Desarrollo para dispositivos móviles con Android: menús.

Contenido

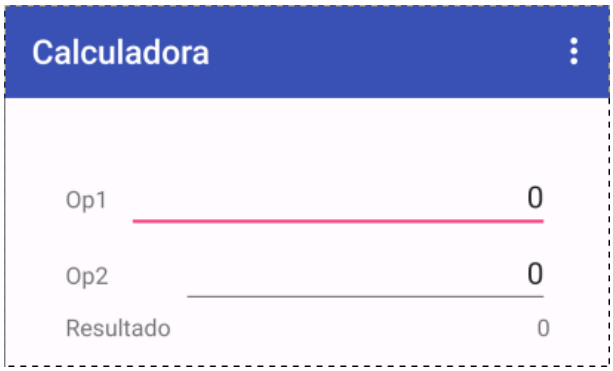
Desarrollo para dispositivos móviles con Android: menús....1
1Introducción....2
2Calculadora....2
3Menú principal de opciones....3
4Menús contextuales....6
5Referencias....8

1 Introducción

En todas las aplicaciones Android suelen aparecer dos tipos de menús: el menú de opciones de la principal de la actividad, y los menús contextuales. Mientras el menú de actividad es accesible desde la barra de título de la misma, en el borde superior de la pantalla, los segundos se activan siempre mediante una pulsación larga de algún control gráfico.

2 Calculadora

Durante este documento se tratará el ejemplo de una calculadora formada por dos entradas de texto y un visor de texto. El objetivo es hacer un cálculo con los números mostrados en ambas entradas, y mostrar el resultado en el visor de texto. Como ejercicio, no hay botones ni diálogos, solo menú. La aplicación se ve apoyada por una lógica de negocio muy simple, la clase **Calculadora**.




```
public class Calculadora {
    public enum OperadoBinario { Add, Sub, Mul, Div };
    public enum OperadorUnario { Neg, Sqr, Sqrt };

    public Calculadora(String op1, String op2)
    {
        this( Double.parseDouble( op1 ), Double.parseDouble( op2 ) );
    }

    public Calculadora(double op1, double op2) {
        this.op1 = op1;
        this.op2 = op2;
    }

    public double calcula(OperadorUnario opr)
    {
        double toret = this.getOp1();

        switch (opr) {
            case Neg:
                toret *= -1;
                break;
            case Sqr:
                toret *= toret;
                break;
            case Sqrt:
                toret = Math.sqrt( toret );
                break;
        }

        return toret;
    }
    public double calcula(OperadoBinario opr) {
        double toret = 0;

        switch (opr) {
            case Add:
                toret = this.getOp1() + this.getOp2();
                break;

            case Sub:
                toret = this.getOp1() - this.getOp2();
                break;
```

```
            case Mul:
                toret = this.getOp1() * this.getOp2();
                break;
            case Div:
                toret = this.getOp1() / this.getOp2();
                break;
        }

        return toret;
    }

    public double getOp1() {
        return this.op1;
    }

    public double getOp2() {
        return this.op2;
    }

    private double op1;
    private double op2;
}
```

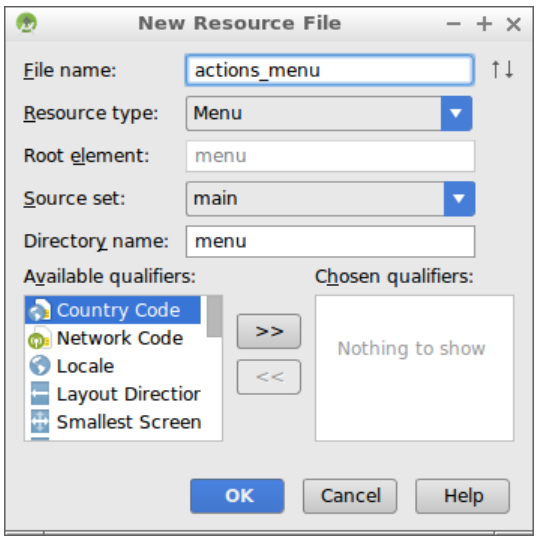
La interfaz Android de la aplicación debe soportar la funcionalidad dada por esta clase. Esto implica las operaciones de suma, resta, multiplicación y división para los dos operandos, así como negación, raiz cuadrada y elevación al cuadrado para un solo operando.

