¿CÓMO CREAR UNA TOOLBAR?

En los siguientes pasos se describe la manera de configurar una `Toolbar` como la barra de app de tu actividad:

1. Agrega la biblioteca de compatibilidad v7 appcompat a tu proyecto, como se describe en Configuración de la biblioteca de compatibilidad.
2. Asegúrate de que la actividad extienda `AppCompatActivity`:

```
public class MyActivity extends AppCompatActivity {
  // ...
}
```

Nota: Realiza este cambio para todas las actividades de tu app que usen una **`Toolbar`** como barra de app.

3. En el manifiesto de la app, configura el elemento `<application>` para usar uno de los temas `NoActionBar` de appcompat. El uso de uno de estos temas evita que en la app se use la clase `ActionBar` nativa para proporcionar la barra de app. Por ejemplo:

```
<application
    android:theme="@style/Theme.AppCompat.Light.NoActionBar"
    />
```

4. Agrega una `Toolbar` al diseño de la actividad. Por ejemplo, el siguiente código de diseño agrega una `Toolbar` y hace que esta flote por encima de la actividad:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <android.support.v7.widget.Toolbar
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/appbar"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:minHeight="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" >

    </android.support.v7.widget.Toolbar>

    <!-- Resto de la interfaz de usuario -->

</LinearLayout>
```

En la especificación de material design se recomienda que las barras de app tengan una elevación de 4 dp.

Ubica la barra de herramientas en la parte superior del diseño de la actividad, ya que la usarás como una barra de app.

5. En el método `onCreate()` de la actividad, llama al método `setSupportActionBar()` de la actividad y pasa la barra de herramientas de la actividad. Este método establece la barra de herramientas como la barra de app de la actividad. Por ejemplo:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my);
    Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);
    setSupportActionBar(myToolbar);
    }
```

Ahora tu app tiene una barra de acciones básica

Usar los métodos de utilidad de la barra de app

Una vez que configures la barra de herramientas como una barra de app de la actividad, podrás acceder a los diferentes métodos de utilidad que proporciona la clase `ActionBar` de la biblioteca de compatibilidad v7 appcompat. Este enfoque te permite realizar varias acciones útiles, como ocultar y mostrar la barra de app.

Para usar los métodos de utilidad de `ActionBar`, llama al método `getSupportActionBar()` de la actividad. Este método muestra una referencia a un objeto appcompat `ActionBar`. Una vez que tengas esa referencia, podrás llamar a cualquiera de los métodos de `ActionBar` para ajustar la barra de app. Por ejemplo, para ocultar la barra de app llama a `ActionBar.hide()`.

PROBLEMAS DE CREAR ASÍ LA TOOLBAR

En cada Activity que la querías poner, tenéis que repetir todo el proceso anterior: el xml, …

SOLUCIÓN a los problemas de arriba

Para ello, declaramos primero el toolbar en un layout XML independiente, por ejemplo en un fichero llamado */res/layout/toolbar.xml*:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:minHeight="?attr/actionBarSize"
    android:background="?attr/colorPrimary"
    android:elevation="4dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light" >

</android.support.v7.widget.Toolbar>
```

Y posteriormente incluir este fragmento en el layout de nuestras actividades haciendo referencia a él mediante `include`:

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <include android:id="@+id/appbar"
        layout="@layout/toolbar" />

    <!-- Resto de la interfaz de usuario -->

</LinearLayout>
```

Como podéis ver, en este caso definimos la propiedad `android:id` del control en en el `include`, y no en el layout independiente del toolbar.

CÓMO PONER UN MENÚ EN LA TOOLBAR Y OPCIONES con ICONOS

Se hace como cualquier menú normal, creando el menú en xml, y luego inflándolo en onCreateOptionsMenu() y programando las acciones de cada ítem del menú. Ver la guía de crear menús.