

# 1 PDM Avanzado Datos Persistentes Preferencias

## 1.1 Sumario

- 1 Introducción
- 2 Preferencias
  - ◆ 2.1 Acceso ás preferencias
  - ◆ 2.2 Gardando novos datos
  - ◆ 2.3 Recuperando datos
- 3 Caso Práctico
  - ◆ 3.1 Creamos a Activity Principal
  - ◆ 3.2 Creamos a Activity Secundaria
- 4 Pantallas de Preferencias
  - ◆ 4.1 Introducción
  - ◆ 4.2 Valores asociados ás pantallas de preferencias
  - ◆ 4.3 Definindo as pantallas de preferencias
  - ◆ 4.4 Definindo categorías dentro das Pantallas de Preferencias
  - ◆ 4.5 Creando novas pantallas de opcións
  - ◆ 4.6 Caso Práctico
    - ◇ 4.6.1 Arquivo de valores
    - ◇ 4.6.2 Pantalla de Preferencias
    - ◇ 4.6.3 Activity Principal
    - ◇ 4.6.4 Creamos a Activity Secundaria

## 1.2 Introducción

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`.

## 1.3 Preferencias

### 1.3.1 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 actividades 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 a perspectiva DDMS de Eclipse có emulador.

### 1.3.2 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();
```

### 1.3.3 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.

## 1.4 Caso Práctico

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.

## 1.4.1 Creamos a Activity Principal

- Nome do proxecto: **UD1\_01\_DatosPersistentes\_Preferencias**
- Nome da activity: **UD1\_01\_DatosPersistentes\_Preferencias**

### Código do layout xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="{relativePackage}.{activityClass}" >

    <EditText
        android:id="@+id/UD1_01_txtDatoGardarLocal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:inputType="text"
        android:hint="Dato a gardar localmente"
        android:textSize="15sp"
        android:layout_centerHorizontal="true"
    >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/UD1_01_btnGardarDatos"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="GARDAR DATOS" />

    <Button
        android:id="@+id/UD1_01_btnCargarDatos"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/UD1_01_btnGardarDatos"
        android:layout_below="@+id/UD1_01_btnGardarDatos"
        android:text="CARGAR DATOS" />

    <Button
        android:id="@+id/UD1_01_btnLanzarActivitySegunda"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/UD1_01_btnCargarDatos"
        android:layout_centerHorizontal="true"
        android:text="LANZAR ACTIVITY" />

    <EditText
        android:id="@+id/UD1_01_txtDatoGardarGlobal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/UD1_01_txtDatoGardarLocal"
        android:layout_below="@+id/UD1_01_txtDatoGardarLocal"
        android:layout_marginTop="78dp"
        android:ems="10"
        android:hint="Dato a gardar globalmente"
        android:textSize="15sp" />

</RelativeLayout>
```

