son almacenadas en la «Pila de Activities»
- La «Home» es la Activity de inicio del resto de Activities
- Al lanzar una aplicación se lanza su Activity con la acción MAIN y se crea su «Back Stack»



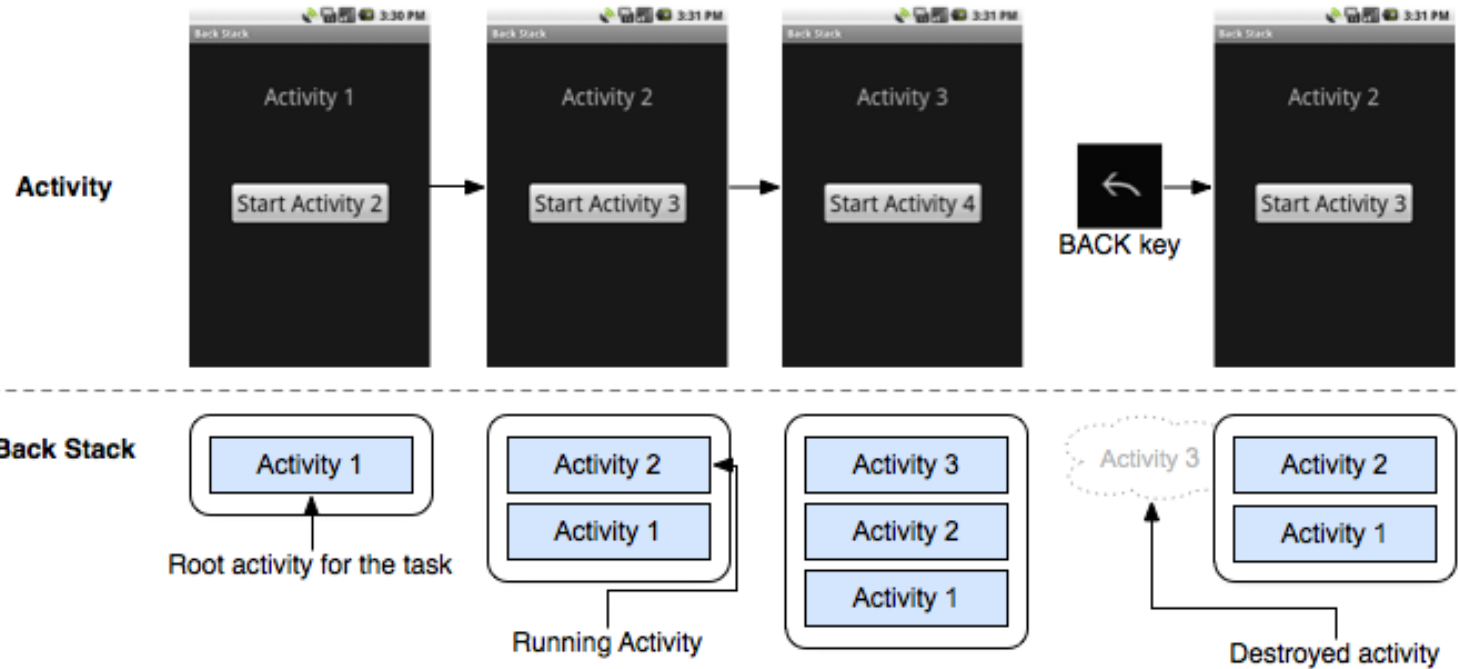
Activities

Conjunto de Activities «Task» y Pila de Activities «Back Stack».

- Cuando una Activity llama a otra, esta se pone en el «TOP» de la «Back Stack»
- La Activity que hace la llamada es detenida y se mantiene en la pila
- Cuando una Activity es detenida el sistema bloquea su GUI
- Al pulsar el botón «BACK» la Activity en primer plano es destruida y quitada de la pila, la activity previa es resumida
- Las Activities en la pila nunca reordenadas, se crean nuevas instancias de la Activity



Activities



Activities

Conjunto de Activities «Task» y Pila de Activities «Back Stack».

