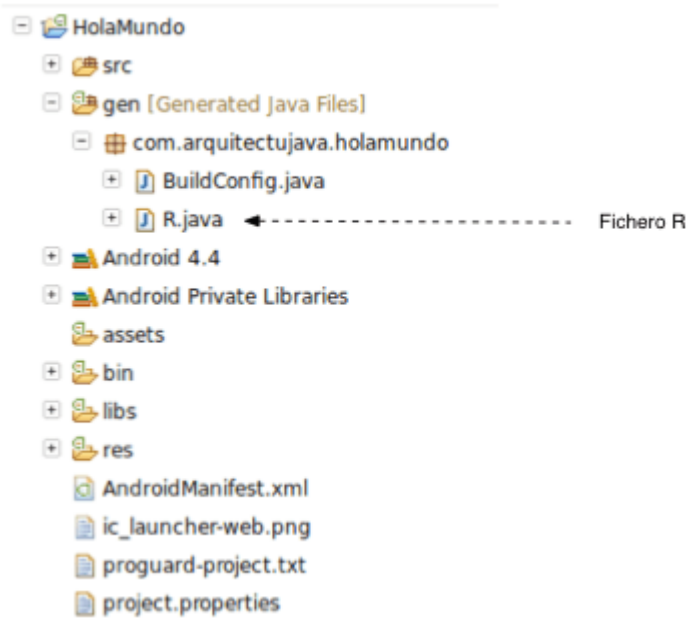
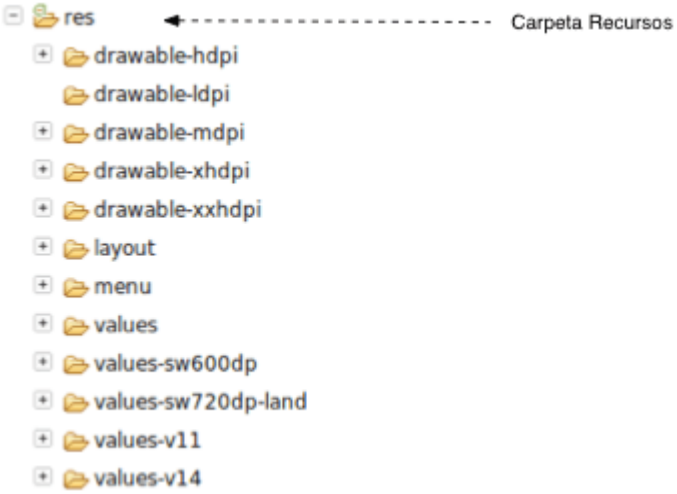
Cuando uno empieza a desarrollar en Android una de las mayores dudas con las que se encuentra es como funciona el fichero R.java que es generado automaticamente por el compilador.

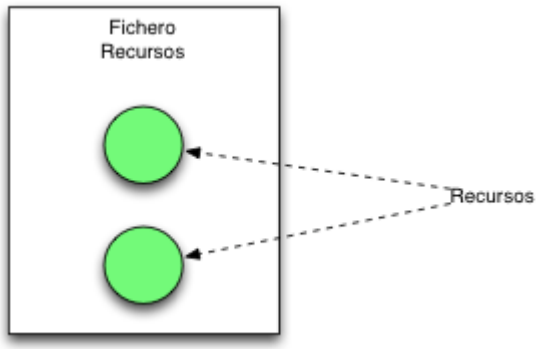


Android y Recursos

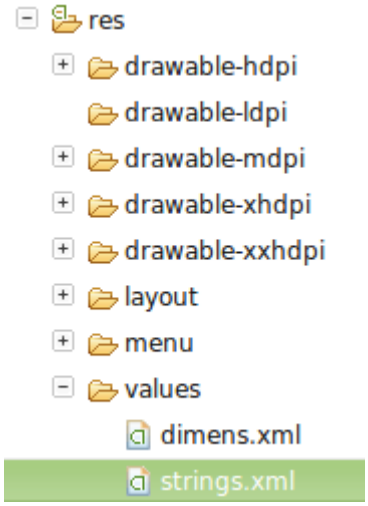
Para entender bien que es el fichero R.java y su contenido debemos hablar primero del concepto de recurso en el mundo de Android y su ubicación en la carpeta res.



Esta carpeta contiene los ficheros de recursos de Android .Ahora bien ¿que es un recurso en Android?. En Android es un concepto amplio y hace referencia a muchos elementos desde imagenes ,textos ,estilos ,xmls que la aplicación necesita a la hora de trabajar . Sin embargo hay que diferenciar entre Fichero de Recursos y Recurso en si .Un fichero de recursos puede incluir varios recursos dentro de él.



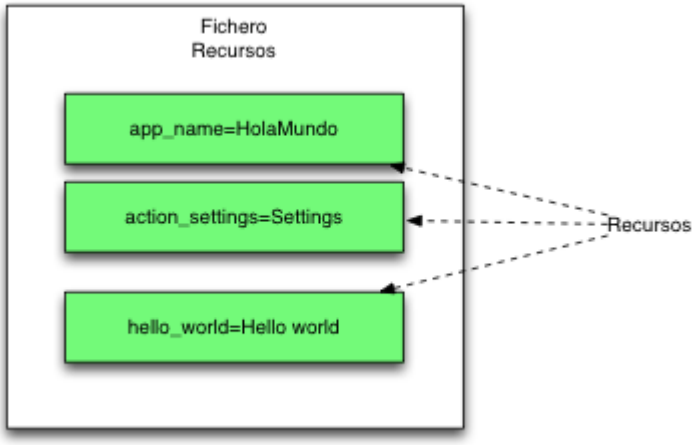
Todos los ficheros de recursos se encuentran ubicados en la carperta /res agrupados por categorias .Vamos a ver el fichero de recursos mas sencillo que se encuentra en la carpeta res/values concretamente se trata del de strings.xml encargado de extraer los textos de la aplicación y parametrizarlos .



