

# PDM Avanzado Datos Persistentes Preferencias

---

## Sumario

---

### Introdución

#### Preferencias

- Acceso ás preferencias
- Gardando novos datos
- Recuperando datos

#### Caso Práctico 1

- Creamos a Activity Principal
- Creamos a Activity Secundaria

#### Pantallas de Preferencias

- Introdución
- Valores asociados ás pantallas de preferencias
- Definindo as pantallas de preferencias
  - Caso Práctico 2
    - Datos da pantalla de preferencia
    - Definición da pantalla de preferencias
    - Carga da pantalla de preferencias
- Definindo categorías dentro das Pantallas de Preferencias
- Creando novas pantallas de opcións
- Caso Práctico 3
  - Arquivo de valores
  - Pantalla de Preferencias
  - Activity Principal
  - Creamos a Activity Secundaria
- Controlando cando cambia algunha preferencia
- Chamando a Activity principal dende a pantalla de preferencias

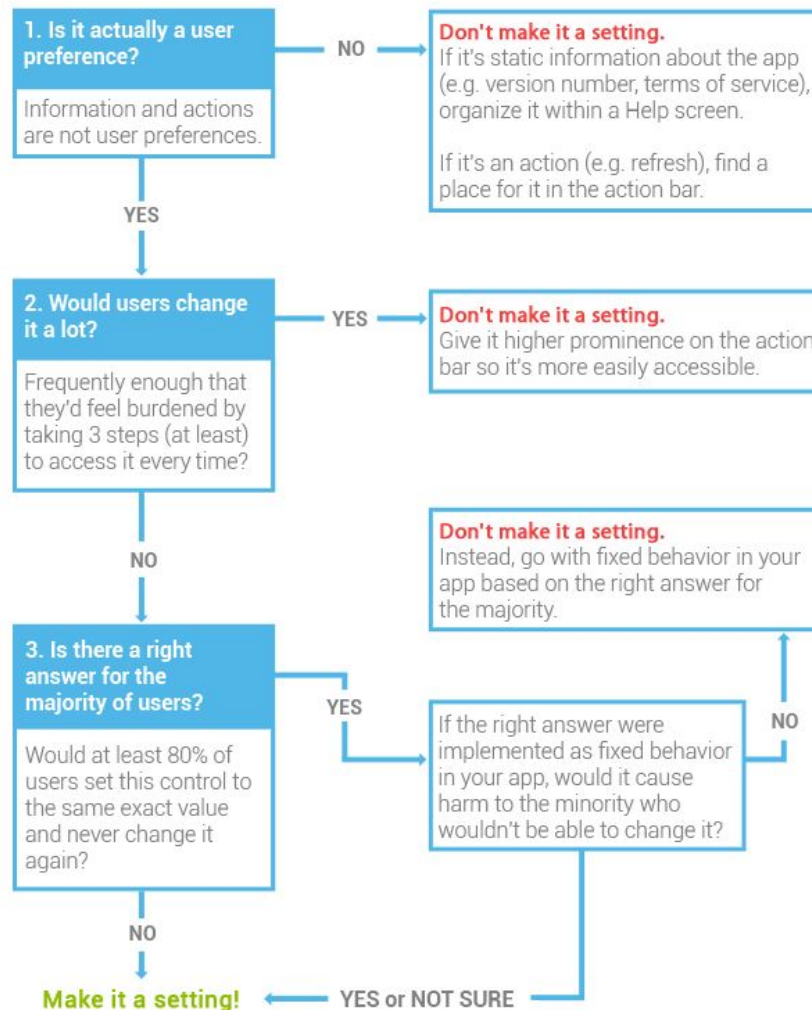
## Introdución

---

As preferencias permiten gardar tipos de datos simples (cadeas, enteiros,...) en forma de pares clave-valor, exemplo "idade-43"

Estes vanse gardar nun obxecto pertencente á clase SharedPreferences (<http://developer.android.com/reference/android/content/SharedPreferences.html>).

Para saber se un axuste é necesario gardalo coma unha preferencia, podemos empregar un pequeno Test feito por Google:



## Preferencias

### Acceso ás preferencias

Para obter un obxecto SharedPreferences temos dúas opcións:

- **getPreferences()**: usado cando só necesitamos un único arquivo de preferencias e polo tanto non necesitamos indicar o nome. Neste caso créase un arquivo de preferencias por Activity e non pode ser accedido dende activities diferentes.
- **getSharedPreferences()**: usado cando temos varios arquivos de preferencias. Necesitamos indicar o nome do arquivo.

Un exemplo:

```
SharedPreferences sharedPref = getPreferences(MODE_PRIVATE);
```

```
SharedPreferences sharedP = getSharedPreferences("datos", MODE_PRIVATE);
```

Como vemos, nas dúas formas temos que indicarlle o 'modo' en como se van utilizar as preferencias:

- **MODE\_PRIVATE**: modo predeterminado que indica que o arquivo só o pode usar a **aplicación** que o creou.
- **MODE\_WORLD\_READABLE**: arquivo lido por todas as aplicacións.
- **MODE\_WORLD\_WRITEABLE**: arquivo actualizable por todas as aplicacións.
- **MODE\_MULTI\_PROCESS**: múltiples procesos poden cambiar os datos do arquivo de preferencias compartidas. Non se sole utilizar.

**Nota:** Máis información en <http://developer.android.com/reference/android/content/Context.html>

Qué diferenza hai entre getPreferences() e getSharedPreferences() ?

- Con **getPreferences()** abrimos un arquivo de preferencias propio de cada activity. É dicir, só terá acceso a activity que o creou.
- Con **getSharedPreferences()** abrimos un arquivo de preferencias indicando un nome. Calquera activity que abra dito arquivo (indicando o nome) vai poder acceder aos datos gardados.

Polo tanto:

- Utilizaremos **getPreferences()** cando queiramos gardar datos locais a unha activity.
- Utilizaremos **getSharedPreferences()** cando queiramos compartir datos entre diferentes activities.

O lugar físico onde se garda o/os arquivos de preferencia atópase en: **/data/data/nome\_paquete/Shared\_prefs/**. Pódese comprobar utilizando o Device File Explorer ([http://wiki.cifprodolfoucha.es/index.php?title=Dispositivos\\_Virtuais\\_de\\_Android:\\_AVD#SDCard\\_-\\_Device\\_File\\_Explorer](http://wiki.cifprodolfoucha.es/index.php?title=Dispositivos_Virtuais_de_Android:_AVD#SDCard_-_Device_File_Explorer)) có emulador.

## Gardando novos datos

Para poder modificar / crear valores nun SharedPreferences necesitamos acceder ao seu **Editor** da forma:

- Obtemos o SharedPreferences (cunha das formas anteriores).

```
SharedPreferences sharedpref = getPreferences(MODE_PRIVATE);
```

- Accedemos ao seu editor:

```
SharedPreferences.Editor editor = sharedpref.edit();
```

- Engadimos ou modificamos as preferencias que necesitamos.

Para isto necesitamos chamar ó método putXXXXX dependendo do tipo de dato.  
Por exemplo:

```
editor.putFloat("NOME_PARAMETRO", VALOR_A_MODIFICAR);
```

- E MOI IMPORTANTE, chamamos ao **commit()** para que garde as modificacións....

```
editor.commit();
```

## Recuperando datos

- Obtemos o SharedPreferences (cunha das formas anteriores).

```
SharedPreferences sharedpref = getPreferences(MODE_PRIVATE);
```

- Chamamos o seu método getXXXXX có tipo de datos correspondente...

```
Float tamaño = sharedpref.getFloat("NOME_PARAMETRO", VALOR_POR_DEFECTO);
```

O **valor por defecto** será o que gardemos na variable tamaño en caso de que non exista ese parámetro no arquivo de preferencias compartidas.

## Caso Práctico 1

O obxectivo desta práctica é ver como gardar datos de forma global e local utilizando SharedPreferences.

A práctica consistirá en dúas Activities.

- A primeira activity (Activity principal) terá este aspecto:



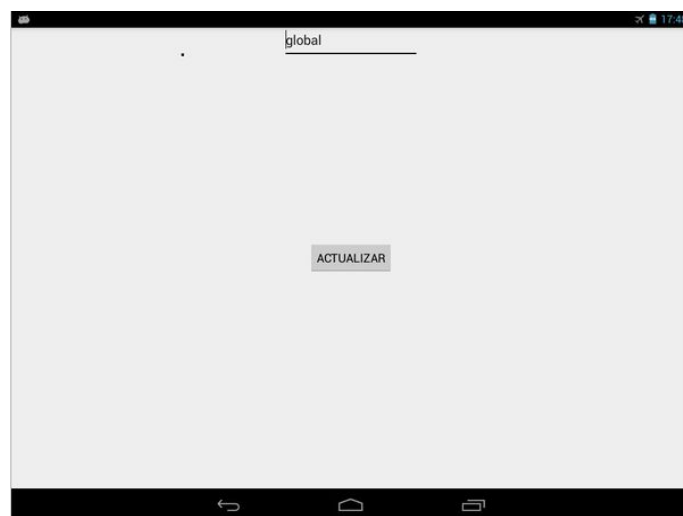
Esta activity está composta por dúas caixas de texto editables.

- A primeira ten un dato que se vai gardar localmente utilizando `getPreferences()`.
- A segunda ten un dato que se vai gardar globalmente utilizando `getSharedPreferences()`.

A función dos botóns será:

- Botón 'Gardar Datos': Garda os datos das dúas caixas de texto.
- Botón 'Cargar Datos': Carga os datos dende os `SharedPreferences`.
- Botón 'Lanzar Activity': Lanza unha segunda activity na que se vai amosar o dato da caixa gardado no `getSharedPreferences()`.

A segunda activity terá este aspecto:



Esta activity está composta por unha caixa de texto editable e un botón.

- Caixa de texto: Carga o dato gardado no `getSharedPreferences()`.
- Botón 'Actualizar': Garda o dato da caixa de texto e pecha a activity.

## Creemos a Activity Principal

- Partimos que xa temos creado o proxecto inicial como xa indicamos anteriormente ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proyecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proyecto_base)).

Se non o temos creado antes, crearemos un novo paquete de nome: **Persistencia** como un subpaquete do teu paquete principal.

Se non o temos creado antes, crearemos un novo paquete de nome: **Preferencias** como un subpaquete do paquete anterior.