## Código da classe UD1\_01\_DatosPersistentes\_Preferencias

Objetivo: Guardar datos de acceso local e global utilizando SharedPreferences.

```
package com.example.carlos.ud1_01_datospersistentes_preferencias;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class UD1_01_DatosPersistentes_Preferencias extends Activity {

    public static final String NOME_ARQUIVO_PREFERENCIAS="datosguardar";
    public static final String NOME_PARAM_DATO="dato";

    private SharedPreferences datosPrivados;
    private SharedPreferences datosGlobales;

    private void xestionarEventos(){

        Button btnCargarDatos = (Button)findViewById(R.id.UD1_01_btnCargarDatos);
        btnCargarDatos.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub

                EditText textoLocal = (EditText)findViewById(R.id.UD1_01_txtDatoGardarLocal);
                textoLocal.setText(datosPrivados.getString(NOME_PARAM_DATO, "Non hai valor gardado localmente"));

                EditText textoGlobal = (EditText)findViewById(R.id.UD1_01_txtDatoGardarGlobal);
                textoGlobal.setText(datosGlobales.getString(NOME_PARAM_DATO, "Non hai valor gardado globalmente"));

            }
        });

        Button btnGardar = (Button)findViewById(R.id.UD1_01_btnGardarDatos);
        btnGardar.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                EditText textoLocal = (EditText)findViewById(R.id.UD1_01_txtDatoGardarLocal);

                SharedPreferences.Editor editor = datosPrivados.edit();
                editor.putString(NOME_PARAM_DATO, textoLocal.getText().toString());
                editor.commit();
                textoLocal.setText("");

                EditText textoGlobal = (EditText)findViewById(R.id.UD1_01_txtDatoGardarGlobal);

                SharedPreferences.Editor editorGlobal = datosGlobales.edit();
                editorGlobal.putString(NOME_PARAM_DATO, textoGlobal.getText().toString());
                editorGlobal.commit();
                textoGlobal.setText("");

                Toast.makeText(getApplicationContext(), "DATOS GARDADOS", Toast.LENGTH_SHORT).show();
            }

        });

        Button btnLanzarActivity = (Button)findViewById(R.id.UD1_01_btnLanzarActivitySegunda);
        btnLanzarActivity.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
```

```

Intent intent = new Intent(getApplicationContext(), UD1_01_Preferencias_SegundaActivity.class);
startActivity(intent);
}

});

}

@Override
protected void onResume() {
    super.onResume();

    EditText textoLocal = (EditText) findViewById(R.id.UD1_01_txtDatoGardarLocal);
    textoLocal.setText(datosPrivados.getString(NOME_PARAM_DATO, "Non hai valor gardado localmente"));

    EditText textoGlobal = (EditText) findViewById(R.id.UD1_01_txtDatoGardarGlobal);
    textoGlobal.setText(datosGlobales.getString(NOME_PARAM_DATO, "Non hai valor gardado globalmente"));

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ud1_01_datos_persistentes__preferencias);

    datosGlobales = getSharedPreferences(NOME_ARQUIVO_PREFERENCIAS, MODE_PRIVATE);
    datosPrivados = getPreferences(MODE_PRIVATE);

    xestionarEventos();

}
}

```

- 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.

## 1.4.2 Creamos a Activity Secundaria

- Nome da activity: **UD1\_01\_Preferencias\_SegundaActivity**

### Código do layout do xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.ud1_01_datos_persistentes__preferencias.${activityClass}" >

```

```

<EditText
    android:id="@+id/UD1_01_SegundaActivity_txtDato"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:hint="Texto das preferencias"
    android:inputType="text"
    android:textSize="20sp"
    >

    <requestFocus />
</EditText>

<Button
    android:id="@+id/UD1_01_SegundaActivity_btnActualizar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:layout_marginTop="42dp"
    android:text="ACTUALIZAR" />

</RelativeLayout>

```

### Código da clase UD1\_01\_Preferencias\_SegundaActivity

**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).

```

package com.example.carlos.ud1_01_datospersistentes_preferencias;
import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class UD1_01_Preferencias_SegundaActivity extends Activity {
    private SharedPreferences datosGlobales;

    private void cargarPreferencias() {

        datosGlobales = getSharedPreferences(UD1_01_DatosPersistentes_Preferencias.NOME_ARQUIVO_PREFERENCIAS, MODE_PRIVATE);

        EditText texto = (EditText)findViewById(R.id.UD1_01_SegundaActivity_txtDato);
        String dato = datosGlobales.getString(UD1_01_DatosPersistentes_Preferencias.NOME_PARAM_DATO,"Non hai dato gardado");
        texto.setText(dato);
    }

    private void xestionarEventos(){

        Button btnActualizar = (Button)findViewById(R.id.UD1_01_SegundaActivity_btnActualizar);
        btnActualizar.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub

                EditText texto = (EditText)findViewById(R.id.UD1_01_SegundaActivity_txtDato);

                SharedPreferences.Editor editorGlobal = datosGlobales.edit();
                editorGlobal.putString(UD1_01_DatosPersistentes_Preferencias.NOME_PARAM_DATO, texto.getText().toString());
                editorGlobal.commit();

                finish();
            }
        });
    }
}

```

```

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ud1_01_preferencias_segunda);