- Se crea una «TASK» por cada aplicación lanzada
- Podemos cambiar entre distintas TASK usando el boton HOME

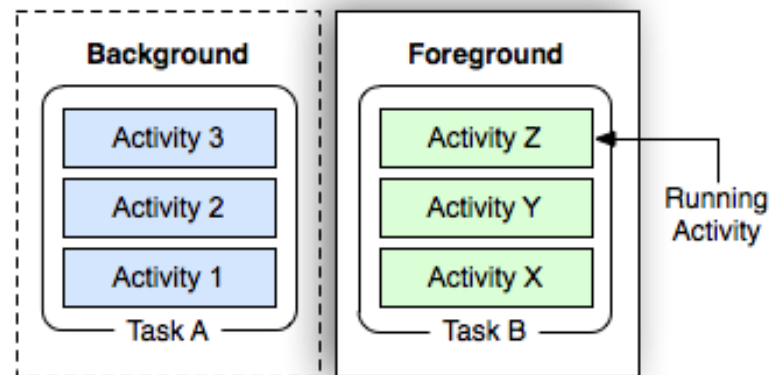
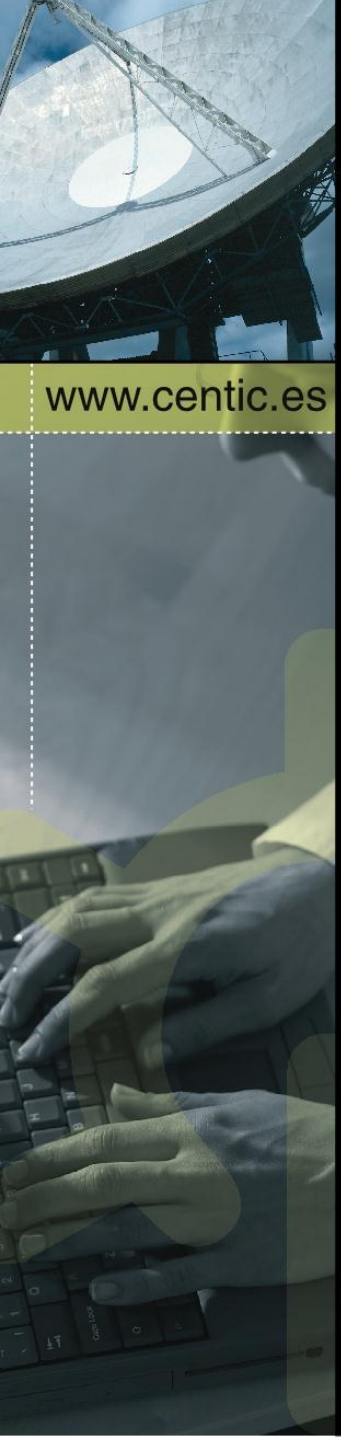


Figure 2. Two tasks: Task A is in the background, waiting to be resumed, while Task B receives user interaction in the foreground.