- Dentro do paquete **Preferencias** crear unha nova 'Empty Activity' de nome: **UD05\_01\_A1\_DatosPersistentes\_Preferencias** de tipo Launcher.

Modifica o arquivo **AndroidManifest.xml** e engade unha label á activity como xa vimos na creación do proxecto base ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

### Código do layout xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".Persistencia.Preferencias.UD05_01_A1_DatosPersistentes_Preferencias"
8     android:id="@+id/relativeLayout">
9
10    <EditText
11        android:id="@+id/UD6_01_A1_etDatoGardarLocal"
12        android:layout_width="0dp"
13        android:layout_height="wrap_content"
14        android:hint="Dato a gardar localmente"
15        android:inputType="text"
16        android:textSize="15sp"
17        app:layout_constraintLeft_toLeftOf="parent"
18        app:layout_constraintRight_toRightOf="parent"
19        app:layout_constraintTop_toTopOf="parent">
20
21        <requestFocus />
22    </EditText>
23
24    <Button
25        android:id="@+id/UD6_01_A1_btnGardarDatos"
26        android:layout_width="200dp"
27        android:layout_height="wrap_content"
28        android:layout_marginBottom="8dp"
29        android:layout_marginEnd="8dp"
30        android:layout_marginStart="8dp"
31        android:text="GARDAR DATOS"
32        app:layout_constraintBottom_toTopOf="@+id/UD6_01_A1_btnCargarDatos"
33        app:layout_constraintEnd_toEndOf="parent"
34        app:layout_constraintStart_toStartOf="parent" />
35
36    <Button
37        android:id="@+id/UD6_01_A1_btnCargarDatos"
38        android:layout_width="200dp"
39        android:layout_height="wrap_content"
40        android:layout_marginBottom="8dp"
41        android:layout_marginEnd="8dp"
42        android:layout_marginStart="8dp"
43        android:text="CARGAR DATOS"
44        app:layout_constraintBottom_toTopOf="@+id/UD6_01_A1_btnLanzarActivitySegunda"
45        app:layout_constraintEnd_toEndOf="parent"
46        app:layout_constraintStart_toStartOf="parent" />
47
48    <Button
49        android:id="@+id/UD6_01_A1_btnLanzarActivitySegunda"
50        android:layout_width="200dp"
51        android:layout_height="wrap_content"
52        android:layout_marginBottom="8dp"
53        android:layout_marginEnd="8dp"
54        android:layout_marginStart="8dp"
55        android:text="LANZAR ACTIVITY"
56        app:layout_constraintBottom_toBottomOf="parent"
57        app:layout_constraintEnd_toEndOf="parent"
58        app:layout_constraintStart_toStartOf="parent" />
59
60    <EditText
61        android:id="@+id/UD6_01_A1_etDatoGardarGlobal"
62        android:layout_width="0dp"
63        android:layout_height="wrap_content"
64        android:layout_marginEnd="8dp"
65        android:layout_marginStart="8dp"
66        android:layout_marginTop="8dp"
67        android:ems="10"
68        android:hint="Dato a gardar globalmente"
69        android:textSize="15sp"
70        app:layout_constraintEnd_toEndOf="parent"
71        app:layout_constraintStart_toStartOf="parent"
72        app:layout_constraintTop_toBottomOf="@+id/UD6_01_A1_etDatoGardarLocal" />
73
74 </android.support.constraint.ConstraintLayout>

```

## Código da clase UD05\_01\_A1\_DatosPersistentes\_Preferencias

**Obxectivo:** Gardar datos de acceso local e global utilizando SharedPreferences.

**Nota:** O código vos dará un erro xa que fai referencia a unha segunda activity que aínda non está creada. O faremos a continuación.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Persistencia.Preferencias;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.EditText;
10 import android.widget.Toast;
11 import es.cursoandroid.cifprodolfoucha.aprendiendo.R;
12
13 public class UD05_01_A1_DatosPersistentes_Preferencias extends AppCompatActivity{
14     public static final String NOME_ARQUIVO_PREFERENCIAS="DATOS_GARDAR";
15     public static final String NOME_PARAM_DATO="DATO";
16
17     private SharedPreferences datosPrivados;
18     private SharedPreferences datosGlobales;
19
20     private void xestionarEventos(){
21
22         Button btnCargarDatos = (Button)findViewById(R.id.UD6_01_A1_btnCargarDatos);
23         btnCargarDatos.setOnClickListener(new View.OnClickListener() {
24
25             @Override
26             public void onClick(View arg0) {
27                 // TODO Auto-generated method stub
28
29                 EditText textoLocal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarLocal);
30                 textoLocal.setText(datosPrivados.getString(NOME_PARAM_DATO, "Non hai valor gardado localmente"));
31
32                 EditText textoGlobal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarGlobal);
33                 textoGlobal.setText(datosGlobales.getString(NOME_PARAM_DATO, "Non hai valor gardado globalmente"));
34
35             }
36         });
37
38         Button btnGardar = (Button)findViewById(R.id.UD6_01_A1_btnGardarDatos);
39         btnGardar.setOnClickListener(new View.OnClickListener() {
40
41             @Override
42             public void onClick(View v) {
43                 // TODO Auto-generated method stub
44                 EditText textoLocal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarLocal);
45
46                 SharedPreferences.Editor editor = datosPrivados.edit();
47                 editor.putString(NOME_PARAM_DATO, textoLocal.getText().toString());
48                 editor.commit();
49                 textoLocal.setText("");
50
51                 EditText textoGlobal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarGlobal);
52
53                 SharedPreferences.Editor editorGlobal = datosGlobales.edit();
54                 editorGlobal.putString(NOME_PARAM_DATO, textoGlobal.getText().toString());
55                 editorGlobal.commit();
56                 textoGlobal.setText("");
57
58                 Toast.makeText(getApplicationContext(), "DATOS GARDADOS", Toast.LENGTH_SHORT).show();
59             }
60         });
61
62         Button btnLanzarActivity = (Button)findViewById(R.id.UD6_01_A1_btnLanzarActivitySegunda);
63         btnLanzarActivity.setOnClickListener(new View.OnClickListener() {
64
65             @Override
66             public void onClick(View arg0) {
67                 // TODO Auto-generated method stub
68
69                 Intent intent = new Intent(getApplicationContext(),UD6_01_A2_DatosPersistentesPreferencias.class);
70                 startActivity(intent);
71             }
72         });
73
74     }
75
76 }
77
78
79
80
81
82

```

```

83  @Override
84  protected void onResume(){
85      super.onResume();
86
87      EditText textoLocal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarLocal);
88      textoLocal.setText(datosPrivados.getString(NOME_PARAM_DATO, "Non hai valor gardado localmente"));
89
90      EditText textoGlobal = (EditText)findViewById(R.id.UD6_01_A1_etDatoGardarGlobal);
91      textoGlobal.setText(datosGlobales.getString(NOME_PARAM_DATO, "Non hai valor gardado globalmente"));
92
93  }
94
95  @Override
96  protected void onCreate(Bundle savedInstanceState) {
97      super.onCreate(savedInstanceState);
98      setContentView(R.layout.activity_UD05_01_a1_datos_persistentes_preferencias);
99
100     datosGlobales = getSharedPreferences(NOME_ARQUIVO_PREFERENCIAS, MODE_PRIVATE);
101     datosPrivados = getPreferences(MODE_PRIVATE);
102
103     xestionarEventos();
104
105  }
106 }

```

- Liña 14: Definimos o nome do arquivo que vai gardar as preferencias.
- Liña 15: Definimos o nome do parámetro onde se vai gardar o dato global.
- Liña 17: Definimos o SharedPreferences por defecto (local a activity).
- Liña 18: Definimos o SharedPreferences que utilizará un nome de arquivo.
- Liñas 23-36: Xestionamos o evento Click sobre o botón Cargar Datos.
  - Liñas 30 e 33: Recollemos o dato gardado no SharedPreferences. *OLLO! ao valor por defecto!*
- Liñas 40-63: Xestionamos o evento Click sobre o botón Gardar Datos.
  - Liñas 47-49: Obtemos o Editor do SharedPreferences local e cambiamos o dato polo contido da caixa de texto.
  - Liñas 55-57: Obtemos o Editor do SharedPreferences global e cambiamos o dato polo contido da caixa de texto.
- Liñas 83-93: Cargamos os datos que se atopen nos SharedPreferences. Ao facelo no **onResume** isto leva consigo a que cando volvamos da Activity Secundaria cargue as modificacións feitas. **OLLO!** tamén vai pasar o mesmo cando iniciemos a aplicación, por primeira vez ou calquera vez.

## Creamos a Activity Secundaria

- Nome da activity: **UD6\_01\_A2\_DatosPersistentesPreferencias**.

**NON É DE TIPO LAUNCHER** xa que vai ser chamada por outra Activity.

### Código do layout do xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".Persistencia.Preferencias.UD6_01_A2_DatosPersistentesPreferencias">
7
8      <EditText
9          android:id="@+id/UD05_01_A2_etDato"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:layout_alignParentTop="true"
13         android:layout_centerHorizontal="true"
14         android:hint="Texto das preferencias"
15         android:inputType="text"
16         android:textSize="20sp"
17     >
18
19     <requestFocus />
20 </EditText>
21
22 <Button
23     android:id="@+id/UD05_01_A2_btnActualizar"
24     android:layout_width="wrap_content"
25     android:layout_height="wrap_content"
26     android:layout_centerHorizontal="true"

```

