

RESUMEN PMDM

Índice

Tipos de Layouts:.....	2
OnClickListener:.....	3
RadioGroup:.....	4
Cambiar el diseño cuando tiene un width diferente:.....	5
Cambio de activity:.....	6
Dar permisos en el manifest:.....	9
Poner alarma en un Intent:.....	9
Controlar version SDK:.....	10
Intent Filter:.....	11
Para hacer que los Ids no sean estáticos:.....	12
Pedir permisos al SO:.....	13
Prohibir la rotación de la pantalla:.....	14
Guardar datos al rotar la pantalla:.....	14
Recuperar datos (Forma 1):.....	14
Recuperar datos (Forma 2):.....	14
Para cambiar el icono:.....	15
Unidades:.....	15

Tipos de Layouts:

1. FrameLayout:

- Coloca todo en la esquina superior izquierda

2. GridLayout:

- Se indican las filas y columnas con **rowCount** y **columnCount**
- La posición se indica con **layout_row** y **layout_column**
- Para que ocupe varias posiciones **rowSpan** y **columnSpan**.

3. LinearLayout:

- Se indica la orientación de los elementos con **orientation**.

4. RelativeLayout:

- Se coloca todo con respecto a otros elementos
- **Propiedades al posicionar una vista con respecto a otra:**
 - **android:layout_above**: sitúa la vista encima del id especificado.
 - **android:layout_below**: sitúa la vista debajo del id especificado.
 - **android:layout_toLeftOf**: sitúa la vista a la izquierda.
 - **android:layout_toRightOf**: sitúa la vista a la derecha.
 - **android:layout_alignLeft**: alinea el borde izqdo. con el del id especificado.
 - **android:layout_alignRight**: alinea el borde dcho. con el del id especificado.
 - **android:layout_alignTop**: alinea el borde superior con el del id especificado
 - **android:layout_alignBottom**: alinea el borde inferior con el del id especificado.
- **Propiedades al posicionar una vista respecto a su contenedor:**
 - **android:layout_alignParentLeft**: si es true, alinea con el borde izqdo. del padre.
 - **android:layout_alignParentRight**: si es true, alinea con el borde dcho. del padre.
 - **android:layout_alignParentTop**: si es true, alinea con el borde sup. del padre.
 - **android:layout_alignParentBottom**: si es true, alinea con el borde inf. del padre.
 - **android:layout_centerHorizontal**: si es true, centra en horiz. respecto al padre.
 - **android:layout_centerVertical**: si es true, centra en vertical respecto al padre.
 - **android:layout_centerInParent**: si es true, centra en ambos sentidos.

5. TableLayout:

- Se indican las filas mediante **TableRow**
- Tendrá tantas columnas como elementos tenga la fila más grande

OnClickListener:

```
//Método 1: Crear clase auxiliar
private class Auxiliar implements View.OnClickListener{

    @Override
    public void onClick(View view) {
        tvMensaje.setText("Has pulsado Si");
    }
}
```

```
//Método 2: Crear una clase anónima (más recomendable que el método 1)
private View.OnClickListener escuchadorNo=new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvMensaje.setText("Has pulsado No");
    }
};
```