    cargarPreferencias();
    xestionarEventos();
}
}

```

- Liña 11: Definimos o SharedPreferences.
- Liña 16: Cargamos o SharedPreferences indicando o nome do arquivo. Dito nome está definido na Activity Principal (definida a continuación)
- Liñas 18-20: 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 34-36: 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.

## 1.5 Pantallas de Preferencias

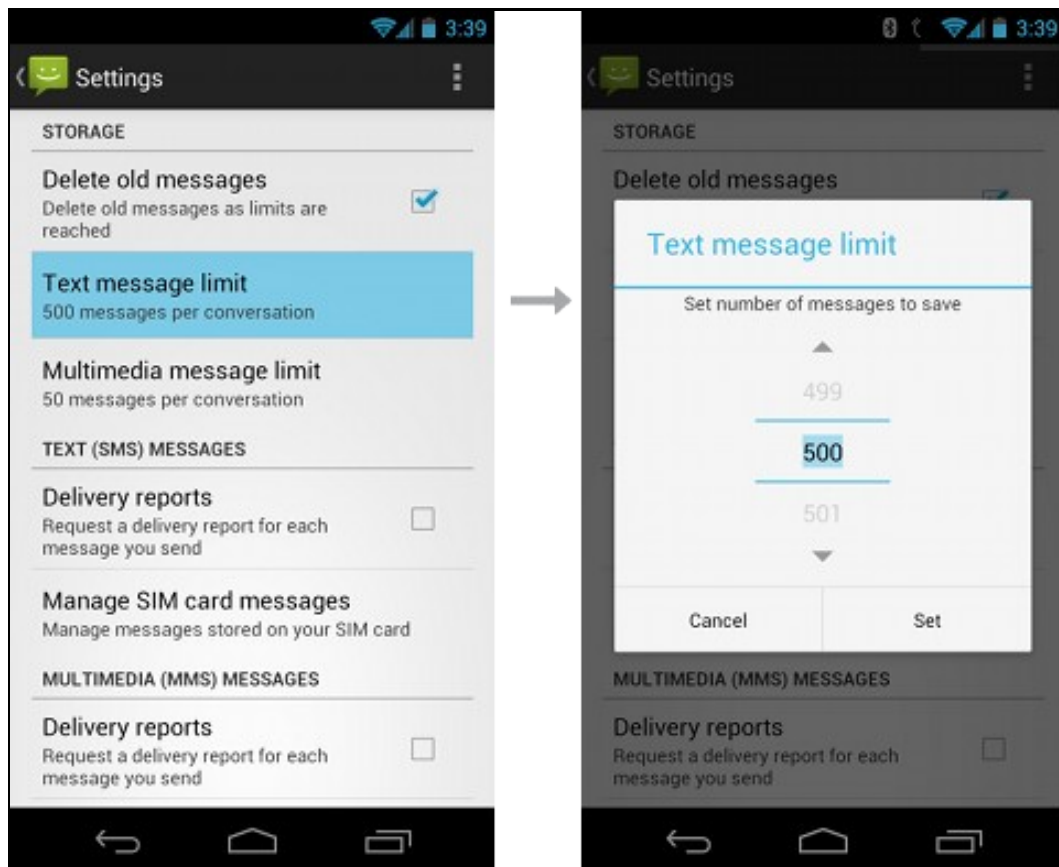
Máis información en:

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

### 1.5.1 Introducción

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).`

Este devolve un obxecto da clase SharedPreferences.

Por exemplo:

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

### 1.5.2 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.

Por exemplo:

```
<?xml version="1.0" encoding="utf-8" ?>
<resources>
<string name="preferencia_numero_por_defecto">10</string>