```

27         android:layout_centerVertical="true"
28         android:layout_marginTop="42dp"
29         android:text="ACTUALIZAR" />
30
31 </RelativeLayout>

```

### Código da clase UD6\_01\_A2\_DatosPersistentesPreferencias

**Obxectivo:** Amosa o dato gardado no SharedPreferences indicando o nome do arquivo.

**Nota:** Vos dará un erro ata que credes a Activity Principal (ven a continuación).

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Persistencia.Preferencias;
2
3 import android.app.Activity;
4 import android.content.SharedPreferences;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9
10 import es.cursoandroid.cifprodolfoucha.aprendiendo.R;
11
12 public class UD6_01_A2_DatosPersistentesPreferencias extends Activity {
13     private SharedPreferences datosGlobales;
14
15
16     private void cargarPreferencias(){
17
18         datosGlobales = getSharedPreferences(UD05_01_A1_DatosPersistentes_Preferencias.NOME_ARQUIVO_PREFERENCIAS, MODE_PRIVATE);
19
20         EditText texto = (EditText)findViewById(R.id.UD05_01_A2_etDato);
21         String dato = datosGlobales.getString(UD05_01_A1_DatosPersistentes_Preferencias.NOME_PARAM_DATO, "Non hai dato gardado");
22         texto.setText(dato);
23     }
24
25     private void xestionarEventos(){
26
27         Button btnActualizar = (Button)findViewById(R.id.UD05_01_A2_btnActualizar);
28         btnActualizar.setOnClickListener(new View.OnClickListener() {
29
30             @Override
31             public void onClick(View arg0) {
32                 // TODO Auto-generated method stub
33
34                 EditText texto = (EditText)findViewById(R.id.UD05_01_A2_etDato);
35
36                 SharedPreferences.Editor editorGlobal = datosGlobales.edit();
37                 editorGlobal.putString(UD05_01_A1_DatosPersistentes_Preferencias.NOME_PARAM_DATO, texto.getText().toString());
38                 editorGlobal.commit();
39
40
41                 finish();
42             }
43         });
44     }
45
46     @Override
47     protected void onCreate(Bundle savedInstanceState) {
48         super.onCreate(savedInstanceState);
49         setContentView(R.layout.activity_ud6_01_a2_datos_persistentes_preferencias);
50
51         cargarPreferencias();
52         xestionarEventos();
53     }
54 }
55 }

```

- Liña 13: Definimos o SharedPreferences.
- Liña 18: Cargamos o SharedPreferences indicando o nome do arquivo. Dito nome está definido na Activity Principal (definida a continuación)
- Liñas 20-22: Obtemos o dato gardado no SharedPreferences a través do nome dun campo que está definido na Activity Principal (definida a continuación) e o asignamos o EditText.
- Liñas 36-38: Obtemos o editor do SharedPreferences e gardamos o contido da caixa de texto no SharedPreferences, en forma de par (Nome\_param - Valor) sendo Nome\_param o definido na Activity Principal.

## Pantallas de Preferencias

Máis información en:

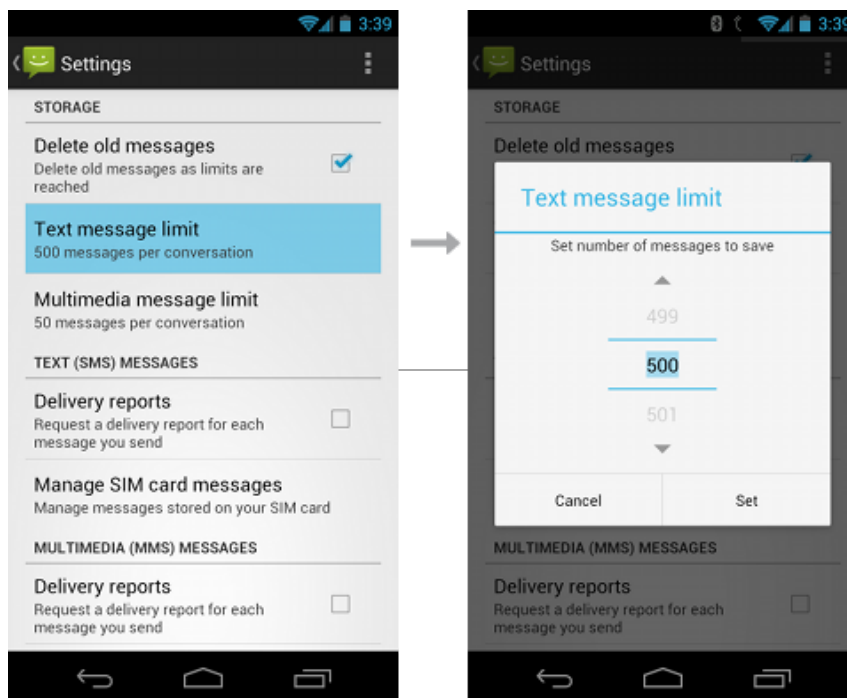
- <http://developer.android.com/guide/topics/ui/settings.html>
- <http://developer.android.com/design/patterns/settings.html>



## Introdución

Relacionado co anterior temos as **Pantallas de Preferencias**.

Se estades acostumados a ver aplicacións podedes observar como as pantallas que conforman os **Settings** ou **Axustes** teñen todas o mesmo aspecto:



Isto o conseguimos facendo uso duns recursos gráficos no que definimos todos os elementos que conforman os nosos axustes (texto, lista, radiobutton,...).

Todas as opcións que escollemos neste tipo de pantallas son gardadas nun `SharedPreferences` igual aos que levamos vistos ata o de agora.

Para obter os datos gardados nas preferencias temos que acceder chamando o método de clase: `PreferenceManager.getDefaultSharedPreferences(Context contexto)` (<http://developer.android.com/reference/android/preference/PreferenceManager.html#getDefaultSharedPreferences%28android.content.Context%29>).

Este devolve un obxecto da clase `SharedPreferences`.

Por exemplo:

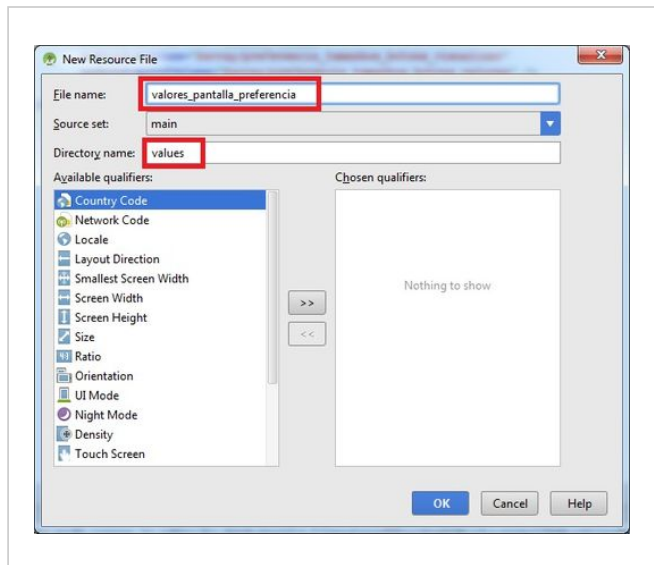
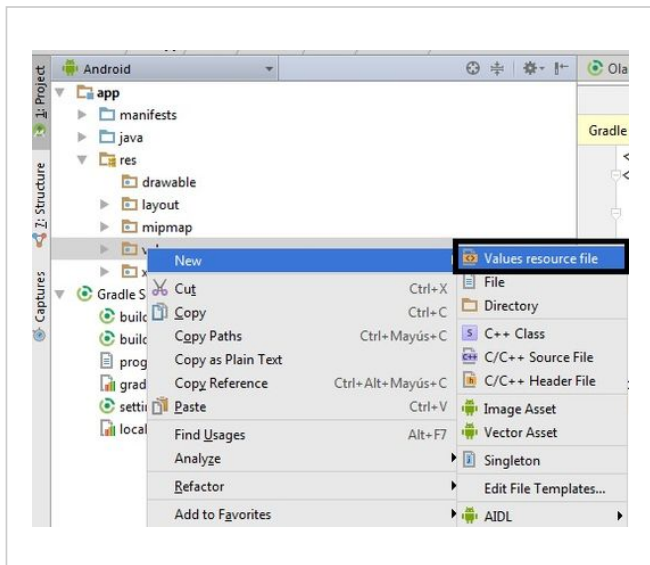
```
SharedPreferences preferencias = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
```

## Valores asociados ás pantallas de preferencias

- As pantallas poden ter campos que alberguen valores (como por exemplo listas) ou caixas de texto que queiramos que teñan un valor por defecto.

É necesario definir ditos valores. Para iso creamos un arquivo de recursos (`/res/values/`) e definimos todos os valores que necesitamos.

### Creando un arquivo values para a pantalla de preferencias



Prememos co rato no botón dereito sobre o cartafol **/res/values**.

Damos un nome ao arquivo 'values' que vai gardar os valores que necesitará a pantalla.

Por exemplo:

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <resources>
3     <string name="preferencia_numero_por_defecto">10</string>
4
5     <string-array name="preferencia_tamamos_botons_visualizar">
6         <item>10 sp</item>
7         <item>12 sp</item>
8         <item>14 sp</item>
9     </string-array>
10    <string-array name="preferencia_tamamos_botons_valores">
11        <item>10</item>
12        <item>12</item>
13        <item>14</item>
14    </string-array>
15    <string name="preferencia_tamano_por_defecto">12</string>
16 </resources>

```

Podemos observar como definimos dous arrays, un é o que vai amosar a lista (preferencia\_tamamos\_botons\_visualizar), será o que o usuario vexa na pantalla, e o outro é o valor que nos imos a obter como programadores para manexar internamente. Así, se temos unha lista de cidades, podemos querer amosar os seus nomes, pero internamente queremos gardar un código.

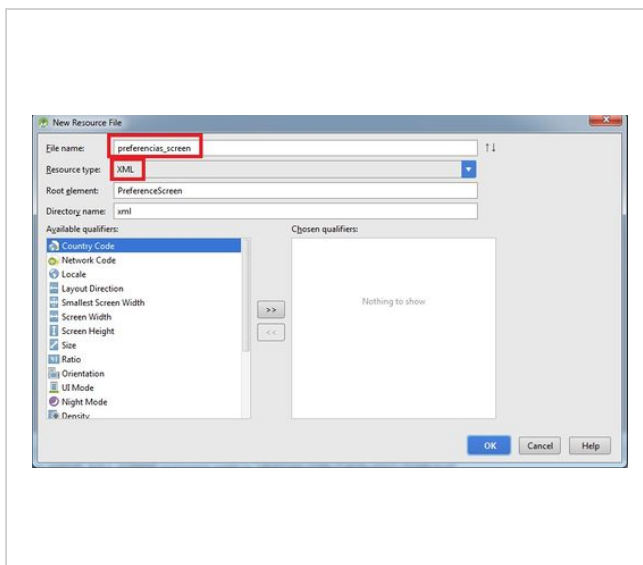
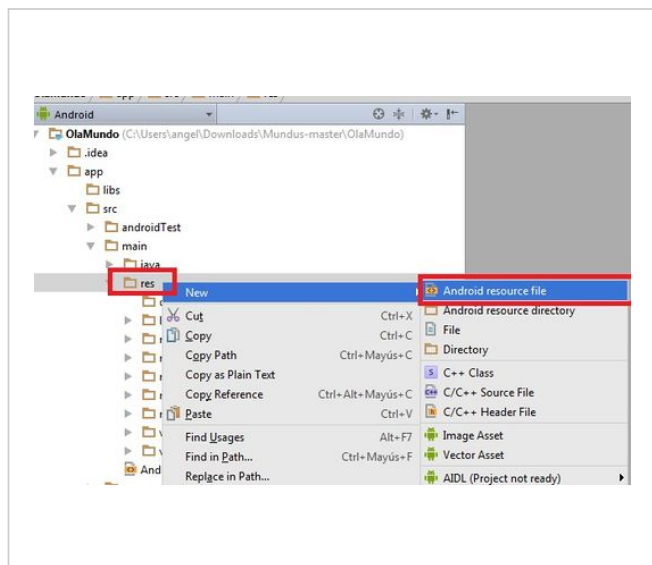
Nota: Isto non é obrigatorio facelo así pode darse o caso que o que visualice a lista sexa o que necesitamos gardar.

## Definindo as pantallas de preferencias

- Definimos a pantalla que vai conformar as nosas preferencias.

Importante: Dito arquivo ten que estar creado no cartafol **/res/xml/**.

### Creando unha pantalla de preferencias



Prememos o botón dereito do rato sobre o cartafol **res** e escollemos a opción **New Android Resource File**.

Damos o nome (ten que ser con letras minúsculas) **preferencias\_screen**, escollemos na opción **Resource Type** a opción **XML**. Prememos o botón **Ok**.

Ao facer o paso segundo crearase no Android Studio un cartafol **/res/xml**.

#### **NOTA IMPORTANTE:**

A partires da API 29, Android vai marcar o código xerado polo asistente como **deprecated**.  
Recomenda facer uso da biblioteca de compatibilidade androidx, como xa vimos nesta Wiki ([https://wiki.cifprodolfoucha.es/index.php?title=PDM\\_UD1\\_Bibliotecas\\_de\\_compatibilidade#Versi.C3.B3n\\_actual](https://wiki.cifprodolfoucha.es/index.php?title=PDM_UD1_Bibliotecas_de_compatibilidade#Versi.C3.B3n_actual)).  
Para que elimine a aviso debemos substituír