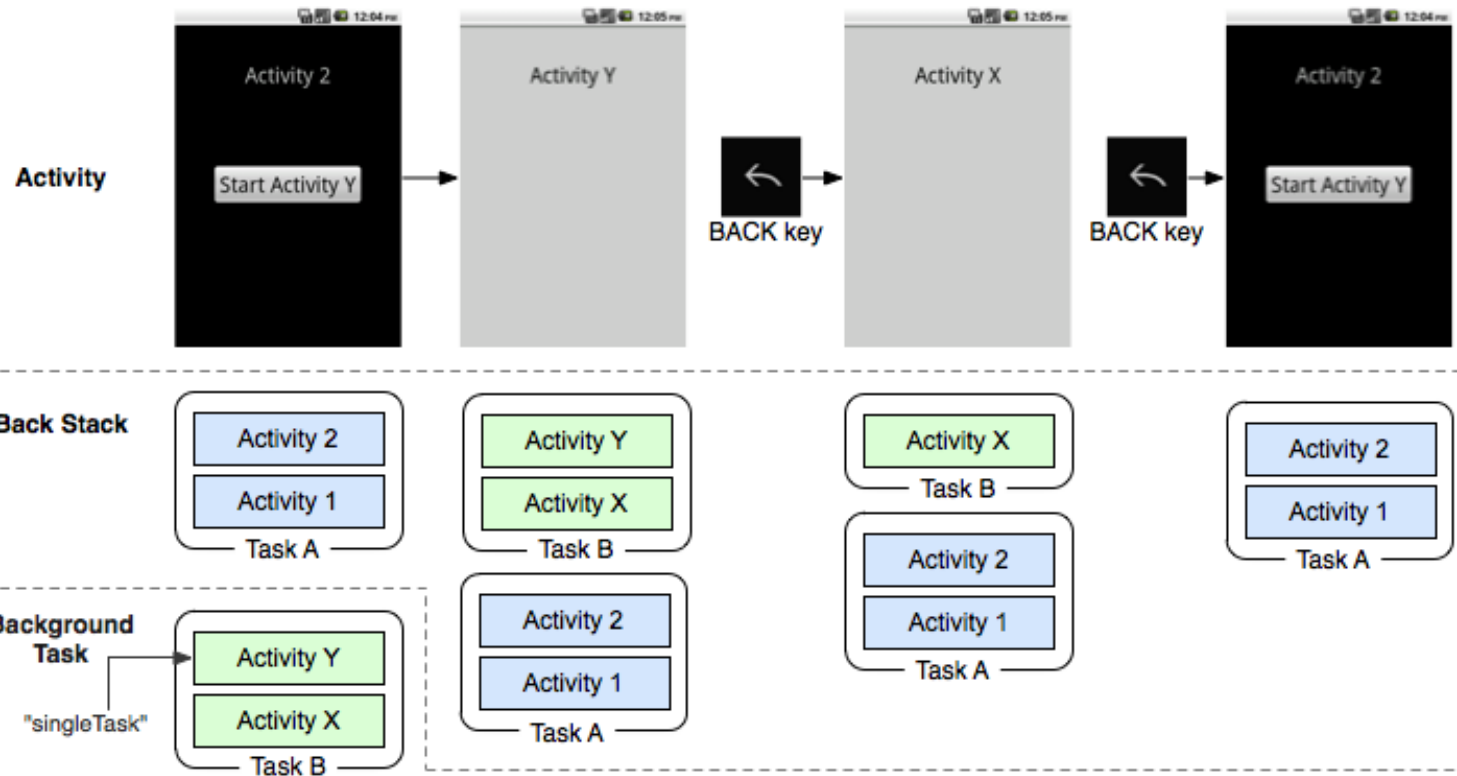
Activities

Modificando el comportamiento de la «TASK»

- Manifest.xml
 - launchMode
 - Standard
 - singleTop
 - singleTask
 - singleInstance
- Flags
 - startActivity()
 - FLAG_ACTIVITY_NEW_TASK
 - FLAG_ACTIVITY_SINGLE_TOP
 - FLAG_ACTIVITY_CLEAR_TOP



Activities



Recursos

Asignar recursos.

www.centic.es



Figure 1. Two different devices, both using default resources.

