

Internacionalización

Sumario

Introdución

Caso práctico

- strings.xml de /res/values
- strings.xml de /res/values-en
- O XML do layout
- O código Java

Consideracións a ter en conta

Introdución

- Tamén temos outras formas de *personalizar* as nosas aplicacións en base a unha serie de sufixos.
- Se queremos que as cadeas de constantes estean traducidas a varios idiomas, só temos que crear un cartafol novo de nome **values-??** onde ?? é o idioma.
- Para o inglés teríamos **/res/values-en**, **/res/values-es** para o español,....
- O idioma por defecto será o que estea gardado en **/res/values**.
- So se deben cambiar os valores das contantes, os identificadores deben ser os mesmos.

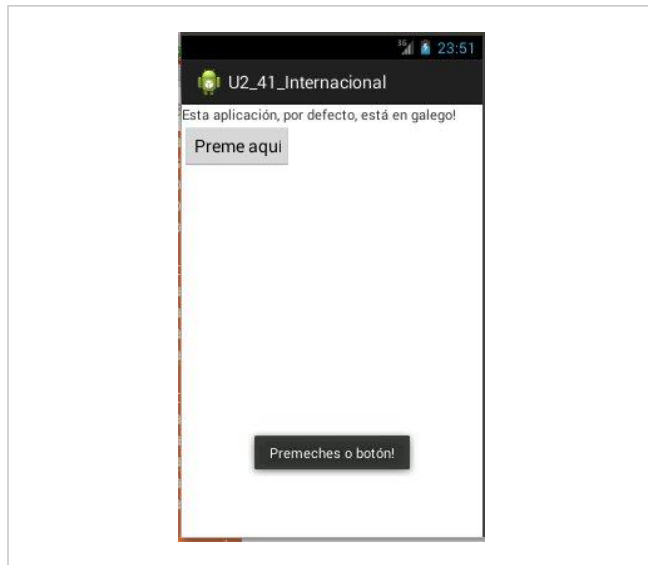
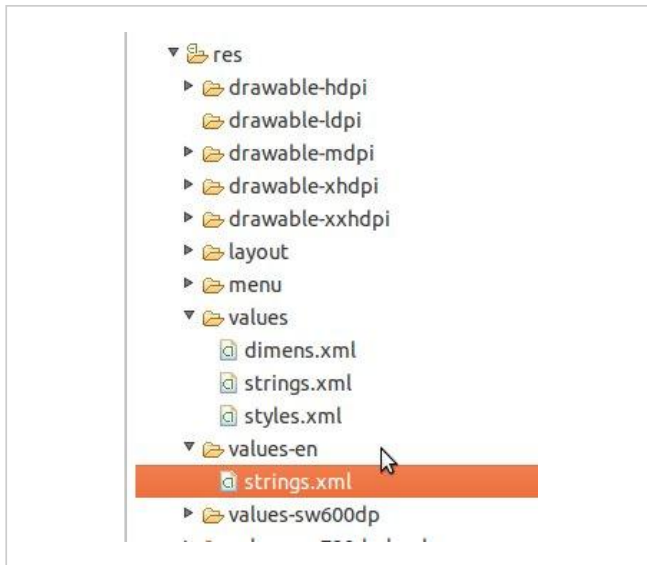
- **Referencias:**

- Localización: <http://developer.android.com/guide/topics/resources/localization.html>

Caso práctico

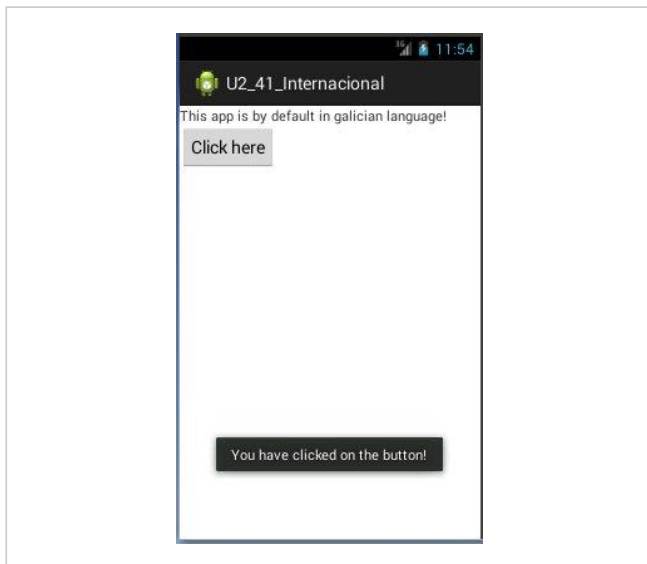
- Crear un novo proxecto: **U2_41_Internacional**
- Imos en **/res/values** definimos as cadeas de texto en galego e esas mesmas cadeas as definimos en inglés en /res/values-en.
- Para probar a aplicación hai que cambiar o idioma do sistema como se viu na Unidade 1.

TableLayout Dinámico



Cartafois de recursos: values e values-en. cada un co seu ficheiro de strings.

A aplicación está en galego, tanto no layout como no código (Toast).



