

Apuntes

[Descargar estos apuntes](#)

Tema 9. Menús Android

Índice


1. [Introducción](#)
2. [Definición de un menú como recurso XML](#)
3. [Overflow menu](#)

Introducción

Los menús forman parte, de forma usual, de la interfaz de una aplicación Android, permitiendo añadir funcionalidad sin ocupar mucho espacio en la pantalla.

[Información menús Material Design](#).

Hay cuatro tipos de menú en Android:

- **Overflow Menu**, menú principal que puede ser activado cuando se pulsa sobre el teléfono sobre el botón correspondiente (sea físico o software, dependiendo de su antigüedad) o sobre el icono  de la `AppBar` o `ToolBar`.
- **Contextual Menu**, son menús que aportan acciones extra para un determinado elemento de la vista. Habitualmente se activan con una pulsación larga sobre el mismo. Podemos encontrar a su vez dos tipos de menús contextuales: **Floating Context Menu** (abre un menú contextual que se superpone a la actividad) y **Contextual Action Mode** (abre una barra de acciones donde se agrupan las acciones a realizar sobre los elementos seleccionados).
- **Popup Menu**, es un menú emergente similar al **Overflow Menu**, pero que se encuentra vinculado a un elemento de la vista como una forma de ampliar las acciones que se pueden ejecutar.
- **Navigation Drawer**, o menú lateral deslizante, aparece dentro de las guías de diseño como un componente específico [Información adicional Material Design](#). Suele encontrarse en la pantalla principal de la app y contar con un botón para desplegarlo aunque también puede ser abierto deslizando el contenido de la pantalla desde el extremo izquierdo.

Definición de un menú como recurso XML

La alternativa más sencilla a la hora de definir un menú es dentro de un archivo XML. Podríamos definirlos también programáticamente, pero es bastante más complicado. Los archivos asociados a menús deben guardarse en la carpeta `res/menu` de nuestro proyecto. [Más información](#)

La estructura genérica de un xml asociado a un menú sería algo así:

```

<?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3   <item android:id="@+id/option_1"
        android:title="@string/option_1" />
        <item android:id="@+id/option_2"
        android:title="@string/option_2" />
        ...
8   <group android:id="@+id/group1_1">
        <item android:id="@+id/group_op1"
        android:title="@string/group_op1" />
        <item android:id="@+id/group_op2"
        android:title="@string/group_op2" />
    </group>
</menu>

```

✎ Aclaraciones definición de un menú Android:

- Línea 2, el elemento **<menu>** es el elemento raíz del documento.
- Línea 3, define un elemento del menú (una opción). Puede contener un elemento **<menu>** para la definición de un submenú. Los atributos habituales son: **android:id**, **android:icon** y **android:title**. Con **android:showAsAction** se indica donde aparecerá esta opción de menú:

Valor	Descripción
ifRoom	Solo coloca este elemento en la barra de la app si hay espacio. Si no hay lugar para todos los elementos marcados como "ifRoom", se muestran como acciones los elementos que tengan los valores orderInCategory más bajos; los restantes aparecerán en el menú ampliado.
withText	Incluye también el texto del título (definido por android:title) con el elemento de acción. Puedes incluir este valor junto con uno de los otros marcadores separándolos con un canal .
never	Nunca coloques este elemento en la barra de la app. En su lugar, enumera el elemento en el menú ampliado de la barra de la app.
always	Siempre coloca este elemento en la barra de la app. Evita usar esta opción a menos que sea esencial que el elemento siempre aparezca en la barra de acción. Configurar varios elementos para que siempre aparezcan como elementos de acción puede hacer que se superpongan con otra IU en la barra de la app.
collapseActionView	Es posible contraer la vista de acción asociada a este elemento de acción (declarada por android:actionLayout o android:actionViewClass). Se introdujo esta opción en la API nivel 14.

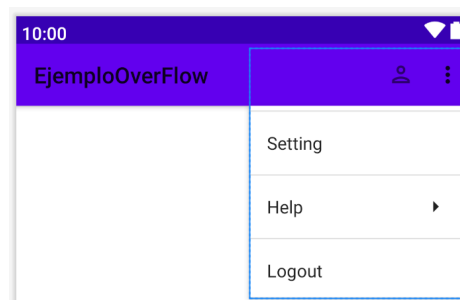
- Línea 8, la etiqueta **<group>** permite agrupar elementos **<item>** para poder actuar sobre ellos (mostrar, ocultar con **android:visible** o bien activar o desactivar con **android:enabled**) de forma conjunta. El atributo **android:checkableBehavior** permite que los elementos de este grupo se puedan activar con un botón de selección, sus valores posibles son: **single** (solo un elemento del grupo se puede seleccionar) **all** (todos se pueden seleccionar) y **none** (ninguno es seleccionable).

Overflow menu

La ubicación en la pantalla donde aparecen los elementos del menú de opciones depende de la versión para la que desarrollaste la aplicación: para versión **Android 2.x (nivel API 10) o versiones anteriores** el menú aparecerá en la parte superior de la pantalla cuando el usuario pulse el botón **Menú**; para **Android 3.0 (nivel API 11) y versiones posteriores** el **Overflow Menu** aparece en la barra de la App.



Vamos a desarrollar un **Overflow menu** con las siguientes opciones:



La primera opción **Edit** aparece con un icono y debe aparecer fuera del **Overflow menu**. Las dos siguientes opciones queremos que estén agrupadas, siendo la opción **Help** un submenú con dos opciones más: **App** y **Android**. Finalmente una opción **Logout**.

Veamos la definición de este menú con ubicación **res/menu/menu_overflow.xml**:


```


<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item android:id="@+id/opEdit"
        android:title="@string/opEdit"
        android:icon="@drawable/outline_person_24"
7      app:showAsAction="ifRoom|withText"/>
8  <group android:id="@+id/group_Config">
    <item android:id="@+id/opSetting"
          android:title="@string/opSetting" />
    <item android:id="@+id/opHelp"
          android:title="@string/opHelp">
13      <menu>
          <item android:id="@+id/opHelpApp"
                android:title="@string/opHelpApp" />
          <item android:id="@+id/opHelpAndroid"
                android:title="@string/opHelpAndroid"/>
        </menu>
      </item>
    </group>
    <item android:id="@+id/opLogout"
          android:title="@string/opLogout" />
  </menu>

```

Aclaraciones:

- **Línea 7**, con `android:showAsAction=ifRoom|withText` hacemos que este elemento del menú aparezca siempre fuera del **Overflow menu** y que si hay espacio suficiente se muestre el texto y el icono.
- **Línea 8**, definimos un grupo con dos `item`.
- **Línea 13**, definimos dentro de la opción **Help** un submenú con dos opciones.

 Comentar que previamente se ha creado el layout para este fragmento y se corresponde con `fragment_uno.xml`.

 **Crea una App fragmentV1, para probar el ejemplo anterior, el fragment1 será estático mientras que el fragment2 será añadido de forma dinámica desde el código**

[ViewModel](#).



Para ello vamos a volver a implementar el último ejemplo.