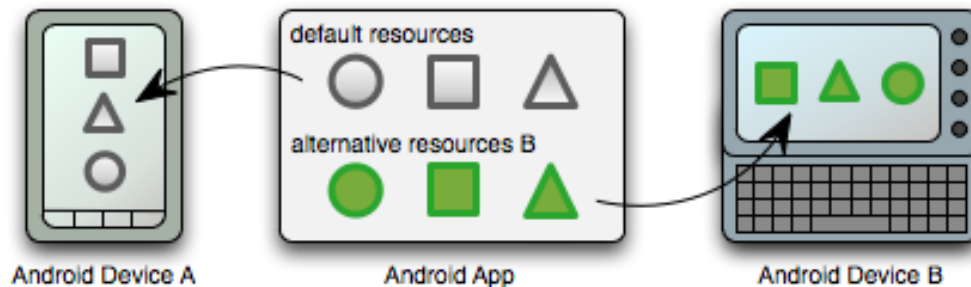


Figure 2. Two different devices, one using alternative resources.

Recursos

Agrupar recursos.

www.centic.es

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      icon.png  
    layout/  
      main.xml  
      info.xml  
    values/  
      strings.xml
```

Recursos

Tipo de recursos.

<code>anim/</code>	XML files that define tween animations. See Animation Resources .
<code>color/</code>	XML files that define a state list of colors. See Color State List Resource
<code>drawable/</code>	Bitmap files (<code>.png</code> , <code>.9.png</code> , <code>.jpg</code> , <code>.gif</code>) or XML files that are compiled into the following drawable resource subtypes: <ul style="list-style-type: none">• Bitmap files• Nine-Patches (re-sizable bitmaps)• State lists• Shapes• Animation drawables• Other drawables

Recursos

Tipo de recursos.

www.centic.es

<code>layout/</code>	XML files that define a user interface layout. See Layout Resource .
<code>menu/</code>	XML files that define application menus, such as an Options Menu, Context Menu, or Sub Menu. See Menu Resource .
<code>raw/</code>	<p>Arbitrary files to save in their raw form. To open these resources with a raw InputStream, call Resources.openRawResource() with the resource ID, which is <code>R.raw.filename</code>.</p> <p>However, if you need access to original file names and file hierarchy, you might consider saving some resources in the <code>assets/</code> directory (instead of <code>res/raw/</code>). Files in <code>assets/</code> are not given a resource ID, so you can read them only using AssetManager.</p>

Recursos

Tipo de recursos.

www.centic.es

values/

XML files that contain simple values, such as strings, integers, and colors.

Whereas XML resource files in other `res/` subdirectories define a single resource based on the XML filename, files in the `values/` directory describe multiple resources. For a file in this directory, each child of the `<resources>` element defines a single resource. For example, a `<string>` element creates an `R.string` resource and a `<color>` element creates an `R.color` resource.

Because each resource is defined with its own XML element, you can name the file whatever you want and place different resource types in one file. However, for clarity, you might want to place unique resource types in different files. For example, here are some filename conventions for resources you can create in this directory:

- arrays.xml for resource arrays ([typed arrays](#)).
- colors.xml for [color values](#)
- dimens.xml for [dimension values](#).
- strings.xml for [string values](#).
- styles.xml for [styles](#).

See [String Resources](#), [Style Resource](#), and [More Resource Types](#).

xml/

Arbitrary XML files that can be read at runtime by calling `Resources.getXML()`. Various XML configuration files must be saved here, such as a [searchable configuration](#).

Recursos

Recursos alternativos.

- Crear un directorio dentro de res/ con el patrón «tipo_recurso-calificador»
 - tipo_recurso: directorio asociado al tipo de recurso
 - Calificador: configuración para la que se aplica dicho recurso