La forma de soportar esta funcionalidad será la siguiente: el usuario rellena los campos de texto y después selecciona la operación deseada en el menú principal. En cuanto a los operadores unarios, se hace una pulsación larga en el **EditText** que tenga el valor a utilizar, y se escoge el operador unario del menú contextual.

3 Menú principal de opciones

Para crear un menú de opciones para la actividad, lo primero es crear la carpeta *menu/* bajo la carpeta *res/*. Una vez hecho esto, será necesario crear un archivo XML que represente las opciones del menú principal. Si este archivo se llama *main_menu.xml*, entonces su ruta completa será *res/menu/main_menu.xml*.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="@string/opSuma"            android:id="@+id/opSuma" />
    <item android:title="@string/opResta"           android:id="@+id/opResta" />
    <item android:title="@string/opMultiplica"      android:id="@+id/opMultiplica" />
    <item android:title="@string/opDivide"          android:id="@+id/opDivide" />
</menu>
```

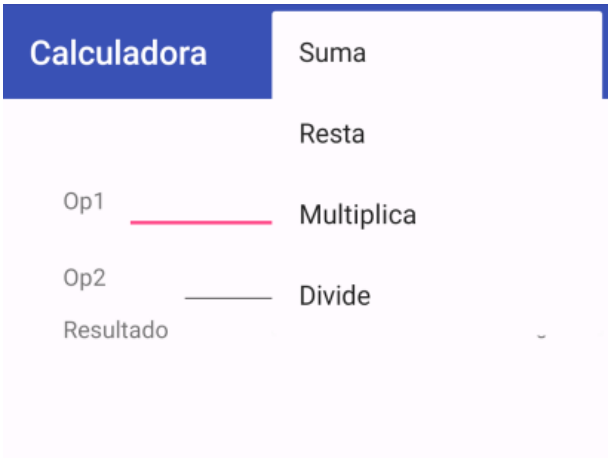


En realidad, el proceso de creación de este archivo es muy sencillo gracias a la asistencia de Android Studio. Basta seleccionar New >> Android Resource File, y, en la ventana de configuración, seleccionar “menu” en el desplegable, para que se cree automáticamente.

En este ejemplo se está empleando el archivo *res/string.xml* para no hacer este archivo de menú dependiente de un determinado idioma, por ejemplo. Se crea la opción con identificador *opSuma* con el texto *@string/opSuma*. Los identificadores de las cadenas se muestran a continuación.

```
<resources>
    <string name="app_name">Calculadora</string>
    <string name="lblOp1">Op1</string>
    <string name="lblOp2">Op2</string>
    <string name="lblResultado">Resultado</string>
    <string name="opSuma">Suma</string>
    <string name="opResta">Resta</string>
    <string name="opMultiplica">Multiplica</string>
    <string name="opDivide">Divide</string>
    <string name="opNegativo">Convierte a negativo</string>
    <string name="opCuadrado">Eleva al cuadrado</string>
    <string name="opRaiz">Raiz cuadrada</string>
</resources>
```

Una vez creado el menú principal, solo es necesario realizar dos acciones: registrarlo, y responder a las activaciones de sus elementos. Lo primero se consigue sobreescribiendo el método *onCreateOptionsMenu(Menu m)*, y lo segundo con *onOptionsItemSelected(MenuItem mi)*. Ambos métodos existen en la clase **Activity**,



Así, en primer lugar es necesario crear el menú cuando el usuario pulse la opción del menú en la barra de la actividad. Para ello, es necesario sobreescribir el método **Activity**.*onCreateOptionsMenu()*.

```
public class Main extends Activity {
            //…
            @Override
            public boolean onCreateOptionsMenu(Menu menu)
            {
                super.onCreateOptionsMenu( menu );

                this.getMenuInflater().inflate( R.menu.actions_menu, menu );
                return true;
            }
            //...
}
```

El método **MenuInflater**.*inflate()*, (del objeto **MenuInflater** obtenido mediante **Activity**.*getMenuInflater()*) devuelve un constructor de menús, que tomando el archivo XML (referenciado como *R.menu.actions_menu*) y un objeto **Menu**, introduce las opciones descritas en el primero, dentro del segundo.

Posteriormente, para saber cuál opción ha sido pulsada, será necesario sobreescribir **Activity**.*onOptionsItemSelected()*. La opción específica se descubre comparando el id de la opción pasada por parámetro con los identificadores definidos en el menú.