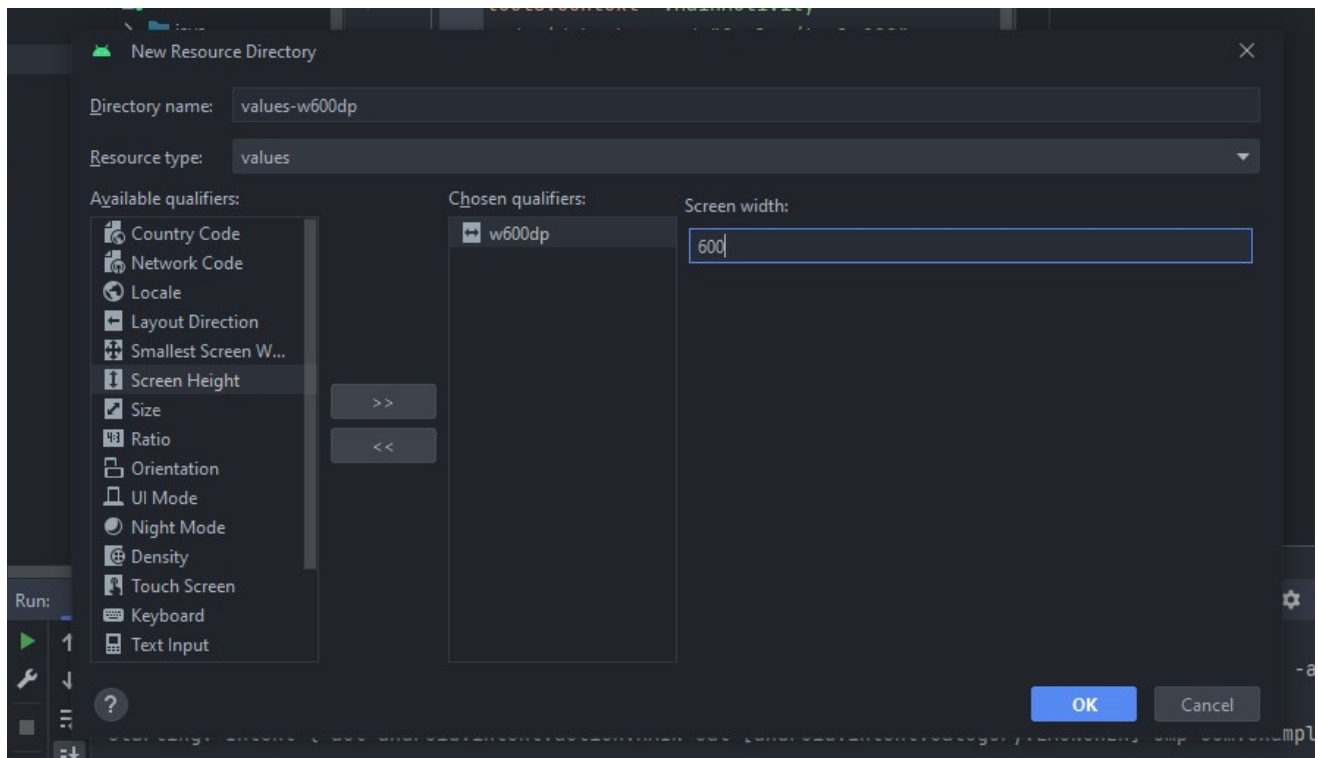
```
//Método 3: Crear el escuchador y asignarlo en un sólo paso
btnAVeces.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        tvMensaje.setText("Has pulsado A Veces");
    }
});
```

```
//Método 4: Implementar la interfaz al MainActivity directamente
@Override
public void onClick(View v) {
    tvMensaje.setText("Has pulsado No sé");
}
//Asignar el escuchador mediante this (MÉTODO 4)
btnNoSe.setOnClickListener(this);
```

RadioGroup:

```
@Override
public void onClick(View view) {
    if (etValor.getText().toString().equals("")){
        Toast.makeText(this, "Debes teclear un valor", Toast.LENGTH_SHORT).show();
    } else {
        try {
            if (rbtAptas.isChecked()) {
                valor = Float.parseFloat(etValor.getText().toString());
                resultado = (float) (valor / 166.38);
                txtResultado.setText(etValor.getText().toString() + " pesetas son " + resultado + " euros.");
            } else {
                valor = Float.parseFloat(etValor.getText().toString());
                resultado = (float) (valor * 166.38);
                txtResultado.setText(etValor.getText().toString() + " euros son " + resultado + " pesetas.");
            }
        }
        catch (NumberFormatException e){
            Toast.makeText(this, "El valor introducido debe ser numérico", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Cambiar el diseño cuando tiene un width diferente:



Cambio de activity:

```
public void onClickToActivity2(View view) {

    if (view.getId() == R.id.btnSaludar){
        try{
            nombreText = nombre.getText().toString();
            String edad = comprobarEdad();
            String genero = generoText;
            texto = "Hola, " + genero + " " + nombreText + "\n" + edad;
            Intent intent = new Intent(this, Saludando.class);
            intent.putExtra("mensaje", texto);
            startActivityForResult(intent, LLAMADA_DESPEDIDA);
        } catch (Exception ex){
            Toast.makeText(this, "Hay campos sin rellenar", Toast.LENGTH_SHORT).show();
        }

    }

}
```

Enviar resultado de una activity a otra que está escuchando:

```
btnFin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (v.getId() == R.id.btnFin){

            Intent intent = new Intent();
            intent.putExtra("mensaje", despedida);
            setResult(RESULT_OK, intent);
            Log.i("ciclo", "Ejecutando listener() on Activity 5");
            finish();
        }
    }
});
```

Escuchar resultado de una activity:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    //Vemos quien nos contesta
    if (requestCode == LLAMADA_DESPEDIDA){
        //testeamos el código del resultado
        if (resultCode == RESULT_OK){
            //operaciones si la actividad llamada finaliza según lo previsto
            Toast.makeText(this, "Todo ok", Toast.LENGTH_SHORT).show();
            tvDespedida.setText(data.getStringExtra("mensaje"));
        } else {
            //operaciones si la actividad llamada no hace lo previsto
            Toast.makeText(this, "Algo falló", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Distintas formas de cambiar de activities:

```
public void onClickCambioActivity(View view) {
    Intent intent = null;
    if (view.getId()==R.id.btnLlamadaAct2){
        intent = new Intent(this, Activity2.class);
        startActivity(intent);

    } else if (view.getId()==R.id.btnLlamadaAct3) {
        intent = new Intent(this, Activity3.class);
        dato = "El Activity 1 envía este mensaje a la activity3";
        intent.putExtra("mensaje", dato);
        startActivity(intent);

    } else if (view.getId()==R.id.btnLlamadaAct4) {
        intent = new Intent(this, Activity4.class);
        dato = "El Activity 1 envía este mensaje a la activity4 a través de un bundle";

        Bundle bundle = new Bundle();
        bundle.putString("mensaje", dato);

        intent.putExtras(bundle);
        startActivity(intent);
    } else if (view.getId() == R.id.btnLlamadaEsperaRespuesta){
        intent = new Intent(this, Activity5.class);

        //Llamada esperando respuesta
        startActivityForResult(intent, LLAMADA_ACTIVIDAD5);
    } else if (view.getId() == R.id.btnLlamadaOtraApp) {
        intent = new Intent();
        intent.setClassName("com.example.ex3_cuentaclick","com.example.ex3_cuentaclick.MainActivity");

        try {
            startActivity(intent);
        } catch (ActivityNotFoundException a){
            Toast.makeText(this, "Ninguna actividad puede realizar esta acción", Toast.LENGTH_SHORT).show();
        }
    }

    //Opción con Package Manager

    /* PackageManager pm = getPackageManager();
    List actividadesPosibles = pm.queryIntentActivities(intent, PackageManager.MATCH_DEFAULT_ONLY);
    if (actividadesPosibles.size()>0){
        startActivity(intent);
    } else {
        Toast.makeText(this, "Error", Toast.LENGTH_SHORT).show();
    }
    */
    } else if (view.getId()==R.id.btnLlamadaSystemApp){
        try {
            intent = new Intent();
            intent.setClassName("com.android.calculator2","com.android.calculator2.Calculator");
            startActivity(intent);
        } catch (ActivityNotFoundException act){
            Toast.makeText(this, "No se encontró la calculadora", Toast.LENGTH_SHORT).show();
        }
    } else if (view.getId()==R.id.btnLlamadaSystemAppSettings){
        try {
            intent = new Intent();
            intent.setClassName("com.android.settings","com.android.settings.Settings");
        }
    }
}
```

```
        startActivity(intent);
    } catch (ActivityNotFoundException act){
        Toast.makeText(this, "No se encontraron las Settings", Toast.LENGTH_SHORT).show();
    }
}
```


Dar permisos en el manifest:

```
<uses-permission android:name="com.android.alarm.permission.SET_ALARM"/>
```

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-feature  
    android:name="android.hardware.telephony"  
    android:required="false" />  
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

Poner alarma en un Intent:

```
public void onClickBtn(View view) {  
  
    String mensaje = "Esto es el aviso actual de mi alarma";  
  
    int hora = tpHora.getHour();  
    int minutos = tpHora.getMinute();  
  
    Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM);  
    intent.putExtra(AlarmClock.EXTRA_MESSAGE, mensaje);  
    intent.putExtra(AlarmClock.EXTRA_HOUR, hora);  
    intent.putExtra(AlarmClock.EXTRA_MINUTES, minutos);  
  
    if (intent.resolveActivity(getPackageManager()) != null){  
  
        startActivity(intent);  
  
