

MENUS



Programación multimedia
y dispositivos móviles
2º DAM

ÍNDICE DE CONTENIDOS

1. Introducción
2. Definir un menú
3. Menú de opciones
4. Menú contextual

1. INTRODUCCIÓN

Los menús son un componente común de la interfaz de usuario en muchos tipos de aplicaciones.

Para proporcionar una experiencia de usuario conocida y uniforme, Android proporciona una API específica para la creación de menús

Esta API nos permite crear, principalmente, tres tipos de menú:

- Menú de opciones
- Menú contextual
- Menú emergente (Popup)

<https://developer.android.com/guide/topics/ui/menus.html?hl=es-419>

2. DEFINIR UN MENÚ

Para todos los tipos de menús, Android proporciona un formato XML estándar para definir los elementos del menú:

- Crear un recurso nuevo en la carpeta menú.
 - Este recurso xml, contendrá los ítems que se tienen que mostrar.
- Una vez definido el menú, se cargará en la actividad o fragmento que corresponda.

2. DEFINIR UN MENÚ

Para definir un menú contamos con los siguientes elementos:

- **<menu>** → Define un Menu, que es un contenedor para elementos del menú. Un elemento **<menu>** debe ser el nodo raíz del archivo y puede tener uno o más elementos **<item>** y **<group>**
- **<item>** → Crea un MenuItem, que representa un único elemento en un menú. Este elemento puede contener un elemento **<menu>** anidado para crear un submenú.
- **<group>** → Contenedor opcional e invisible para elementos **<item>**. Te permite categorizar los elementos del menú para que compartan propiedades, como el estado de una actividad o visibilidad.

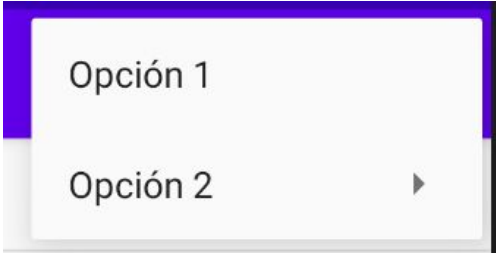
2. DEFINIR UN MENÚ

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:id="@+id/ajustes" android:title="Ajustes"></item>  
    <item android:id="@+id/acerca" android:title="Acerca de"></item>  
</menu>
```

2. DEFINIR UN MENÚ

Podemos crear submenús, indicando una nueva etiqueta `<menu>` dentro de una etiqueta `<item>`.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/opcion1" android:title="Opción 1"></item>
  <item android:id="@+id/opcion2" android:title="Opción 2">
    <menu>
      <item android:id="@+id/subOpcion1" android:title="SubOpción 1"></item>
      <item android:id="@+id/subOpcion2" android:title="SubOpción 2"></item>
    </menu>
  </item>
</menu>
```



Opción 1

Opción 2



Opción 2

SubOpción 1

SubOpción 2

3. MENÚ DE OPCIONES

- El menú de opciones es la colección principal de elementos de menú para una actividad.
- Se utilizan para colocar las acciones que tienen un impacto global en la app, como “Buscar”, “Redactar correo electrónico” y “Ajustes”.

3. MENÚ DE OPCIONES

Para agregar un menú de opciones a nuestra Activity debemos usar la función `onCreateOptionsMenu()` de la siguiente forma:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_opciones, menu);
    return true;
}
```



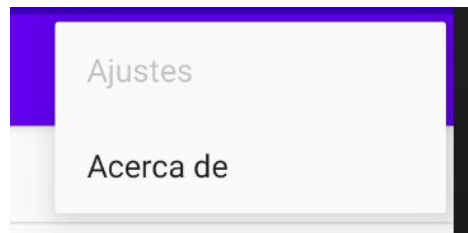
Ajustes

Acerca de

3. MENÚ DE OPCIONES

- Del mismo modo que sucede con los layouts convencionales, es posible modificar los elementos del menú de forma programática, es decir, en tiempo de ejecución.
- En el siguiente ejemplo podemos ver cómo deshabilitamos menuitem Ajustes de forma que se hace imposible hacer clic en él.