```
public class Main extends Activity {
    //…
    @Override
    public boolean onOptionsItemSelected(MenuItem menuItem)
    {
        boolean toret = false;

        switch( menuItem.getItemId() ) {
```

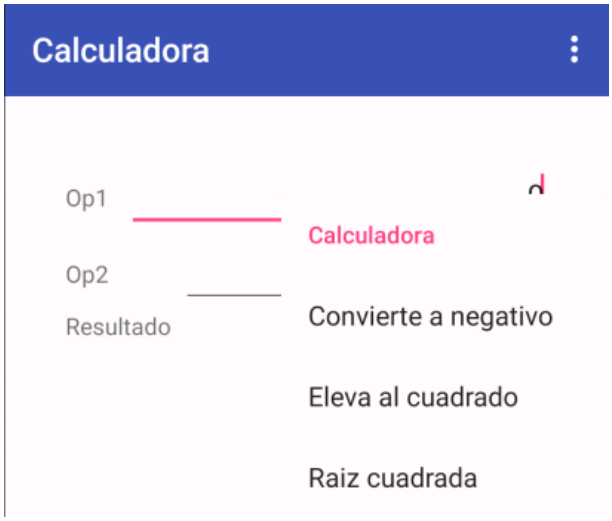
```
            case R.id.opSuma:
                this.operaBinario( Calculadora.OperadoBinario.Add );
                toret = true;
                break;
            case R.id.opResta:
                this.operaBinario( Calculadora.OperadoBinario.Sub );
                toret = true;
                break;
            case R.id.opMultiplica:
                this.operaBinario( Calculadora.OperadoBinario.Mul );
                toret = true;
                break;
            case R.id.opDivide:
                this.operaBinario( Calculadora.OperadoBinario.Div );
                toret = true;
                break;
        }

        return toret;
    }

    private void operaBinario(Calculadora.OperadoBinario op)
    {
        EditText edOp1 = (EditText) this.findViewById( R.id.edOp1 );
        EditText edOp2 = (EditText) this.findViewById( R.id.edOp2 );
        TextView edResultado = (TextView) this.findViewById( R.id.edResultado );

        try {
            Calculadora calc =
                      new Calculadora( edOp1.getText().toString(), edOp2.getText().toString() );
            edResultado.setText( Double.toString( calc.calcula( op ) ) );
        }
        catch(NumberFormatException | ArithmeticException exc)
        {
            edResultado.setText( "0" );
            Toast.makeText( this, "Arithmetic error", Toast.LENGTH_LONG ).show();
        }
     }
}
```

4 Menús contextuales



El usuario obtiene menús contextuales mediante una pulsación larga sobre un control determinado. Todos los menús contextuales de cualquier control dentro de una actividad, se tratan también desde la misma actividad, registrándolos (mediante **Activity**.*registerForContextMenu()*) para cada control que los soporte desde el método *onCreate()*. El resto del manejo es similar al de los menús principales: su creación y la pulsación de los items, se obtiene sobreescribiendo los métodos **Activity**.*onCreateContextMenu()* y **Activity**.*onContextItemSelected()*. En el caso del primero, dado que pueden existir varios menús contextuales, es necesario comprobar si se trata del control (**View**) en el que se puede crear el menú contextual.

El menú en concreto a emplear como contextual se crea como un archivo de recursos de Android conteniendo la descripción de un menú, al igual que como se explicaba más arriba para menús principales. Específicamnte, las opciones disponibles son las siguientes, correspondientes a los operadores unarios. Refiérase más arriba al archivo *string.xml*.

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

    <item android:title="@string/opNegativo"       android:id="@+id/context_op_negativo" />
    <item android:title="@string/opCuadrado"       android:id="@+id/context_op_cuadrado" />
    <item android:title="@string/opRaiz"           android:id="@+id/context_op_raiz" />
</menu>
```