E o mesmo pasa en inglés. O layout e o código Java é o mesmo para os dous casos.

strings.xml de /res/values

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">U2_41_Internacional</string>
5     <string name="action_settings">Settings</string>
6     <string name="idioma">Esta aplicación, por defecto, está en galego!</string>
7     <string name="boton">Preme aquí</string>
8     <string name="mensaxe_toast">Premeches o botón!</string>
9
10 </resources>

```

strings.xml de /res/values-en

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">U2_41_Internacional</string>
5     <string name="action_settings">Settings</string>
6     <string name="idioma">This app is by default in galician language!</string>
7     <string name="boton">Click here</string>
8     <string name="mensaxe_toast">You have clicked on the button!</string>
9
10 </resources>

```

O XML do layout

```

1 <LinearLayout 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     android:orientation="vertical" >
6
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:text="@string/idioma" />
11
12    <Button
13        android:layout_width="wrap_content"
14        android:layout_height="wrap_content"
15        android:text="@string/boton"
16        android:onClick="onBotonClick" />
17
18
19 </LinearLayout>

```

- Liñas 10 e 15: Agora temos que traballar con constantes, non podemos poñer o texto a *lume*.

O código Java

```

1 package com.example.u2_41_internacional;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.View;
7 import android.widget.Toast;
8
9 public class U2_41_Internacional extends Activity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_u2_41_internacional);
15     }
16
17     @Override
18     public boolean onCreateOptionsMenu(Menu menu) {
19         // Inflate the menu; this adds items to the action bar if it is present.
20         getMenuInflater().inflate(R.menu.u2_41_internacional, menu);
21         return true;
22     }
23
24     public void onBotonClick(View v) {
25         Toast.makeText(this, R.string.mensaxe_toast, Toast.LENGTH_SHORT).show();
26     }
27 }

```

- Liña 25 observar como o toast tamén colle a cadea de texto dos recursos tipo string.

Consideracións a ter en conta

Estes sufixos que estamos a utilizar (-land; -es) poden ser combinados entre eles e con outros moitos diferentes.

No seguinte [enlace](http://developer.android.com/guide/topics/resources/providing-resources.html#table2) (<http://developer.android.com/guide/topics/resources/providing-resources.html#table2>) aparece dita orde de prefixos.

Por exemplo:

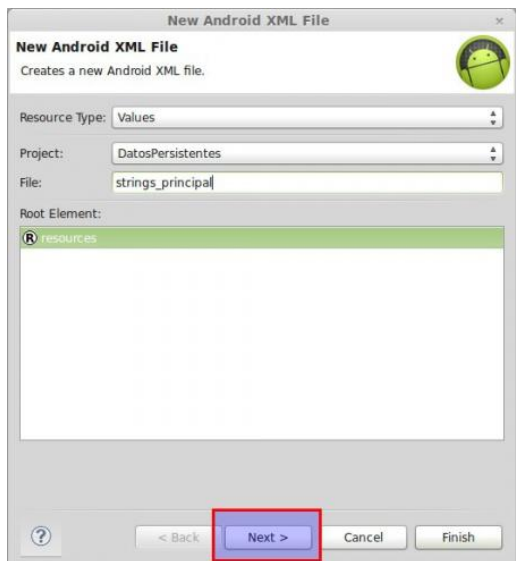
- res/values-en-rUS/strings.xml
- res/values-en-rUK/strings.xml

Onde o segundo sufixo fai referencia a inglés 'americano' (en-rUS) ou 'inglés' do Reino Unido (en-rUK).

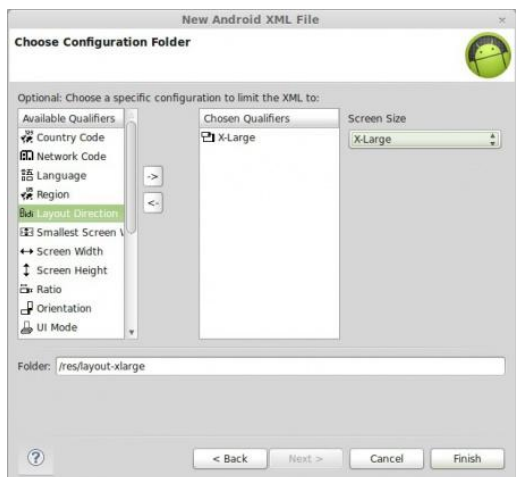
Nota: a letra r fai referencia á rexión.

Para facilitarnos o traballo, Eclipse dispón dunha pantalla na que podemos indicar que 'características' queremos que teña o noso arquivo de recursos e automaticamente xa engade por nós os sufixos necesarios.

Cando creamos un arquivo de recursos aparece a seguinte pantalla:



Se prememos o botón 'Next' pasamos á pantalla comentada anteriormente. Neste exemplo estamos a crear un recurso de tipo layout para pantallas de 7-10 pulgadas (tamaño x-large), entón teríamos que crear un layout nun cartafol 'layout-xlarge':



- Debemos pasar a 'propiedade' SIZE dende a lista da esquerda á dereita (premendo o botón coa dirección dereita). Unha vez pasada debemos darlle un valor, no noso caso x-large.
- Ó facelo, podedes observar como na parte baixa aparece o nome co sufixo correcto.

Podemos combinar moitas máis 'propiedades'. Así, no seguinte exemplo, estamos a crear un arquivo de recursos de tipo 'Values' que se cargará cando utilizemos un dispositivo Android, co idioma Español, nunha pantalla de tipo Small e cunha densidade Media.

- O sufixo que crea neste caso será: -es-small-mdpi.



- Na versión **Lollipop** (API 21, android 5.0) xa se inclúe o idioma galego.

-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).

Obtenido de «<https://manuais.iessanclemente.net/index.php?title=Internacionalización&oldid=57136>»

Esta página se editó por última vez el 28 jul 2015 a las 09:12.

El contenido está disponible bajo la licencia [Creative Commons: CC-BY-NC-SA](#), a menos que se indique lo contrario.