<string-array name="preferencia_tamanhos_botons_visualizar">
<item>10 sp</item>
<item>12 sp</item>
<item>14 sp</item>
</string-array>
<string-array name="preferencia_tamanhos_botons_valores">
<item>10</item>
<item>12</item>
<item>14</item>
</string-array>
<string name="preferencia_tamanho_por_defecto">12</string>
</resources>
```

Podemos observar como definimos dous arrays, un é o que vai amosar a lista (preferencia\_tamanhos\_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.

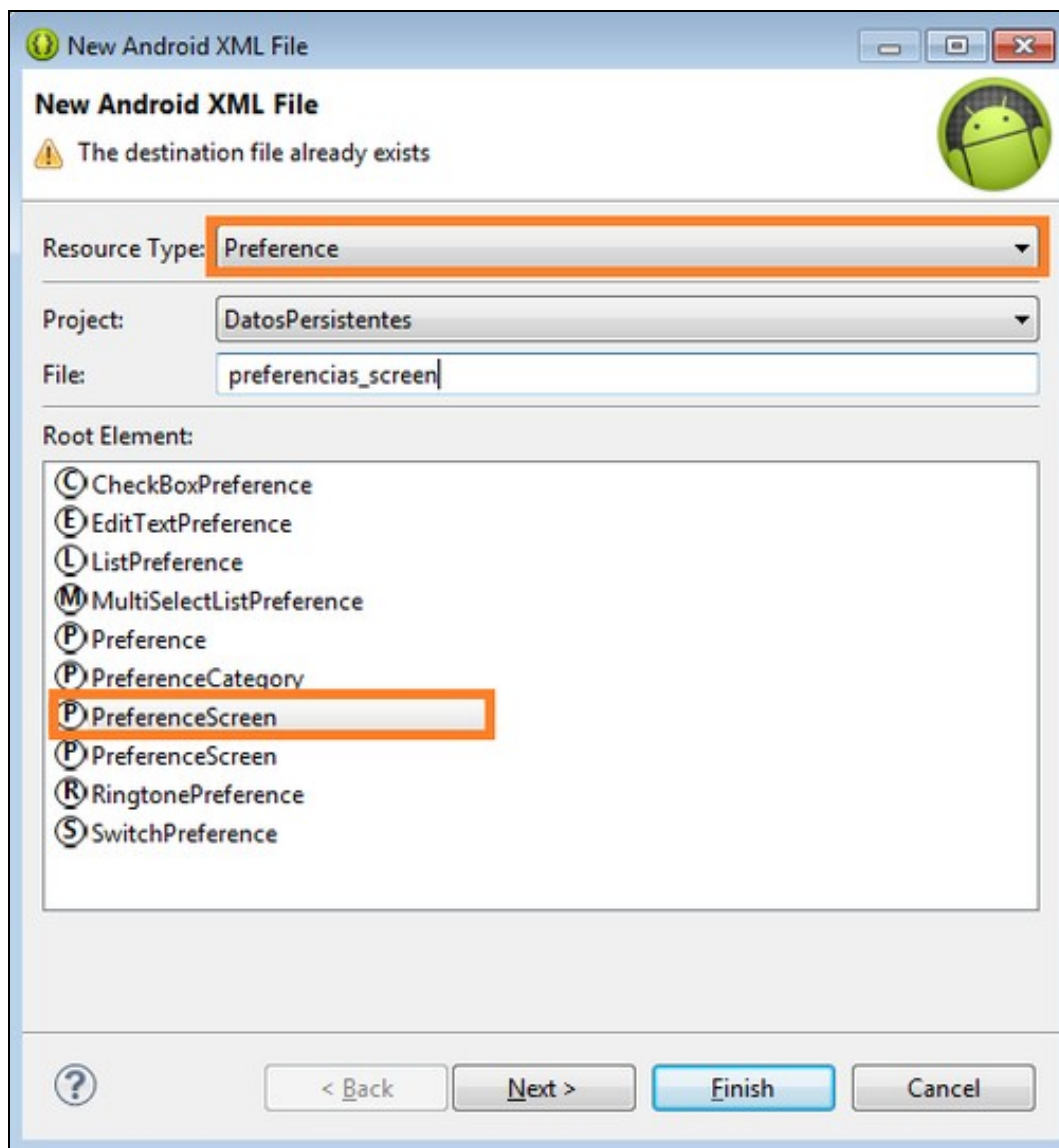
Nota: Isto non é obrigatorio facelo asi pode darse o caso que o que visualice a lista sexa o que necesitamos gardar.

### 1.5.3 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/.

Creamos o cartafol (se non existe) e engadimos un novo arquivo xml de tipo Preference:



Temos que entender a PreferenceScreen como unha activity e o seu layout.

Por exemplo:

```
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
    android:title="EXEMPLO DE SETTINGS" >
    <ListPreference
        android:key="ID_PARAM_TAMAÑO"
        android:title="Preferencias de tamaño"
        android:summary="Garda o tamaño das caixas de texto"
        android:defaultValue="@string/preferencia_tamanho_por_defecto"
        android:dialogTitle="Indicar tamaño"
        android:entries="@array/preferencia_tamanhos_botons_visualizar"
        android:entryValues="@array/preferencia_tamanhos_botons_valores" />
</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.
- **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.

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

- **CheckBoxPreference**: amosa un checkbox. Garda un valor true/false.