```
<PreferenceScreen
    xmlns:app="http://schemas.android.com/apk/res-auto">

</PreferenceScreen>
```

Por

```
<androidx.preference.PreferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android">

</androidx.preference.PreferenceScreen>
```

**Isto teremos que facelo ao final de todo, cando se teñamos deseñado a PreferenceScreen, xa que graficamente, se o facemos antes, non deixa arrastrar os compoñentes da paleta...**

Sigo explicando as PreferenceScreen... Temos que entender a PreferenceScreen coma unha activity e o seu layout.

Por exemplo:

```
1 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
2   android:title="EJEMPLO DE SETTINGS" >
3   <ListPreference
4       android:key="ID_PARAM_TAMAÑO"
5       android:title="Preferencias de tamaño"
6       android:summary="Garda o tamaño das caixas de texto"
7       android:defaultValue="@string/preferencia_tamano_por_defecto"
8       android:dialogTitle="Indicar tamaño"
9       android:entries="@array/preferencia_tamano_botons_visualizar"
10      android:entryValues="@array/preferencia_tamano_botons_valores" />
```

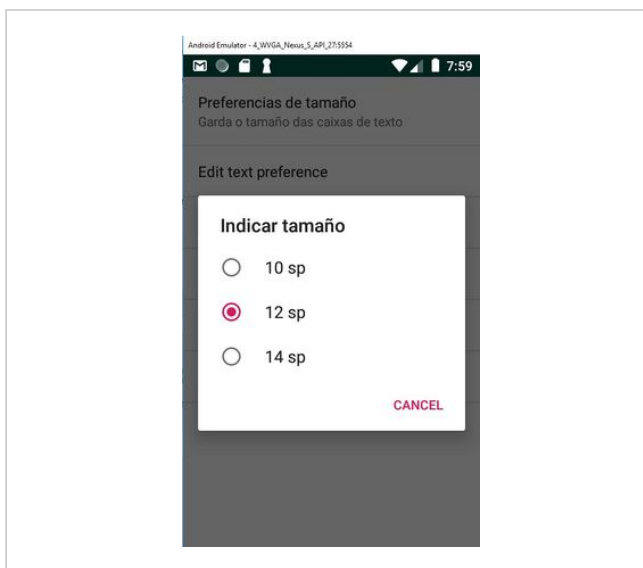
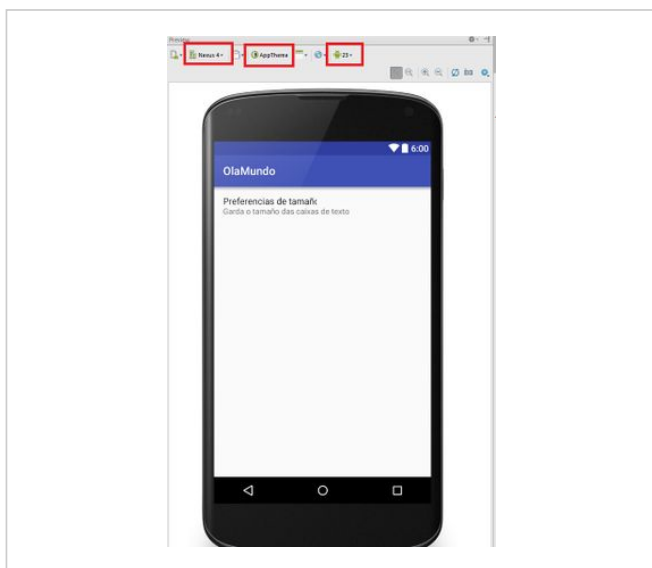
```
11
12 </PreferenceScreen>
```

Imos explicar os atributos utilizados:

- **android:key:** debe ser único e vai ser usado polas SharedPreferences como key para gardar/recuperar o valor (ven ser o nome do parámetro).
- **android:title:** o texto que vai visualizar a pantalla.
- **android:defaultValue:** valor por defecto a cargar polas SharedPreferences. Debería indicarse sempre. No caso das listas que teñen dous tipos de arrays (o que se amosa e os valores) o valor por defecto debe corresponderse cun dos valores definido no array de valores.
- **android:entries:** os valores que vai visualizar a lista.
- **android:entryValues:** os valores que realmente se gardan e que imos usar na programación. Non son obrigatorios telos e pode suceder que o valor que aparece na lista sexa o que necesitamos.
- **android:dialogTitle:** Título do diálogo que vai aparecer cando introduzamos o valor.
- **android:summary:** Texto aclaratorio que aparece debaixo do title da preferencia.

Na parte da dereita do Android Studio podemos observar como quedaría dita pantalla nun dispositivo Android:

### ListPreference



Pantalla de preferencias cun ListPreference. Pódese cambiar o dispositivo, o aspecto (theme) e versión do S.O. Android (API) entre outros parámetros, na parte superior.

Cadro que aparece para introducir a preferencia. Fixarse no título que corresponde coa propiedade 'android:dialogTitle'

Algúns dos elementos gráficos que podemos ter dentro dunha PreferenceScreen son:

- Checkboxpreference (<http://developer.android.com/reference/android/preference/CheckBoxPreference.html>): amosa un checkbox. Garda un valor true/false.

```
1 <CheckBoxPreference
2 android:key="opcion1"
3 android:title="Preferencia 1"
4 android:defaultValue="true"
5 android:summary="Descripción da preferencia 1" />
```

- ListPreference (<http://developer.android.com/reference/android/preference/ListPreference.html>): unha lista de elementos. Xa visto no exemplo.
- EditTextPreference (<http://developer.android.com/reference/android/preference/EditTextPreference.html>): Amosa un diálogo cun EditText. Garda un String.

```
1 <EditTextPreference
2 android:key="opcion2"
3 android:title="Preferencia 2"
4 android:defaultValue="Texto por defecto"
```

```
5 android:summary="Descrición da preferencia 2"
6 android:dialogTitle="Introduce un valor" />
```

- **RingtonePreference** (<https://developer.android.com/reference/android/preference/RingtonePreference>): Para cambiar o tono de son do teléfono.

```
1 <RingtonePreference
2     android:key="ringtone_preference_1"
3     android:title="Ringtone preference" />
4     android:defaultValue="content://settings/system/notification_sound"
5     android:summary="Descrición da preferencia ringtone_preference_1"
6     android:dialogTitle="Título do diálogo" />
```

- **SwitchPreference** (<https://developer.android.com/reference/android/preference/SwitchPreference>): Gráfico tipo switch con dous estados, verdadeiro ou falso.

```
1 <SwitchPreference
2     android:defaultValue="false"
3     android:key="switch_preference_1"
4     android:title="Switch preference" />
```

- **MultiSelectListPreference** (<https://developer.android.com/reference/android/preference/MultiSelectListPreference>): Permite amosar unha lista de elementos con check's asociados, puidendo seleccionar varias.

