1 Internacionalización

1.1 Sumario

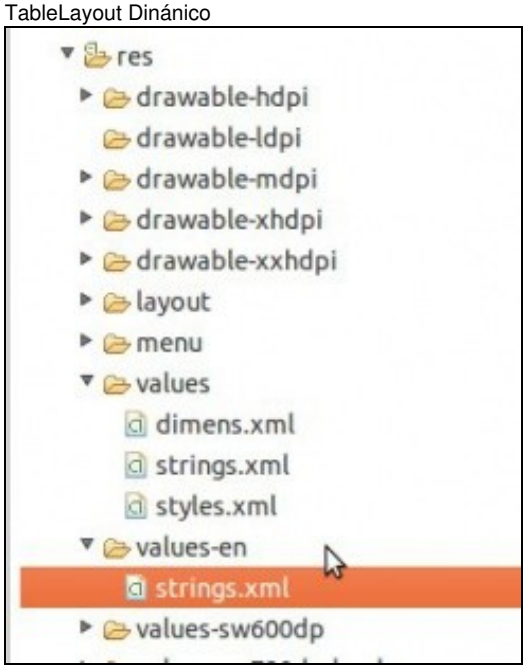
1.2 Introdución

- 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 teriamos /**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 contanstes, os identificadores deben ser os mesmos.

- **Referencias**:
    - Localización: http://developer.android.com/guide/topics/resources/localization.html

1.3 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 Dinánico
- 

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.

1.3.1 strings.xml de /res/values

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">U2_41_Internacional</string>
    <string name="action_settings">Settings</string>
    <string name="idioma">Esta aplicación, por defecto, está en galego!</string>
<string name="boton">Preme aquí</string>
    <string name="mensaxe_toast">Premeches o botón!</string>

</resources>
```

1.3.2 strings.xml de /res/values-en

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
```

```
    <string name="app_name">U2_41_Internacional</string>
    <string name="action_settings">Settings</string>
    <string name="idioma">This app is by default in galician language!</string>
    <string name="boton">Click here</string>
    <string name="mensaxe_toast">You have clicked on the button!</string>

</resources>
```

1.3.3 O XML do layout

```
<LinearLayout 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"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/idioma" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/boton"
        android:onClick="onBotonClick" />


</LinearLayout>
```

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

1.3.4 O código Java

```
package com.example.u2_41_internacional;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.Toast;

public class U2_41_Internacional extends Activity {

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_u2_41__internacional);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.u2_41__internacional, menu);
return true;
}

public void onBotonClick(View v) {
Toast.makeText(this, R.string.mensaxe_toast, Toast.LENGTH_SHORT).show();
}
}
```

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

1.4 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 aparece dita orde de prefixos.

Por exemplo:

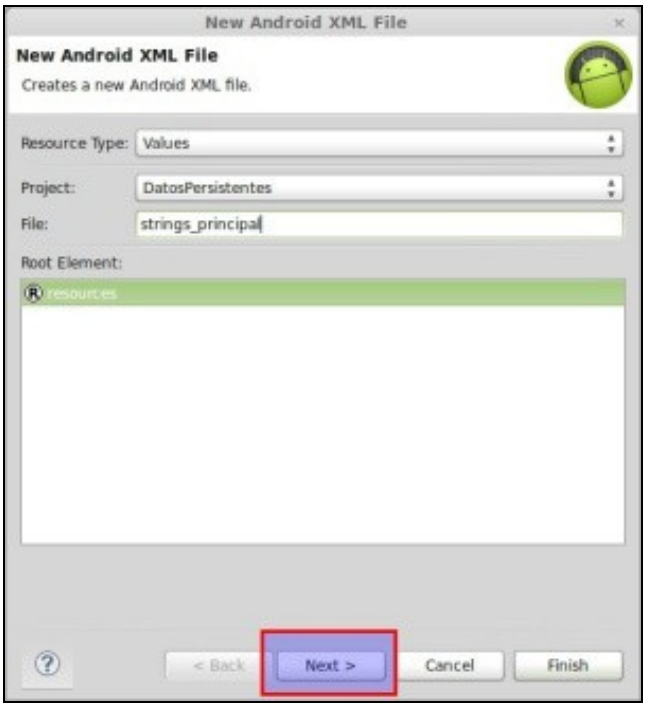
- res/values-en-rUS/strings.xml
- res/values-en-rUK/strings.xml

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:

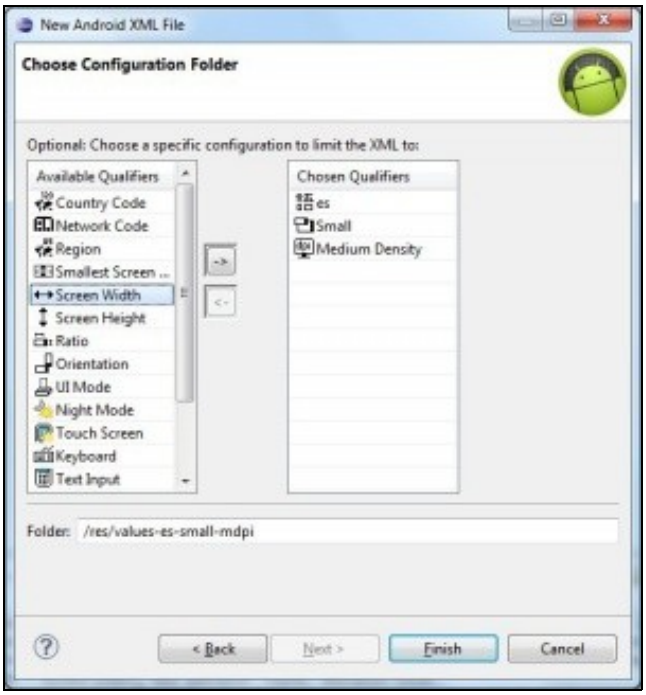


Se prememos o botón 'Next' pasamos á pantalla comentada anteriormente. Neste exemplo estamos a crear un recurso de tipo layout para pantallas de 7-10 pulgadas (tamaño x-large), entón teríamos que crear un layout nun cartafol ?layout-xlarge?:

- 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 utilicemos 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).