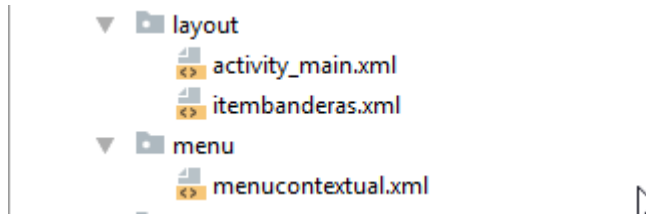


Clase 25 MENU CONTEXTUALES.

Vamos a crear un menú contextual para la aplicación de banderas que se vio en la ultima sesión, agregaremos lo necesario para que podamos hacer uso de los menus contextuales al extender un click o toque sobre la pantalla. Para esto vamos a modificar el código de nuestra aplicación:

Agregaremos un menú el cual será nuestro menú contextual.



y se nombrará como menucontextual.xml y tendrá lo siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/action_territorio"
        android:title="@string/action_terrotorio"
    />
    <item
        android:id="@+id/action_idioma"
        android:title="@string/actionond_idioma"
    />
    <item
        android:id="@+id/action_poblacion"
        android:title="@string/action_poblacion"
    />
</menu>
```

Como podemos apreciar, hace referencia a cadenas de texto que están dentro de string.xml

```
<string name="action_poblacion">Poblacion</string>
<string name="action_terrotorio">Territorio</string>
<string name="actionond_idioma">Idioma</string>
```

Con esto hemos definido nuestro menú.

MODIFICACION DE LA ACTIVIDAD PRINCIPAL.

Vamos a sobre escribir dos métodos, el primero:

```
@Override
public void onCreateContextMenu
    (ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)
{
    super.onCreateContextMenu(menu, v, menuInfo);

    int id = v.getId();
    MenuInflater inflater = getMenuInflater();

    if (id==R.id.lstBanderas){
        inflater.inflate(R.menu.menucontextual, menu);
    }
}
```

Este método creará nuestro menú (debemos de identificar el objeto sobre el cual estamos haciendo un click largo para crear su menú relacionado, en este caso como solo es uno, no hay problema, pero de igual forma se verifica que sea el que se desea programar.

Como es un ListView el que estamos agregando un menú contextual, debemos de saber a que elemento hemos afectado, por tanto, sobre escribimos el segundo método:

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info =
    (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    int posicion;
    String mensaje;
    String pais = banderas.get(info.position).getTxtPais().toString();
    switch (item.getItemId()){
        case R.id.action_idioma:{
            posicion = info.position;
            String [] idiomas =
            getResources().getStringArray(R.array.idiomas);
            mensaje = " En " + pais + " se habla " +
            item.getTitle().toString() + " " + idiomas[posicion];
            break;
        }

        case R.id.action_poblacion:{
            posicion = info.position;
            String [] poblacion =
            getResources().getStringArray(R.array.personas);
            mensaje = " En " + pais + " tiene una " +
            item.getTitle().toString() + " de " + poblacion[posicion];
        }
    }
}
```

Y al igual, debemos de detectar que selecciona el usuario para tomar acciones, en este caso solo es desplegar una notificación en función de los datos que están almacenados en diferentes arreglos de textos:

```
<string-array name="territorios">
  <item>3,450,000 km</item>
  <item>5,400,000 km</item>
  <item>7,120,000 km</item>
  <item>2,300,100 km</item>
  <item>50,300 km</item>
  <item>100,120 km</item>
  <item>850,000 km</item>
  <item>1,100,000 km</item>
  <item>60,600 km</item>
  <item>6,200,600 km</item>
  <item>23,500 km</item>
</string-array>
```

[illegible]

Para Poblacion:

```
<string-array name="personas">
    <item>2,300,100 personas</item>
    <item>50,300 personas</item>
    <item>100,120 personas</item>
    <item>850,000 personas</item>
    <item>1,100,000 personas</item>
    <item>60,600 personas</item>
    <item>3,450,000 personas</item>
    <item>5,400,000 personas</item>
    <item>7,120,000 personas</item>
    <item>100,000 personas</item>
    <item>34,600 personas</item>
</string-array>
```

Note que el nombre del país lo obtenemos del ArrayList<Banderas>.

Con esto complementamos el uso de los adaptadores y holder con un menú contextual.