```
@Override  
public boolean onPrepareOptionsMenu(Menu menu) {  
    menu.findItem(R.id.ajustes).setEnabled(false);  
    return true;  
}
```



3. MENÚ DE OPCIONES

Evento:

- Cuando el usuario selecciona un elemento del menú de opciones el sistema llama al método `onOptionsItemSelected()` de la actividad.

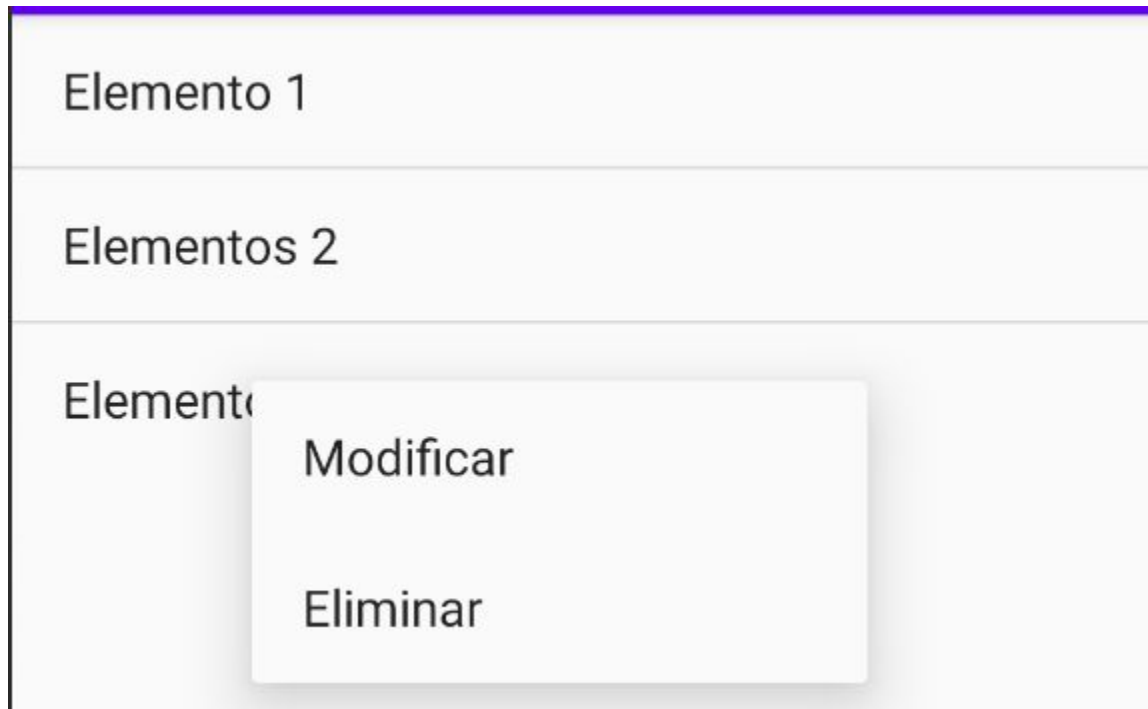
```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case R.id.ajustes:
            Toast.makeText(context: this, text: "Ajustes", Toast.LENGTH_SHORT).show();
            break;
        case R.id.acerca:
            Toast.makeText(context: this, text: "Acerca de", Toast.LENGTH_SHORT).show();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

4. MENÚ CONTEXTUAL

Un menú contextual es un menú flotante que aparece cuando el usuario hace un clic largo en un elemento. Proporciona acciones que afectan el contenido seleccionado o el marco contextual.

Puedes proporcionar un menú contextual para cualquier vista, aunque estos se usan con mayor frecuencia para elementos en **ListView**, **GridView** u otras colecciones de vistas en las cuales el usuario puede realizar acciones directamente en cada elemento.

4. MENÚ CONTEXTUAL



4. MENÚ CONTEXTUAL

El primer paso para crear un menú contextual es, como ya vimos para el menú de opciones, definir un XML con las opciones que queremos mostrar.

- Se define del mismo modo y, en la vista previa, se mostrará como si de un menú de opciones se tratase, no debemos preocuparnos por ello.

El siguiente paso será registrar nuestro elemento (ListView, GridView, etc) para que pueda utilizar el menú contextual mediante el método `registerForContextMenu(Vista)`.

```
registerForContextMenu(lista);
```

4. MENÚ CONTEXTUAL

Para agregar un menú contextual a nuestra Activity debemos usar la función `onCreateContextMenu()` de la siguiente forma:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_contextual, menu);
}
```

4. MENÚ CONTEXTUAL

Para manejar la pulsación sobre cualquiera de las opciones del menú debemos crear el método `onContextItemSelected(...)`

En él, mediante un `switch`, vamos a establecer las acciones a realizar para cada uno de los ítems del menú a través del `id` que les dimos en el XML

```
@Override
public boolean onContextItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case R.id.modificar:
            break;
        case R.id.eliminar:
            break;
    }
    return super.onContextItemSelected(item);
}
```


4. MENÚ CONTEXTUAL

Podemos acceder a datos sobre el objeto que hemos pulsado a través del AdapterView del objeto item

```
@Override
public boolean onContextItemSelected(@NonNull MenuItem item) {
    AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.modificar:
            Toast.makeText(context: this, text: "Modificar " + info, Toast.LENGTH_SHORT).show();
            break;
        case R.id.eliminar:
            Toast.makeText(context: this, text: "Eliminar " + info, Toast.LENGTH_SHORT).show();
            break;
    }
    return super.onContextItemSelected(item);
}
```

5. DUDAS Y PREGUNTAS