En nuestro código, el primer paso es registrar el menú contextual en los dos **EditText**, dentro del método *onCreate()*. A continuación, sobrecargar el método *onCreateContextMenu()*.

```
public class Main extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_main );

        EditText edOp1 = (EditText) this.findViewById( R.id.edOp1 );
        EditText edOp2 = (EditText) this.findViewById( R.id.edOp2 );

        this.registerForContextMenu( edOp1 );
        this.registerForContextMenu( edOp2 );
    }

    public void onCreateContextMenu(ContextMenu contxt, View v, ContextMenu.ContextMenuInfo cmi)
    {
        if ( v.getId() == R.id.edOp1
          || v.getId() == R.id.edOp2 )
        {
            this.getMenuInflater().inflate( R.menu.context_menu, contxt );
            contextMenu.setHeaderTitle( R.string.app_name );
        }
    }
```

Finalmente, es necesario responder a las pusaciones de las opciones. Dado que el menú contextual puede ser lanzado estando seleccionado cualquiera de los dos **EditText**, será necesario deducir cuál obteniendo el comoponente que tiene el foco.

```
    @Override
    public boolean onContextItemSelected(MenuItem menuItem)
    {
        boolean toret = false;

        switch( menuItem.getItemId() ) {
            case R.id.context_op_cuadrado:
                this.operaUnario( Calculadora.OperadorUnario.Sqr );
                toret = true;
                break;
            case R.id.context_op_raiz:
                this.operaUnario( Calculadora.OperadorUnario.Sqrt );
                toret = true;
                break;
            case R.id.context_op_negativo:
                this.operaUnario( Calculadora.OperadorUnario.Neg );
                toret = true;
                break;
        }

        return toret;
}

private void operaUnario(Calculadora.OperadorUnario op)
    {
        // Buscar el elemento con el foco
        View componente = this.getCurrentFocus();

        if ( componente instanceof EditText ) {
            EditText edOp = (EditText) componente;

            try {
                Calculadora calc = new Calculadora( edOp.getText().toString(), "-1" );
                edOp.setText( Double.toString( calc.calcula( op ) ) );
            }
            catch(NumberFormatException | ArithmeticException exc)
            {
                edOp.setText( "0" );
                Toast.makeText( this, "Arithmetic error", Toast.LENGTH_LONG ).show();
            }

        }

        return;
    }
```

La clase es muy sencilla, y el ejemplo subyacente suficientemente simple como para entender por completo el mecanismo de menús de Android.

Un ejemplo un poco más complejo se produce cuando el menú contextual se aplica a un **ListView**. En ese caso, querremos saber cuál fue la opción el el foco cuando se invocó el menú contextual. Lo único que cambia es la forma de recuperar la posición del control, dentro de *onContextItemSelected()*. En el siguiente ejemplo, el **ListView** guarda información sobre series de televisión. El usuario puede incrementar el número de episodio ya visto, o incluso la temporada (aeason) entera.

```
@Override
public boolean onContextItemSelected(MenuItem item)
{
        switch( item.getItemId() ) {
                case R.id.contextMenuIncEpisode:
                        int pos = ( (AdapterView.AdapterContextMenuInfo)
                                                item.getMenuInfo() ).position;
                        this.listSeries.get(  pos ).incEpisode();
                        Toast.makeText( this, "Episode number incremented",
                                                Toast.LENGTH_SHORT ).show();
                        break;
                case R.id.contextMenuIncSeason:
                        int pos = ( (AdapterView.AdapterContextMenuInfo)
                                                item.getMenuInfo() ).position;
                        this.listSeries.get( pos ).incSeason();
                        Toast.makeText( this, "Season number incremented",
                                                Toast.LENGTH_SHORT ).show();
                        break;
        }
}
```

5 Referencias

- Documentación y recursos de Android para desarrolladores (accedido en sept. 2016) http://developer.android.com/
- Android: menús https://developer.android.com/reference/android/view/Menu.html
- Tutoriales: menus https://developer.android.com/guide/topics/ui/menus.html
- Aplicación de ejemplo seguida a lo largo de este tema: https://github.com/Baltasarq/TestMenus/
- Ejemplo de App Android con menús: https://github.com/Baltasarq/DroidSeries/