```
1 <MultiSelectListPreference
2     android:key="multi_select_list_preference_1"
3     android:title="Multi select list preference" />
4     android:defaultValue="@array/empty_array"
5     android:entries="@array/planetas_UD02_01_spinner"
6     android:entryValues="@array/planetas_UD02_01_spinner"
7     android:summary="Descrición da preferencia multi_select_list_preference_1"
8     android:dialogTitle="Título do diálogo" />
```

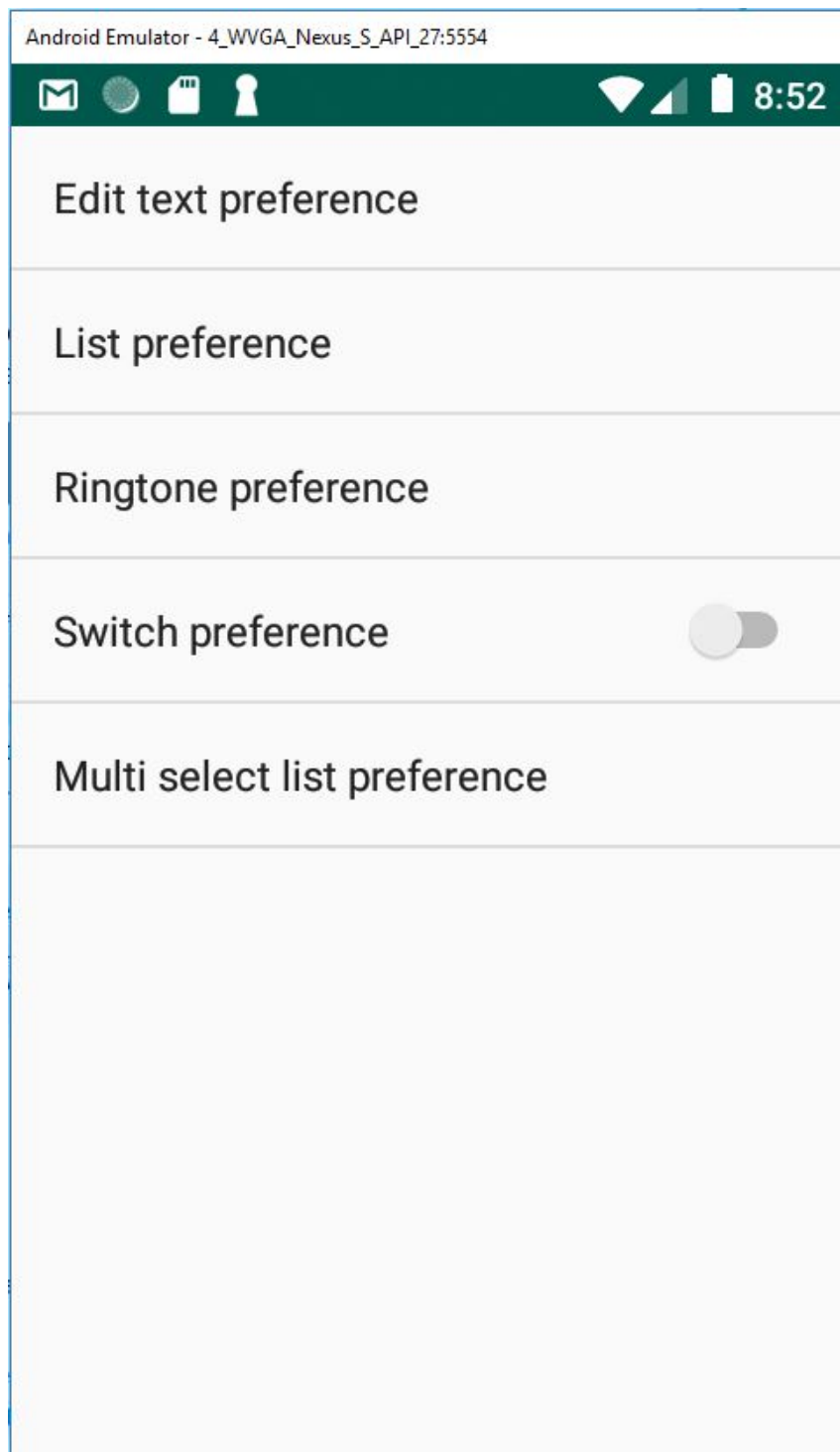
## Caso Práctico 2

- Dentro do paquete **Preferencias** crear unha nova 'Empty Activity' de nome: **UD05\_02\_A1\_PreferenceScreen** de tipo Launcher.

Modifica o arquivo **AndroidManifest.xml** e engade unha label á activity como xa vimos na creación do proxecto base ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

- Iremos facer paso a paso unha pantalla de preferencias de mostra con todos os elementos gráficos que podemos empregar para gardar unha preferencia (sen contar coa opción de definir o noso propio tipo de preferencia).
- Partimos que xa temos creado o proxecto inicial como xa indicamos anteriormente ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proxecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proxecto_base)).

Se non o temos creado antes, crearemos un novo paquete de nome: **Persistencia** como un subpaquete do teu paquete principal.



### Datos da pantalla de preferencia

- Primeiro imos ter que crear un arquivo en `/res/values/` onde imos gardar todos os datos que van a facer uso os elementos que conforman a pantalla de preferencias.

No exemplo imos definir:

- `EditTextPreference`:
  - Un default value de tipo String.
- `ListPreference`:
  - Un array para os valores que vai a amosar ao usuario.
  - Un array para os valores que imos gardar na preferencia.
  - Un valor por defecto.
- `RingtonePreference`:

- Un default value de tipo String.
- SwitchPreference:
  - Nada
- MultiSelectListPreference:
  - Un array para os valores que vai a amosar ao usuario.
  - Un array para os valores que imos gardar na preferencia.
  - Un array para os valores marcados por defecto.

### Arquivo res/values/valores\_preferencias\_UD05\_02.xml

```

1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <resources>
4
5   <string name="preferencia_edittext_por_defecto">Valor por defecto</string>
6
7   <string-array name="preferencia_tamanhos_botons_visualizar">
8     <item>10 sp</item>
9     <item>12 sp</item>
10    <item>14 sp</item>
11  </string-array>
12  <string-array name="preferencia_tamanhos_botons_valores">
13    <item>10</item>
14    <item>12</item>
15    <item>14</item>
16  </string-array>
17  <string name="preferencia_tamanho_boton_por_defecto">12</string>
18
19  <string name="preferencia_tono_por_defecto">content://settings/system/notification_sound</string>
20
21  <string-array name="preferencia_lista_lugares">
22    <item>SANTIAGO</item>
23    <item>MADRID</item>
24    <item>BARCELONA</item>
25  </string-array>
26
27  <string-array name="preferencia_valores_lista_lugares">
28    <item>cod_sant</item>
29    <item>cod_madrid</item>
30    <item>cod_barcelona</item>
31  </string-array>
32
33  <string-array name="preferencia_valores_por_defecto_lista_lugares">
34    <item>cod_sant</item>
35    <item>cod_barcelona</item>
36  </string-array>
37
38 </resources>

```

### Definición da pantalla de preferencias

- Agora definimos a pantalla de preferencia. O podemos facer graficamente como vimos antes ou directamente no código XML.

### Arquivo /res/xml/pantalla\_preferencias\_UD05\_02.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
3
4   <EditTextPreference
5     android:key="edit_text_preference_1"
6     android:title="Edit text preference"
7     android:defaultValue="@string/preferencia_edittext_por_defecto"
8     android:selectAllOnFocus="true"
9     android:singleLine="true"
10    />
11  <ListPreference
12    android:key="list_preference_1"
13    android:title="List preference"
14    android:defaultValue="@string/preferencia_tamanho_boton_por_defecto"
15    android:entries="@array/preferencia_tamanhos_botons_visualizar"

```

```

16     android:entryValues="@array/preferencia_tamanhos_botons_valores"
17     />
18     <RingtonePreference
19         android:defaultValue="@string/preferencia_tono_por_defecto"
20         android:key="ringtone_preference_1"
21         android:title="Ringtone preference" />
22     <SwitchPreference
23         android:defaultValue="false"
24         android:key="switch_preference_1"
25         android:title="Switch preference" />
26     <MultiSelectListPreference
27         android:defaultValue="@array/preferencia_valores_por_defecto_lista_lugares"
28         android:entries="@array/preferencia_lista_lugares"
29         android:entryValues="@array/preferencia_valores_lista_lugares"
30         android:key="multi_select_list_preference_1"
31         android:title="Multi select list preference" />
32 </PreferenceScreen>

```

- Nota: Non están todas as propiedades, coma summary ou dialogtitle, que se poderían empregar.

### Carga da pantalla de preferencias

- Existen varias formas de cargar a pantalla de preferencia.

Nos imos ver unha delas, pero vos podedes encontrar código que o fai diferente.

Podedes consultar neste enlace (<http://www.hermosaprogramacion.com/2015/04/como-crear-actividad-preferencias-android/>) como facelo de outras formas (e hai máis).

- Primeiro debemos crear unha clase que derive da clase 'PreferenceFragmentCompat' e que vai ser a clase que cargue a PreferenceScreen (agora mesmo a clase PreferenceFragment está deprecated).

#### Clase UD05\_02\_A2\_MySettingsFragment:

```

public class UD05_02_A2_MySettingsFragment extends PreferenceFragmentCompat {
    @Override
    public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
        setPreferencesFromResource(R.xml.pantallapreferencias, rootKey);
    }
}

```

- Agora debemos crear unha Activity que sustitúa ou seu contido, polo contido da clase anterior.

Para iso é necesario facer uso dos Fragment.

Esta activity **non ten un content view asociado**, e polo tanto cando a credes, desmarcade a opción 'Generate Layout File'

#### Activity UD05\_02\_A3\_SettingsActivity:

```

public class UD05_02_A3_SettingsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getSupportFragmentManager().beginTransaction()
            .add(android.R.id.content, new UD05_02_A2_MySettingsFragment())
            .commit();
    }
}

```

- Agora queda crear unha Activity coma as de sempre e chamar á activity anterior. O podemos facer ao premer unha opción do menú ou con calquera outro evento.

Na carga de dita activity, imos darlle todos os valores por defecto as propiedades, chamando ao método 'PreferenceManager.setDefaultValues'.

Neste exemplo imos poñer un botón que ao premer, amosa a pantalla de preferencias.