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

- **ListPreference**: unha lista de elementos. Xa visto no exemplo.
- **EditTextPreference**: Amosa un diálogo cun EditText. Garda un String.

```
<EditTextPreference
android:key="opcion2"
android:title="Preferencia 2"
android:defaultValue="Texto por defecto?"
android:summary="Descripción da preferencia 2"
android:dialogTitle="Introduce un valor" />
```

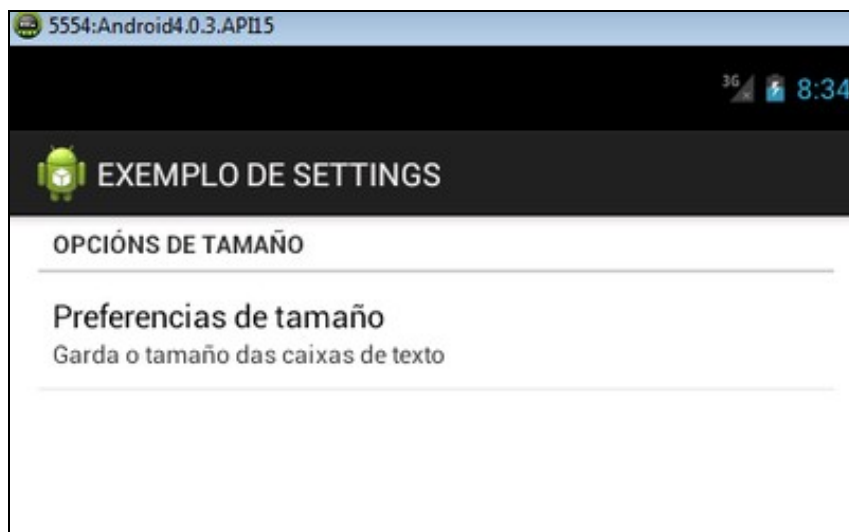
### 1.5.4 Definindo categorías dentro das Pantallas de Preferencias

Podemos agrupar as preferencias por categorías.

Por exemplo o seguinte arquivo de PreferenceScreen:

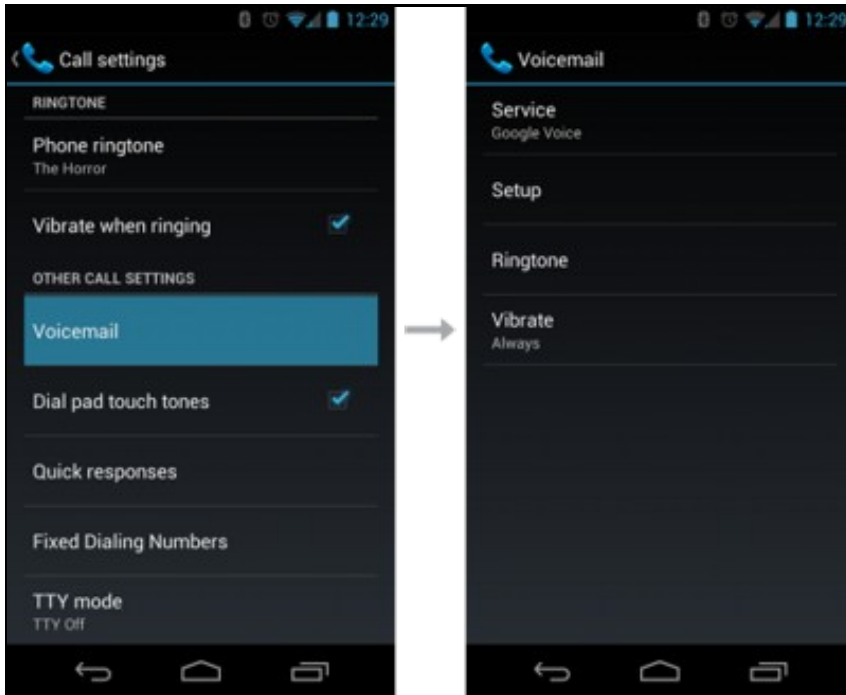
```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" android:title="EXEMPLO DE SETTINGS">
    <PreferenceCategory android:title="OPCIONES DE TAMAÑO">
        <ListPreference
            android:key="@string/preferencia_tamanho_nome_parametro"
            android:title="Preferencias de tamaño"
            android:summary="Garda o tamaño das caixas de texto"
            android:dialogTitle="Indicar tamaño"
            android:defaultValue="@string/preferencia_tamanho_por_defecto"
            android:entries="@array/preferencia_tamanhos_botons"
            android:entryValues="@array/preferencia_tamanhos_botons_valores" />
        </PreferenceCategory>
    </PreferenceScreen>
```

Da como resultado:



## 1.5.5 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:

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
    android:title="EXEMPLO DE SETTINGS" >
    <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
        android:title="PREFERENCIAS DE TAMAÑO">
        <ListPreference

            android:key="@string/preferencia_tamanho_nome_parametro"
            android:title="Preferencias de tamaño"
            android:summary="Garda o tamaño das caixas de texto"
            android:dialogTitle="Indicar tamaño"
            android:defaultValue="@string/preferencia_tamanho_por_defecto"
            android:entries="@array/preferencia_tamanhos_botons"
            android:entryValues="@array/preferencia_tamanhos_botons_valores" />
        </PreferenceScreen>
    </PreferenceScreen>
```

Dará como resultado:



### 1.5.6 Caso Práctico

- Nome do proxecto e da activity principal: **UD1\_02\_DatosPersistentes\_Preferencias**

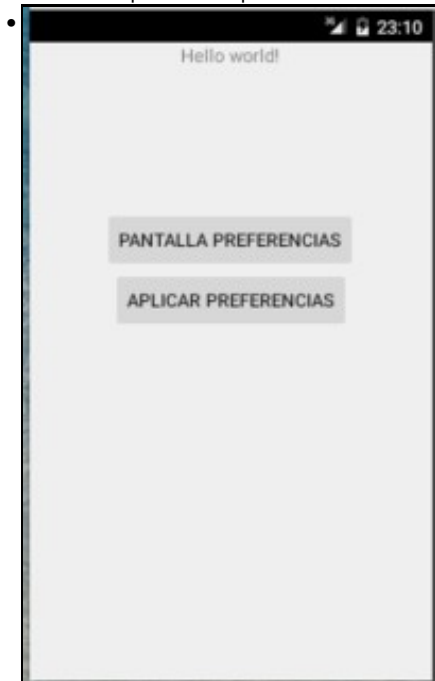
O obxectivo desta práctica é crear unha pantalla de preferencias e acceder aos valores escollidos nesa pantalla dende unha activity.

Consta de dúas activities:

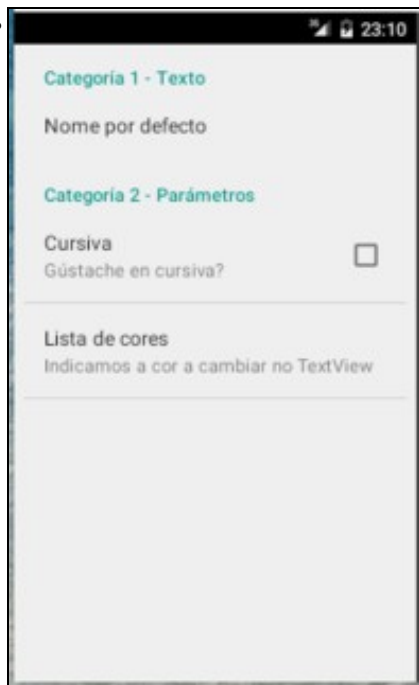
- Activity principal: Dende onde lanzamos a pantalla de preferencias e dende onde accedemos aos valores escollidos.
- Activity secundaria: Carga a pantalla de preferencias.

O aspecto desta práctica é o seguinte:

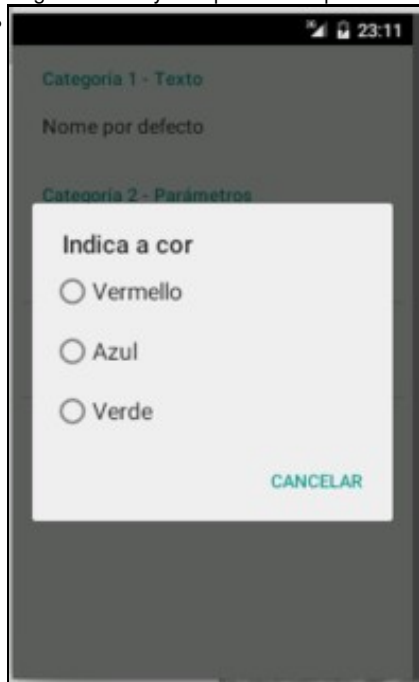
- Uso dunha pantalla de preferencias



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.

#### 1.5.6.1 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 `ud1_02_datospreferencias.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="ud1_02_preferencia_valor_nome_por_defecto">Nome Exemplo</string>
    <string-array name="ud1_02_preferencia_textos_lista">
        <item>Vermello</item>
        <item>Azul</item>
        <item>Verde</item>
    </string-array>
    <string-array name="ud1_02_preferencia_valores_lista">
        <item>#ff0000</item>
        <item>#0000ff</item>
        <item>#00ff00</item>
    </string-array>
</resources>
```