    } else {  
  
        Toast.makeText(this, "Error al realizar la acción", Toast.LENGTH_SHORT).show();  
  
    }  
}
```

Controlar version SDK:

```
if (android.os.Build.VERSION.SDK_INT >= 23) {  
  
    hora = tpHora.getHour();  
    minutos = tpHora.getMinute();  
  
} else {  
  
    hora = tpHora.getCurrentHour();  
    minutos = tpHora.getCurrentMinute();  
  
}
```

Intent Filter:

```
<activity
    android:name=".ImplicitActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data android:scheme="http"/>
        <data android:scheme="https"/>
    </intent-filter>
</activity>
```

Intent Explícito e Intent Implícito:

```
public void onClickbtn(View view) {

    String url = etUrl.getText().toString();

    if (url.isEmpty()){
        Toast.makeText(this, "El campo URL no debe de estar vacío", Toast.LENGTH_SHORT).show();
    } else {
        Intent intent = null;

        if (view.getId() == R.id.btnWeb){

            intent = new Intent(this, Navegador.class);
            intent.putExtra("enlace", url);
            startActivity(intent);

        } else if (view.getId() == R.id.btnWebImpl) {

            intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
            startActivity(intent);

        }
    }
}
```

Para hacer que los Ids no sean estáticos:

```
android.nonFinalResIds = false
```

Añadimos esta línea en gradle.properties

Pedir permisos al SO:

```
if (ActivityCompat.checkSelfPermission(this, Manifest.permission.CALL_PHONE) ==  
PackageManager.PERMISSION_GRANTED){  
  
    //realizar la llamada  
    intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:(+34)685749875"));  
    startActivity(intent);  
  
} else { //solicitamos al SO la gestión del permiso  
  
    ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CALL_PHONE},  
LLAMADA_TELEFONO2);  
  
}
```

Método para recibir la respuesta a los permisos del usuario:

```
//Método en donde recibimos la respuesta que ha dado el usuario  
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[]  
grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
    if ((requestCode == LLAMADA_TELEFONO)){  
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED){  
            //acciones correspondientes al permiso concedido  
            Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:(+34)685749875"));  
            startActivity(intent);  
            Toast.makeText(this, "El usuario ha concedido los permisos", Toast.LENGTH_SHORT).show();  
        } else {  
            //acciones correspondientes al permiso NO concedido  
            Toast.makeText(this, "El usuario ha denegado los permisos", Toast.LENGTH_SHORT).show();  
        }  
    }  
}
```

Prohibir la rotación de la pantalla:

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

Guardar datos al rotar la pantalla:

```
@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt("contador", num);
    Log.i("ciclo", "Ejecutando onSaveInstanceState()");
}
```

Recuperar datos (Forma 1):

```
@Override
protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    num = savedInstanceState.getInt("contador");
    Log.i("ciclo", "Ejecutando onRestoreInstanceState()");

    if (num == 1) {
        status.setText("Has pulsado " + num + " vez.");
    } else {
        status.setText("Has pulsado " + num + " veces.");
    }
}
```

Recuperar datos (Forma 2):

Dentro del onCreate()

```
if (savedInstanceState != null) {
    num = savedInstanceState.getInt("contador");

    Log.i("ciclo", "contador = " + num);

    if (num == 1) {
        status.setText("Has pulsado " + num + " vez.");
    } else {
        status.setText("Has pulsado " + num + " veces.");
    }
}
```

RECORDAR DAR VALORES A LOS GETTERS

Para cambiar el icono:

FORMA 1:

1. Vamos a drawable en res y pegamos la imagen en formato .png
2. Cambiamos la ruta en el manifest a la de nuestra imagen:

```
android:icon="@mipmap/ic_launcher"
```

```
android:icon="@drawable/icono"
```

3. Cambiamos tambien la ruta del round:

```
android:roundIcon="@mipmap/ic_launcher_round"
```

```
android:icon="@drawable/icono"
```

FORMA 2:

1. En res click derecho -> new -> image asset
2. En asset type marcamos clip art/image y elegimos icono en clip art/path

Unidades:

- dp: tamaños de dimensiones
- sp: tamaños de fuente
- dpi: densidad