```
res/  
    drawable/  
        icon.png  
        background.png  
    drawable-hdpi/  
        icon.png  
        background.png
```



Recursos

Recursos alternativos. Calificadores.

MCC and MNC	Examples: <code>mcc310</code> <code>mcc310-mnc004</code> <code>mcc208-mnc00</code> etc.	Screen aspect	<code>long</code> <code>notlong</code>
		Screen orientation	<code>port</code> <code>land</code>
Language and region	Examples: <code>en</code> <code>fr</code> <code>en-rUS</code> <code>fr-rFR</code> <code>fr-rCA</code> etc.	Dock mode	<code>car</code> <code>desk</code>
		Night mode	<code>night</code> <code>notnight</code>
Screen size	<code>small</code> <code>normal</code> <code>large</code> <code>xlarge</code>		

Recursos

Recursos alternativos. Calificadores.

Screen pixel density (dpi)	<code>ldpi</code> <code>mdpi</code> <code>hdpi</code> <code>xhdpi</code> <code>nodpi</code>	Primary text input method	<code>nokeys</code> <code>qwerty</code> <code>12key</code>
		Navigation key availability	<code>navexposed</code> <code>navhidden</code>
		Primary non-touch navigation method	<code>nonav</code> <code>dpad</code> <code>trackball</code> <code>wheel</code>
Touchscreen type	<code>notouch</code> <code>stylus</code> <code>finger</code>		
Keyboard availability	<code>keysexposed</code> <code>keyshidden</code> <code>keyssoft</code>	Platform Version (API Level)	Examples: <code>v3</code> <code>v4</code> <code>v7</code> etc.

Recursos

Recursos alternativos. Calificadores.

- Normas.
 - Puedes especificar distintos tipos de calificadores separados por un guión
 - Debes respetar el orden de los calificadores
 - No pueden anidarse
 - Son insensibles a las mayúsculas
 - Solo puede usarse un calificador de cada tipo

www.centic.es

Recursos

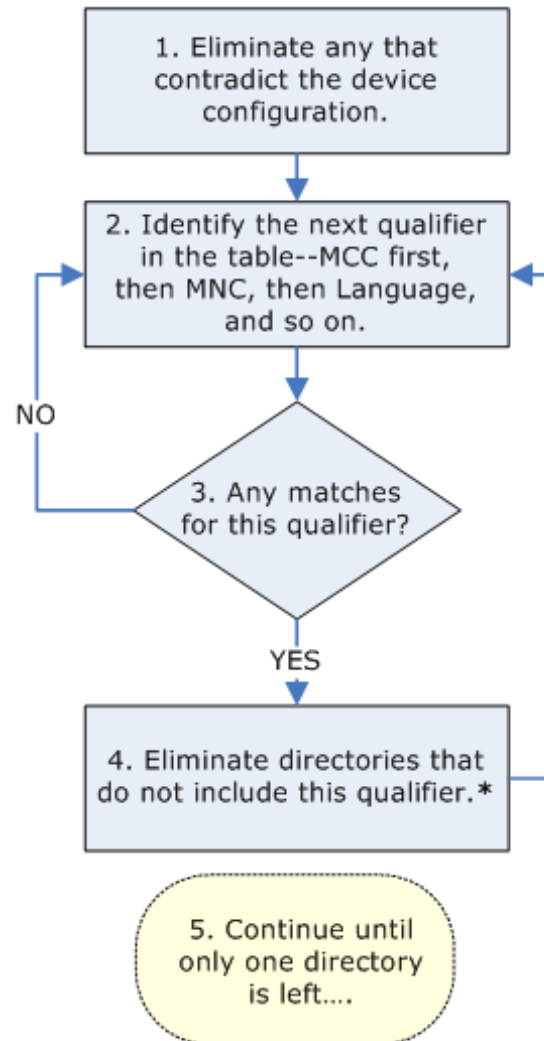
Compatibilidad.

- Usar siempre los directorios «por defecto»
- Usar values/string para el idioma por defecto
- Usar layout para las interfaces por defecto
- Ajustar siempre la propiedad minSdkVersion
- Adaptar las imágenes a las distintas densidades de pantalla
- La versión 1.5 no soporta:
 - Density
ldpi, mdpi, hdpi, nodps
 - Screen Size
small, normal, large
 - Screen aspect
long, notlong



Recursos

Selección de recursos.



* If the qualifier is the screen density, Android selects a "best" match and the process is done.

Recursos

Acceder a los recursos.

- R.java, Autogenerado
- Recursos
 - R.tipo_recurso.id_recurso
 - Android:name=id_recurso
 - XML
@tipo_recurso/id_recurso
 - Código
R.id.id_recurso
- Context.getResources()

www.centic.es

Recursos

Acceder a los recursos.

- Ejemplo desde Código