#### 1.5.6.2 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 `ud1_02_preferencias.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
    android:title="EXEMPLO DE SETTINGS">

    <PreferenceCategory
        android:key="Categoria"
        android:title="Categoría 1 - Texto">
        <EditTextPreference
            android:defaultValue="@string/ud1_02_preferencia_valor_nome_por_defecto"
```

```

        android:key="ud1_02_preferencia_idNome"
        android:title="Nome por defecto" />
</PreferenceCategory>

<PreferenceCategory android:title="Categoría 2 - Parámetros">
    <CheckBoxPreference
        android:defaultValue="False"
        android:key="CURSIVA"
        android:summary="Gústache en cursiva?"
        android:title="Cursiva" />

    <ListPreference
        android:dialogTitle="Indica a cor"
        android:entries="@array/ud1_02_preferencia_textos_lista"
        android:entryValues="@array/ud1_02_preferencia_valores_lista"
        android:key="ud1_02_preferencia_listaCores"
        android:summary="Indicamos a cor a cambiar no TextView"
        android:title="Lista de cores" />
</PreferenceCategory>

</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 refencialos da forma android:key="@string/....

### 1.5.6.3 Activity Principal

#### Código XML do layout

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="${relativePackage}.${activityClass}" >

    <TextView
        android:id="@+id/UD1_02_txtTexto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/hello_world" />

    <Button
        android:id="@+id/UD1_02_btnPantallaPreferencias"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/UD1_02_txtTexto"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="113dp"
        android:text="Pantalla Preferencias" />

    <Button
        android:id="@+id/UD1_02_btnAplicarPreferencias"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/UD1_02_btnPantallaPreferencias"
        android:layout_centerHorizontal="true"
        android:text="Aplicar Preferencias" />

</RelativeLayout>

```

## O Código Java

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

```

package com.example.carlos.ud1_02_datospersistentes_preferencias;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;

```



```

import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class UD1_02_DatosPersistentes_Preferencias extends Activity {

    private void xestionarEventos() {

        Button btnPantallaPreferencias = (Button) findViewById(R.id.UD1_02_btnPantallaPreferencias);
        btnPantallaPreferencias.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent intent = new Intent(getApplicationContext(), UD1_02_DatosPersistentes_PantallaPreferencias.class);
                startActivity(intent);
            }
        });
        Button btnAplicarPreferencias = (Button) findViewById(R.id.UD1_02_btnAplicarPreferencias);
        btnAplicarPreferencias.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                aplicarPreferencias();
            }
        });
    }

    private void aplicarPreferencias() {
        SharedPreferences preferencias = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());

        String valorNome = preferencias.getString("ud1_02_preferencia_idNome", "SEN VALOR NO NOME");
        TextView nome = (TextView) findViewById(R.id.UD1_02_txtTexto);
        nome.setText(valorNome);

        String corDefecto = preferencias.getString("ud1_02_preferencia_listaCores", "#000000");
        Toast.makeText(getApplicationContext(), corDefecto, Toast.LENGTH_LONG).show();
        nome.setTextColor(Color.parseColor(corDefecto));

        if (preferencias.getBoolean("CURSIVA", false))
            nome.setTypeface(null, Typeface.ITALIC);
        else
            nome.setTypeface(null, Typeface.NORMAL);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ud1_02__datos_persistentes__preferencias);

        xestionarEventos();
    }
}

```

- 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.

#### 1.5.6.4 Creamos a Activity Secundaria

- Nome da activity: **UD1\_02\_DatosPersistentes\_PantallaPreferencias**

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

#### Código Java

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

```
package com.example.carlos.ud1_02_datospersistentes_preferencias;
import android.app.Activity;
import android.os.Bundle;
import android.preference.PreferenceFragment;

public class UD1_02_DatosPersistentes_PantallaPreferencias extends Activity {

    public static class SettingsFragment extends PreferenceFragment {
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);

            // Load the preferences from an XML resource
            addPreferencesFromResource(R.xml.ud1_02_preferencias);
        }
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);

        getFragmentManager().beginTransaction().replace(android.R.id.content,
            new SettingsFragment()).commit();
    }
}
```

- 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.
- Co comando adb ou co DDMS pódese ir ao directorio shared\_prefs da aplicación e ver o contido do ficheiro xml de preferencias.