### Layout activity\_UD05\_02\_a1\_preference\_screen

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

```



```

3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".UD6.Preferencias.UD05_02_A1_PreferenceScreen">
8
9   <Button
10      android:id="@+id/btnPreferencias_UD05_02_A1"
11      android:layout_width="wrap_content"
12      android:layout_height="wrap_content"
13      android:text="Preferencias"
14      app:layout_constraintBottom_toBottomOf="parent"
15      app:layout_constraintEnd_toEndOf="parent"
16      app:layout_constraintHorizontal_bias="0.5"
17      app:layout_constraintStart_toStartOf="parent"
18      app:layout_constraintTop_toTopOf="parent" />
19 </android.support.constraint.ConstraintLayout>

```

### Código da Activity UD05\_02\_A1\_PreferenceScreen:

```

1 public class UD05_02_A1_PreferenceScreen extends AppCompatActivity {
2
3     private void xestionarEventos(){
4         findViewById(R.id.btnPreferencias_UD05_02_A1).setOnClickListener(new View.OnClickListener() {
5             @Override
6             public void onClick(View v) {
7
8                 Intent intento = new Intent(getApplicationContext(),UD05_02_A3_SettingsActivity.class);
9                 startActivity(intento);
10            }
11        });
12    }
13    @Override
14    protected void onCreate(Bundle savedInstanceState) {
15        super.onCreate(savedInstanceState);
16        setContentView(R.layout.activity_UD05_02_a1_preference_screen);
17
18        xestionarEventos();
19        PreferenceManager.setDefaultValues(this,R.xml.pantalla_preferencias_UD05_02,false);
20    }
21 }

```

## Definindo categorías dentro das Pantallas de Preferencias

Podemos agrupar as preferencias por categorías.

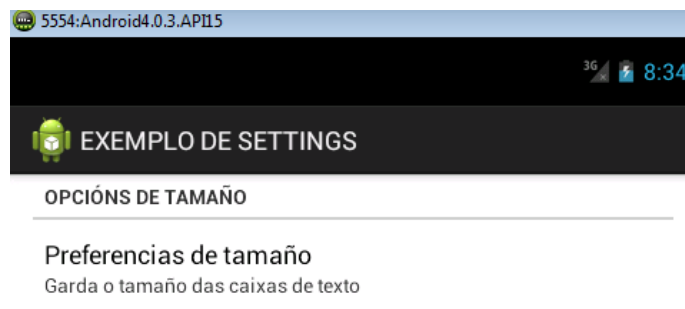
Por exemplo o seguinte arquivo de PreferenceScreen:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" android:title="EXEMPLO DE SETTINGS">
3     <PreferenceCategory android:title="OPCIÓN DE TAMAÑO">
4         <ListPreference
5
6             android:key="@string/preferencia_tamanho_nome_parametro"
7             android:title="Preferencias de tamaño"
8             android:summary="Garda o tamaño das caixas de texto"
9             android:dialogTitle="Indicar tamaño"
10            android:defaultValue="@string/preferencia_tamanho_por_defecto"
11            android:entries="@array/preferencia_tamanhos_botons"
12            android:entryValues="@array/preferencia_tamanhos_botons_valores" />
13     </PreferenceCategory>
14 </PreferenceScreen>

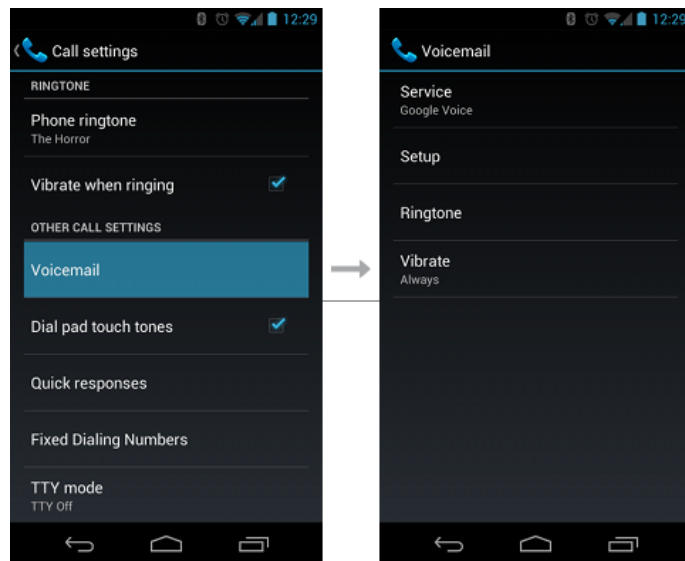
```

Da como resultado:



## Creando novas pantallas de opcións

Tamén podemos crear novas pantallas de preferencias dentro de pantallas de preferencias, de tal forma que ao pulsar unha preferencia determinada abra unha nova pantalla.



Imaxe obtida de <http://developer.android.com/guide/topics/ui/settings.html>

Isto se consegue cun PreferenceScreen dentro doutro.

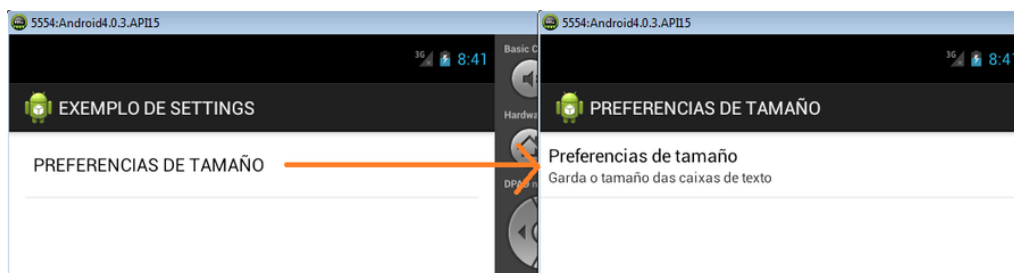
Por exemplo o seguinte arquivo de PreferenceScreen:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3     android:title="EXEMPLO DE SETTINGS" >
4     <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
5         android:title="PREFERENCIAS DE TAMAÑO">
6         <ListPreference
7
8             android:key="@string/preferencia_tamanho_nome_parametro"
9             android:title="Preferencias de tamaño"
10            android:summary="Garda o tamaño das caixas de texto"
11            android:dialogTitle="Indicar tamaño"
12            android:defaultValue="@string/preferencia_tamanho_por_defecto"
13            android:entries="@array/preferencia_tamanhos_botons"
14            android:entryValues="@array/preferencia_tamanhos_botons_valores" />
15        </PreferenceScreen>
16    </PreferenceScreen>
17 </PreferenceScreen>

```

Dará como resultado:



### Caso Práctico 3

- Partimos que xa temos creado o proxecto inicial como xa indicamos anteriormente ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proyecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proyecto_base)).

Se non o temos creado antes, crearemos un novo paquete de nome: **Preferencias** como un subpaquete do teu paquete principal.

- Dentro do paquete **Preferencias** crear unha nova 'Empty Activity' de nome: **UD05\_03\_DatosPersistentes\_Preferencias** de tipo Launcher.

Modifica o arquivo **AndroidManifest.xml** e engade unha label á activity como xa vimos na creación do proxecto base ([http://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Creando\\_proyecto\\_base](http://wiki.cifprodolfoucha.es/index.php?title=PDM_Creando_proyecto_base)).

- O obxectivo desta práctica é crear unha pantalla de preferencias onde imos poder escoller se queremos que un texto que visualice en cursiva e cunha cor determinada.

Accederemos aos valores escollidos nesa pantalla dende unha activity e aplicaremos o aspecto a un texto.

Imos ver que para coller estes valores, imos sobreescribir o método `onResume()` o cal se chama de forma automática cando voltamos de acceder á activity de preferencias.

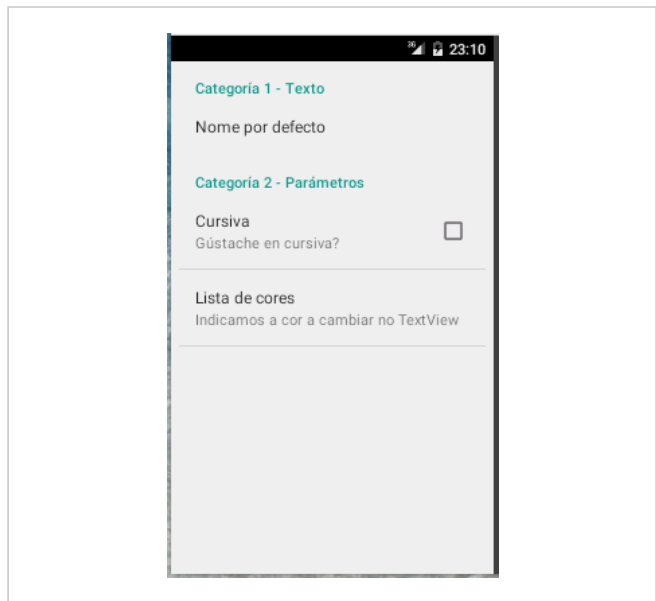
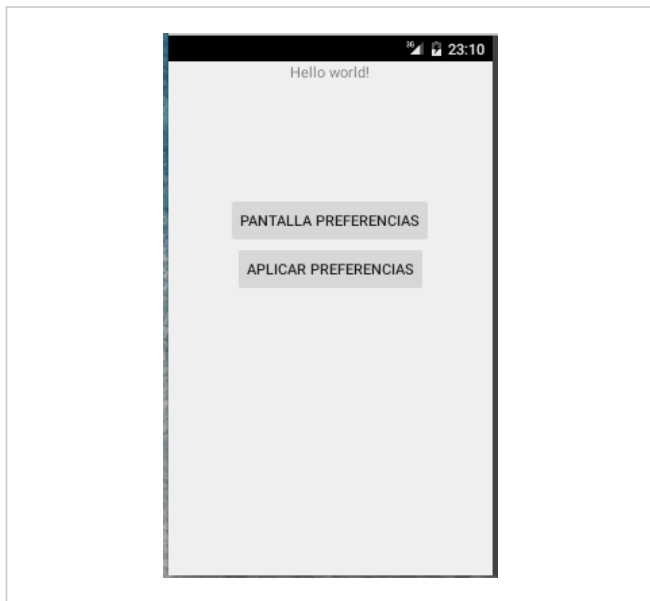
O malo desta solución é que sempre imos a aplicar as preferencias aínda que non se produzan cambios e se volvemos doutras activities que non sexa a de preferencias.

Consta de dúas activities:

- Activity principal: Dende onde lanzamos a pantalla de preferencias e dende onde accedemos aos valores escollidos.
- Activity secundaria: Carga a pantalla de preferencias.

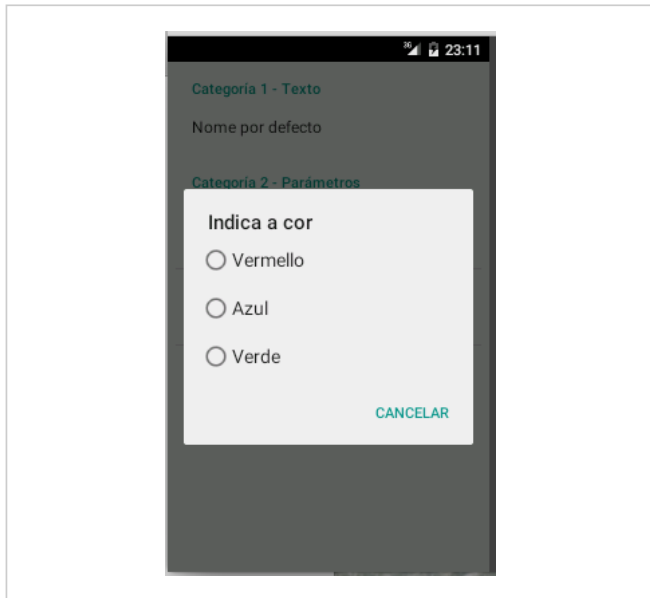
O aspecto desta práctica é o seguinte:

#### Uso dunha pantalla de preferencias

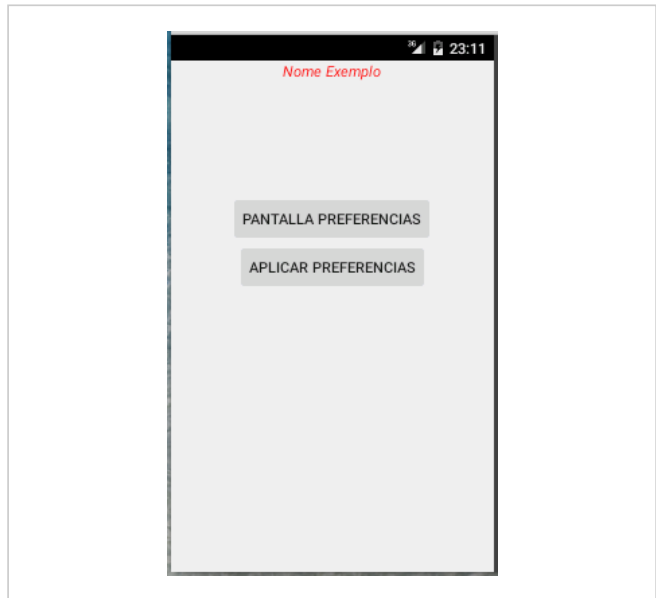


Pantalla principal. Aquí podemos lanzar a pantalla de preferencias ou cargar os valores escollidos en dita pantalla. O nome amosarase nun TextView na parte superior e a cor será indicada por unha mensaxe (Toast).

Segunda activity. É a pantalla de preferencias. Gardamos un nome e escollemos unha cor.



Esta é a lista coas cores da pantalla de preferencias.



Unha vez seleccionada a cor vermella e cursiva e non poñendo ningún nome, aplicamos as preferencias.

### Arquivo de valores

Como comentamos anteriormente as pantallas de preferencias poden necesitar datos (como no caso das listas) e valores por defecto.

Todos estes valores deben estar definidos nun arquivo xml no cartafol /res/values/

### Código do arquivo /res/values/ud6\_o3\_datospreferencias.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="UD6_o3_preferencia_valor_nome_por_defecto">Nome Exemplo</string>
4   <string-array name="UD6_o3_preferencia_textos_lista">
5     <item>Vermello</item>
6     <item>Azul</item>
7     <item>Verde</item>
8   </string-array>
9   <string-array name="UD6_o3_preferencia_valores_lista">
10    <item>#ff0000</item>

