

PRACTICA 21 CREAR UN MENU DESDE CERO.

Crear una aplicación con un menú para entender cómo interactúa cada uno de los elementos de la aplicación.

Cree una aplicación con la plantilla Empty y cambie el Layout a LinearLayout.

Una vez creada, acceda al gradle module y agregue lo siguiente, sustituyendo las XX por la versión soportada:

```
compile 'com.android.support:design:XX.+'
```

Como se indica en la figura:

```
27 compile 'com.android.support:appcompat-v7:26.+'
28 compile 'com.android.support.constraint:constraint-layout:1.0.2'
29 testCompile 'junit:junit:4.12'
30 compile 'com.android.support:design:26.+'
```

Debe de tener cuidado que el número de versión coincida, en este caso la línea 27 indica la versión, finalmente sincronice la aplicación.

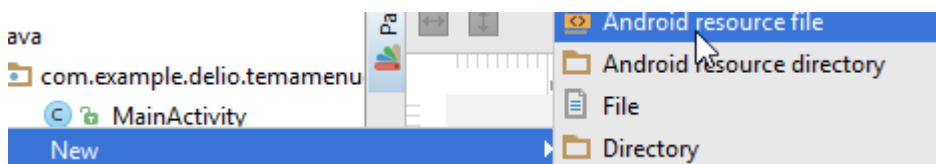
Ahora iremos al archivo style.xml y cambiaremos el tema de:

```
parent="Theme.AppCompat.Light.DarkActionBar"
```

por

```
parent="Theme.AppCompat.Light.NoActionBar"
```

Bien ahora crearemos un archivo xml que contendrá el menú, para esto debemos de ir a la carpeta res y dar botón dercho y generar un nuevo archivo de tipo Android resource



Y desplegara la siguiente venta y cambiamos a menú el tipo de recurso.

New Resource File

File name: mymenu

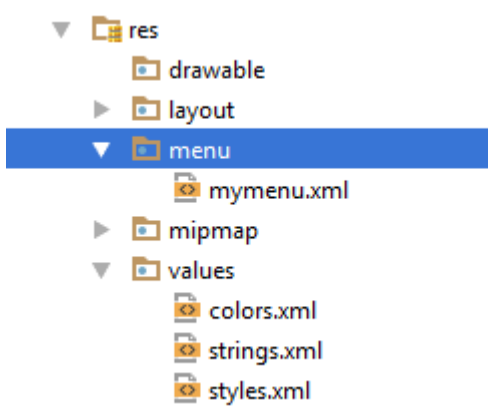
Resource type: Menu

Root element: menu

Source set: main

Directory name: menu

De tal forma que se generara una carpeta tipo menú como se indica:



Ahora procedemos a editarlo y agregar su contenido del menú, inicialmente esta como se indica:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

Agregamos el siguiente código:

```
<item
    android:id="@+id/action_setting"
    android:title="@string/setting"
    app:showAsAction="always"
/>
<item
    android:id="@+id/action_about"
    android:title="@string/about"
    app:showAsAction="ifRoom"
/>
```

Quedando asi:

```
<?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/action_setting"
        android:title="@string/setting"
        app:showAsAction="always"
    />
```

```

        <item
            android:id="@+id/action_about"
            android:title="@string/about"
            app:showAsAction="ifRoom"
        />

    </menu>

```

Ahora vamos a agregar un Toolbar a nuestra actividad principal para que ahí se genere nuestro menú.

Modificaremos el código del Toolbar para que queden bien las letras, ya que por default son negras, modificamos el código de:

```

<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"
/>

```

Al siguiente:

```

<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
/>

```

Ahora implementemos la lógica para desplegar nuestro menú:

Definimos nuestro Toolbar :

```
private Toolbar toolbar;
```

Y lo enlazamos al objeto visual:

```
toolbar = (Toolbar) findViewById(R.id.toolbar);
```

```
setSupportActionBar(toolbar);
```

Una vez hecho esto, debemos de crear un evento que genere el menú, por tanto generamos este evento:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    return super.onCreateOptionsMenu(menu);
}
```

A este evento debemos de modificarlo:

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

Hasta este punto, ya hemos creado nuestro menú, usted puede ejecutar la apps y vera los resultados en su emulador o dispositivo móvil.

Ahora vamos a darle funcionalidad a los elementos del menú mediante el evento:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    return super.onOptionsItemSelected(item);
}
```

A este código lo modificaremos y quedara:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if(item.getItemId()==R.id.action_setting){
        Toast.makeText(getApplicationContext(),
            "Setting is Selected",
            Toast.LENGTH_LONG).show();
    }
    if(item.getItemId()== R.id.action_about){
        Toast.makeText(getApplicationContext(),
            "About is Selected",
            Toast.LENGTH_LONG).show();
    }
    return super.onOptionsItemSelected(item);
}
```

Ya con esto tenemos nuestra aplicación funcional, solo hay que adaptar los eventos a nuestras necesidades.