```
// Load a background for the current screen from a drawable resource
getWindow().setBackgroundDrawableResource(R.drawable.my_background_image) ;

// Set the Activity title by getting a string from the Resources object, because
// this method requires a CharSequence rather than a resource ID
getWindow().setTitle(getResources().getText(R.string.main_title));

// Load a custom layout for the current screen
setContentView(R.layout.main_screen);

// Set a slide in animation by getting an Animation from the Resources object
mFlipper.setInAnimation(AnimationUtils.loadAnimation(this,
    R.anim.hyperspace_in));

// Set the text on a TextView object using a resource ID
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello_message);
```

Recursos

Acceder a los recursos.

- Ejemplo desde XML

```
<?xml version="1.0" encoding="utf-8"?>
<EditText xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textColor="@color/opaque_red"
    android:text="@string/hello" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="opaque_red">#f00</color>
    <string name="hello">Hello!</string>
</resources>
```


Recursos

Acceder a los recursos.

- Recursos del sistema

```
<?xml version="1.0" encoding="utf-8"?>
<EditText xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textColor="@android:color/secondary_text_dark"
    android:text="@string/hello" />
```

```
setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, myarray));
```


Recursos

Tipos de recursos.

- Animation : res/anim/filename.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@[package:]anim/interpolator_resource"
    android:shareInterpolator=["true" | "false"] >
    <alpha
        android:fromAlpha="float"
        android:toAlpha="float" />
    <scale
        android:fromXScale="float"
        android:toXScale="float"
        android:fromYScale="float"
        android:toYScale="float"
        android:pivotX="float"
        android:pivotY="float" />
    <translate
        android:fromXDelta="float"
        android:toXDelta="float"
        android:fromYDelta="float"
        android:toYDelta="float" />
    <rotate
        android:fromDegrees="float"
        android:toDegrees="float"
        android:pivotX="float"
        android:pivotY="float" />
    <set>
        ...
    </set>
</set>
```



Recursos

Tipos de recursos.

- String: res/values/string.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello!</string>
</resources>
```

Recursos

Tipos de recursos.

- Layout: res/layout/string.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ViewGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+[package:]id/resource_name"
    android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
    android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
    [ViewGroup-specific attributes] >
    <View
        android:id="@+[package:]id/resource_name"
        android:layout_height=["dimension" | "fill_parent" | "wrap_content"]
        android:layout_width=["dimension" | "fill_parent" | "wrap_content"]
        [View-specific attributes] >
        <requestFocus/>
    </View>
    <ViewGroup >
        <View />
    </ViewGroup>
    <include layout="@layout/layout_resource"/>
</ViewGroup>
```

Recursos

Tipos de recursos.

- Selector: res/color/filename.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="true"
        android:color="#ffff0000"/> <!-- pressed -->
  <item android:state_focused="true"
        android:color="#ff0000ff"/> <!-- focused -->
  <item android:color="#ff000000"/> <!-- default -->
</selector>
```

Ejercicios

1. Crear una Actividad que muestre un mensaje en el LOG con todos sus posibles estados
2. Crear una jerarquía de Actividades compuesta de 5 actividades
 1. Lanzadas por contexto
 2. Lanzadas por su acción
 3. Cambiar el orden de las actividades
3. Capturar los eventos de giro de pantalla y cambiar el color de fondo según la orientación de la pantalla
4. Capturar el contenido de un campo de texto en una actividad
5. Crear una Actividad que se muestre en 3 idiomas
6. Crear una actividad que muestre un mensaje con el tipo de resolución de la pantalla y un color distinto en función de dicha resolución