```

```

11         <item>#0000ff</item>
12         <item>#00ff00</item>
13     </string-array>
14 </resources>

```

## Pantalla de Preferencias

Agora necesitamos definir a nosa pantalla de preferencias.

Lembrar que é un arquivo xml que se atopa en **/res/xml/**

### Código do arquivo **/res/xml/pantalla\_preferencias\_UD05\_03.xml**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
3     android:title="EJEMPLO DE SETTINGS">
4
5     <PreferenceCategory
6         android:key="Categoria"
7         android:title="Categoría 1 - Texto">
8         <EditTextPreference
9             android:defaultValue="@string/UD6_03_preferencia_valor_nombre_por_defecto"
10             android:key="UD6_03_preferencia_idNombre"
11             android:selectAllOnFocus="true"
12             android:title="Nombre por defecto" />
13     </PreferenceCategory>
14
15     <PreferenceCategory android:title="Categoría 2 - Parámetros">
16         <CheckBoxPreference
17             android:defaultValue="False"
18             android:key="CURSIVA"
19             android:summary="¿Gústache en cursiva?"
20             android:title="Cursiva" />
21
22         <ListPreference
23             android:dialogTitle="Indica a cor"
24             android:entries="@array/UD6_03_preferencia_textos_lista"
25             android:entryValues="@array/UD6_03_preferencia_valores_lista"
26             android:key="UD6_03_preferencia_listaCores"
27             android:summary="Indicamos a cor a cambiar no TextView"
28             android:title="Lista de cores" />
29     </PreferenceCategory>
30
31 </PreferenceScreen>

```

Nota: Os valores dos identificadores dos campos (android:key) están postos directamente no código, pero o ideal é definilos nun arquivo de recursos de tipo String e referencialos da forma android:key="@string/....

## Activity Principal

### Código XML do layout: **/res/layout/activity\_UD05\_03\_DatosPersistentes\_Preferencias.xml**

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     tools:context=".Persistencia.Preferencias.UD05_03_datospersistentes_preferencias" >
6
7     <TextView
8         android:id="@+id/UD6_03_txtTexto"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:layout_alignParentTop="true"
12        android:layout_centerHorizontal="true"
13        android:text="@string/hello_world" />
14
15     <Button
16         android:id="@+id/UD6_03_btnPantallaPreferencias"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:layout_below="@+id/UD6_03_txtTexto"
20         android:layout_centerHorizontal="true"
21         android:layout_marginTop="113dp"
22         android:text="Pantalla Preferencias" />
23
24     <Button
25         android:id="@+id/UD6_03_btnAplicarPreferencias"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:layout_below="@+id/UD6_03_btnPantallaPreferencias"
29         android:layout_centerHorizontal="true"
30         android:text="Aplicar Preferencias" />
31
32 </RelativeLayout>

```

## O Código Java

**Obxectivo:** Utilizar unha pantalla de preferencias.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Persistencia.Preferencias
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.graphics.Color;
7 import android.graphics.Typeface;
8 import android.os.Bundle;
9 import android.preference.PreferenceManager;
10 import android.view.View;
11 import android.view.View.OnClickListener;
12 import android.widget.Button;
13 import android.widget.TextView;
14 import android.widget.Toast;
15
16 public class UD05_03_DatosPersistentes_Preferencias extends AppCompatActivity {
17
18
19     private void xestionarEventos() {
20
21         Button btnPantallaPreferencias = (Button) findViewById(R.id.UD6_03_btnPantallaPreferencias);
22         btnPantallaPreferencias.setOnClickListener(new OnClickListener() {
23
24             @Override
25             public void onClick(View v) {
26                 // TODO Auto-generated method stub
27                 Intent intent = new Intent(getApplicationContext(), UD05_03_DatosPersistentes_PantallaPreferencias.class);
28                 startActivity(intent);
29             }
30         });
31         Button btnAplicarPreferencias = (Button) findViewById(R.id.UD6_03_btnAplicarPreferencias);
32         btnAplicarPreferencias.setOnClickListener(new OnClickListener() {
33
34             @Override
35             public void onClick(View v) {
36                 // TODO Auto-generated method stub
37                 aplicarPreferencias();
38             }
39         });
40     }
41
42 }
43
44 private void aplicarPreferencias() {
45     SharedPreferences preferencias = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
46
47     String valorNome = preferencias.getString("UD6_03_preferencia_idNome", "SEN VALOR NO NOME");
48     TextView nome = (TextView) findViewById(R.id.UD6_03_txtTexto);
49     nome.setText(valorNome);
50
51     String corDefecto = preferencias.getString("UD6_03_preferencia_listaCores", "#000000");
52     Toast.makeText(getApplicationContext(), corDefecto, Toast.LENGTH_LONG).show();
53     nome.setTextColor(Color.parseColor(corDefecto));
54
55     if (preferencias.getBoolean("CURSIVA", false))
56         nome.setTypeface(null, Typeface.ITALIC);
57     else
58         nome.setTypeface(null, Typeface.NORMAL);
59 }
60
61 @Override
62 protected void onCreate(Bundle savedInstanceState) {
63     super.onCreate(savedInstanceState);
64     setContentView(R.layout.activity_UD05_03_datospersistentes_preferencias);
65
66     xestionarEventos();
67 }
68 }

```

- Liñas 27-28: Lanzamos a segunda activity que carga a pantalla de preferencias.
  - Liña 45: Cargamos o SharedPreferences utilizado pola pantalla de preferencias.
  - Liñas 47-49: Cargamos o nome gardado na pantalla de preferencias no TextView da activity principal.
  - Liñas 51-53: Amosamos a cor escollida na pantalla de preferencias nun Toast e cambiamos a cor do nome.
  - Liñas 55-58: Aplicamos ao nome cursiva ou normal en función do escollido nas preferencias.
- 
- Liña 44: Se chamamos a este método en **onResume()** xa se aplicarían as preferencias cando volveramos da pantalla de preferencias e tamén cando se cargue a aplicación a vindeira vez.

## Creamos a Activity Secundaria

- Nome da activity: **UD05\_03\_DatosPersistentes\_PantallaPreferencias**

Esta activity terá como obxectivo cargar a pantalla de preferencias definida no arquivo xml de preferencias anterior.

A diferenza do caso práctico anterior vemos que a clase que deriva de PreferenceFragment está definida dentro da propia Activity.

Pódese facer así ou como está feito no exemplo anterior, definindo a clase 'SettingsFragment' nun arquivo diferente.

### Código Java

**Obxectivo:** Carga a pantalla de preferencias definida anteriormente.

```

1 package es.cursoandroid.cifprodolfoucha.aprendiendo.Persistencia.Preferencias
2 import androidx.appcompat.app.AppCompatActivity;
3 import androidx.preference.PreferenceFragmentCompat;
4 import androidx.preference.PreferenceManager;
5
6 public class UD05_03_DatosPersistentes_PantallaPreferencias extends AppCompatActivity{
7
8     public static class SettingsFragment extends PreferenceFragmentCompat{
9         @Override
10         public void onCreate(Bundle savedInstanceState) {
11             super.onCreate(savedInstanceState);
12
13             // Load the preferences from an XML resource
14             addPreferencesFromResource(R.xml.pantalla_preferencias_UD05_03);
15         }
16     }
17
18     @Override
19     public void onCreate(Bundle savedInstanceState) {
20         // TODO Auto-generated method stub
21         super.onCreate(savedInstanceState);
22
23         getSupportFragmentManager().beginTransaction()
24             .replace(android.R.id.content, new SettingsFragment())
25             .commit();
26     }
27
28 }

```

- Liñas 8-16: Creamos unha clase interna que derive de PreferenceFragment e que cargue o arquivo onde está definida a pantalla de preferencias.
- Liñas 23-24: Fixarse como nesta activity non cargamos un layout no método onCreate. O que facemos nestas liñas é substituír o contido da activity por un obxecto da clase SettingsFragment que é onde se atopa cargada a pantalla de preferencias.
- Có comando adb ou có Device File Explorer, pódese ir ao directorio shared\_prefs da aplicación e ver o contido do ficheiro xml de preferencias.

- Indicar que se queremos recoller o valor dunha preferencia de tipo 'MultiSelectListPreference' teremos que facelo desta forma:

```

1     SharedPreferences sharedPreferences = PreferenceManager.getDefaultSharedPreferences(this);
2     ArrayList<String>datos = new ArrayList<String>(sharedPreferences.getStringSet("multi_select_list_preference_1",new
HashSet<String>()));

```

Sendo 'multi\_select\_list\_preference\_1' o id da opción de preferencia.

### Controlando cando cambia algunha preferencia

- Como vimos no exemplo anterior, sempre se vai chamar ao método onResume() da Activity.

Dependendo da nosa aplicación pode ser que non necesitemos empregar dito método e que os axustes se apliquen cando se entra nunha nova activity ou ao premer algún compoñente gráfica da activity principal.

- Se necesitamos aplicar un cambio estando na pantalla de preferencias, podemos facelo da seguinte forma:
- Primeiro debemos de modificar a clase que deriva de PreferenceFragment e implemtaremos a interface que xestiona calquera cambio sobre as preferencias: `SharedPreferences.OnSharedPreferenceChangeListener` (<https://developer.android.com/reference/android/content/SharedPreferences.OnSharedPreferenceChangeListener>).

Ao implementar dita interface teremos que sobreescribir o método `onSharedPreferenceChanged`:

```
1 public class UD05_03_DatosPersistentes_PantallaPreferencias extends AppCompatActivity {
2     .....
3     public static class SettingsFragment extends PreferenceFragmentCompat implements
4     SharedPreferences.OnSharedPreferenceChangeListener {
5         .....
6         @Override
7         public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
8             Toast.makeText(getActivity(), "Cambiada a propiedad " + key + " ao valor " +
9             sharedPreferences.getString(key, ""), Toast.LENGTH_LONG).show();
10        }
11    }
```

Dito método ten dous parámetros.

- Un obxecto `SharedPreferences` onde están gardadas todas as preferencias da pantalla.
- Un `String` `key` que ven ser o identificador de cada un das preferencias.

Nota: Fixarse que dentro da clase 'SettingsFragment' podemos obter unha referencia a activity chamando ao método `getActivity()` e facendo un 'cast' a `UD05_03_DatosPersistentes_PantallaPreferencias` podemos chamar a calquera método da mesma.

No exemplo estamos facendo un `Toast` e chamando ao método `getString`. Indicar que isto o podemos facer coas preferencias que sexan deste tipo. Se temos unha preferencia de tipo `MultiSelectListPreference`, `booleana`,... a aplicación fallará. Nos como programadores, teremos que saber que queremos facer cando cambie unha determinada `key` e chamar ao método correspondente.

- Como con todas as interfaces temos que facer que a clase 'SettingsFragment' se registre para capturar este tipo de eventos.

Para facelo podemos sobreescribir os métodos `onResume()` para rexistrarse e o método `onPause()` para desrexistrarse:

```
1 public class UD05_03_DatosPersistentes_PantallaPreferencias extends AppCompatActivity {
2     .....
3     public static class SettingsFragment extends PreferenceFragmentCompat implements
4     SharedPreferences.OnSharedPreferenceChangeListener {
5         .....
6         @Override
7         public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
8             setPreferencesFromResource(R.xml.pantallapreferencias, rootKey);
9         }
10
11        @Override
12        public void onResume() {
13            super.onResume();
14            getPreferenceScreen().getSharedPreferences().registerOnSharedPreferenceChangeListener(this);
15        }
16
17        @Override
18        public void onPause() {
19            super.onPause();
20            getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(this);
21        }
22
23        @Override
24        public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
25            Toast.makeText(getActivity(), "Cambiada a propiedad " + key + " ao valor " +
26            sharedPreferences.getString(key, ""), Toast.LENGTH_LONG).show();
27            //((UD05_03_DatosPersistentes_PantallaPreferencias)getActivity()).valorCambiado(); // Podemos chamar a un método da
28            Activity
29        }
30    } // Acaba a clase SettingsFragment
31    public void valorCambiado(){
32        Toast.makeText(this, "valor cambiado", Toast.LENGTH_SHORT).show();
33    }
34 }
```



```

31 }
32
33 @Override
34 public void onCreate(Bundle savedInstanceState) {
35     // TODO Auto-generated method stub
36     super.onCreate(savedInstanceState);
37     getSupportFragmentManager().beginTransaction().replace(android.R.id.content,
38         new SettingsFragment()).commit();
39 }
40 }

```

- **Nota:** Lembrar que vimos na unidade UD6, ventás de diálogos ([https://wiki.cifprodolfoucha.es/index.php?title=Vent%C3%A1s\\_de\\_Di%C3%A1logos#Devolvendo\\_informaci%C3%B3n\\_dunha\\_forma\\_m%C3%A1is\\_elegante](https://wiki.cifprodolfoucha.es/index.php?title=Vent%C3%A1s_de_Di%C3%A1logos#Devolvendo_informaci%C3%B3n_dunha_forma_m%C3%A1is_elegante)) que había unha forma máis correcta de chamar a unha activity dende outra clase, mediante o emprego de Interfaces.

## Chamando a Activity principal dende a pantalla de preferencias

- Se necesitamos facer algún tipo de acción cando se cambie algunha opción na pantalla de preferencias, necesitaremos chamar a algún método da activity principal.

O problema o temos en que no exercicio anterior, podemos ter unha referencia a activity que visualiza a pantalla de preferencias, pero non temos unha referencia de quen a chama.

Unha posible solución é empregar unha Interface da seguinte forma:

```

1 interface MiInterface extends Serializable{
2     public void pasoDatos(String key, String valor);
3 }

```

Definimos na interface os métodos que queiramos empregar para enviar datos dende a pantalla de preferencias.

**Fixarse que extende de Serializable** xa que imos 'enviala' como un obxecto no Intent.

- Implementamos dita interface na activity principal:

```

1 public class UD05_03_DatosPersistentes_Preferencias extends AppCompatActivity implements MiInterface {
2     ....
3
4     @Override
5     public void pasoDatos(String key, String valor) {
6         // Non se pode facer o Toast xa que visualmente non está a activity principal.
7         Log.i("INFORMACION", "Cambiada a propiedad " + key + " ao valor " + valor);
8     }
9 }

```

- Agora necesitamos 'enviar' a interface a activity que vai cargar a pantalla de preferencias.

```

1     .....
2     private void xestionarEventos() {
3         Button btnPantallaPreferencias = (Button) findViewById(R.id.UD6_03_btnPantallaPreferencias);
4         btnPantallaPreferencias.setOnClickListener(new View.OnClickListener() {
5             @Override
6             public void onClick(View v) {
7                 // TODO Auto-generated method stub
8                 Intent intent = new Intent(getApplicationContext(), UD05_03_DatosPersistentes_PantallaPreferencias.class);
9                 intent.putExtra("MIINTERFACE", UD05_03_DatosPersistentes_Preferencias.this);
10                 startActivity(intent);
11             }
12         });
13     }

```

- Ahora na Activity que carga a pantalla de preferencias, recollemos dito dato nunha propiedade local no método onCreate e xa podemos empregalo dentro da clase que xestioana cando hai un cambio nas preferencias...

```

1 public class UD05_03_DatosPersistentes_PantallaPreferencias extends AppCompatActivity {
2
3     private static MiInterface miClaseInterface;
4
5     public static class SettingsFragment extends PreferenceFragmentCompat implements
6     SharedPreferences.OnSharedPreferenceChangeListener {
7
8         @Override
9         public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
10
11             setPreferencesFromResource(R.xml.pantallapreferencias, rootKey);
12
13         }
14
15         @Override
16         public void onResume() {
17             super.onResume();
18             getPreferenceScreen().getSharedPreferences().registerOnSharedPreferenceChangeListener(this);
19         }
20
21         @Override
22         public void onPause() {
23             super.onPause();
24             getPreferenceScreen().getSharedPreferences().unregisterOnSharedPreferenceChangeListener(this);
25         }
26
27         @Override
28         public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
29             //Toast.makeText(getActivity(), "Cambiada a propiedade " + key + " ao valor " +
30             sharedPreferences.getString(key, ""), Toast.LENGTH_LONG).show();
31             //((UD05_03_DatosPersistentes_PantallaPreferencias) getActivity()).valorCambiado();
32
33             miClaseInterface.pasoDatos(key, sharedPreferences.getString(key, ""));
34
35         }
36     }
37
38     public void valorCambiado(){
39         Toast.makeText(this, "valor cambiado", Toast.LENGTH_SHORT).show();
40     }
41
42     @Override
43     public void onCreate(Bundle savedInstanceState) {
44         // TODO Auto-generated method stub
45         super.onCreate(savedInstanceState);
46
47         miClaseInterface = (MiInterface) getIntent().getExtras().getSerializable("MIINTERFACE");
48
49         getFragmentManager().beginTransaction().replace(android.R.id.content,
50             new SettingsFragment()).commit();
51     }
52 }

```

**[Enlace a la página principal de la UD5 \(https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n\\_de\\_dispositivos\\_m%C3%B3viles#UNIDADE\\_5:\\_Datos\\_Persistentes\)](https://wiki.cifprodolfoucha.es/index.php?title=Programaci%C3%B3n_de_dispositivos_m%C3%B3viles#UNIDADE_5:_Datos_Persistentes)**

**[Enlace a la página principal del curso \(https://wiki.cifprodolfoucha.es/index.php?title=Programación\\_de\\_dispositivos\\_móviles\)](https://wiki.cifprodolfoucha.es/index.php?title=Programación_de_dispositivos_móviles)**

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2014).

---

Obtenido de «[https://wiki.cifprodolfoucha.es/index.php?title=PDM\\_Avanzado\\_Datos\\_Persistentes\\_Preferencias&oldid=12115](https://wiki.cifprodolfoucha.es/index.php?title=PDM_Avanzado_Datos_Persistentes_Preferencias&oldid=12115)»

---

Esta página se editó por última vez el 17 ene 2021 a las 17:13.