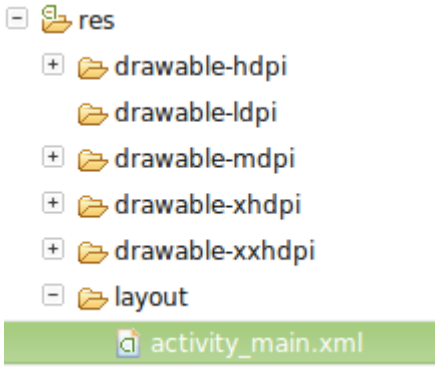
A continuación su contenido:

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

<string name="app_name">HolaMundo</string>
 <string name="action_settings">Settings</string>
 <string name="hello_world">Hello world!</string>

</resources>
```

Como vemos en este caso es un fichero muy sencillo dispone de una etiqueta principal <resources> y dentro de ella se encuentran ubicados los diferentes recursos concretos con su nombre y su valor .



Como acabamos de comentar existen muchos tipos de recursos .No vamos a centrarnos ahora en las categorias de estos sino que vamos a introducir otro de los ficheros de recursos principales el de layout ubicado en res/layout y que se suele denominar activity_main.xml.



Vamos a ver su contenido

```
<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"
```

```
 android:paddingBottom="@dimen/activity_vertical_margin"
 android:paddingLeft="@dimen/activity_horizontal_margin"
 android:paddingRight="@dimen/activity_horizontal_margin"
 android:paddingTop="@dimen/activity_vertical_margin"
 tools:context=".MainActivity" >

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

</RelativeLayout>
```

Realmente no parece algo sencillo de entender ,pero es medianamente normal ya que se trata del fichero que define la estructura de capa de presentación de la aplicación . Por ahora nos basta con entender que el layout contiene una etiqueta (TextView)

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

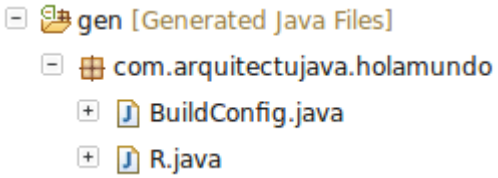
Vamos a modificar a continuación esta etiqueta <TextView> para asociarla un identificador (+id) que se encarga de convertir una etiqueta en un recurso con un nombre determinado (texto1)

```
<TextView android:id="@+id/texto1"
 android:layout_width="wrap_content"
```

```
 android:layout_height="wrap_content"
 android:text="@string/hello_world" />
```

Android R.java

Una vez que tenemos claro que es un fichero de recursos , que dentro de el hay recursos y que podemos añadir nuevos vamos a ver el contenido del fichero R.java que el compilador genera en la siguiente carpeta y que cuando uno empieza con Android genera muchos problemas.

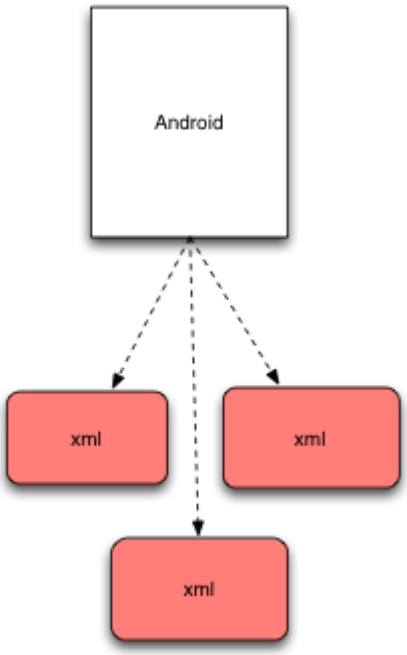


El fichero es pequeño pero su estructura parece chino como se muestra a continuación.

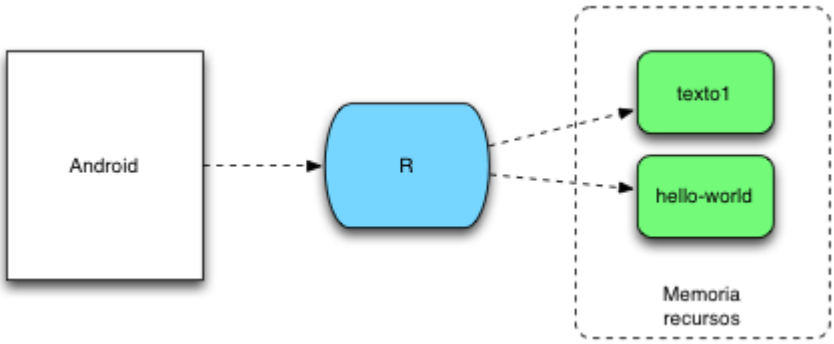
```
package com.arquitecturajava.holamundo;

public final class R {
 public static final class attr {
 }
 public static final class dimen {

 public static final int activity_horizontal_margin=0x7f040000;
 public static final int activity_vertical_margin=0x7f040001;
 }
 public static final class drawable {
 public static final int ic_launcher=0x7f020000;
 }
```

```
 public static final class id {
 public static final int action_settings=0x7f080001;
 public static final int texto1=0x7f080000;
 }
 public static final class layout {
 public static final int activity_main=0x7f030000;
 }
 public static final class menu {
 public static final int main=0x7f070000;
 }
 public static final class string {
 public static final int action_settings=0x7f050001;
 public static final int app_name=0x7f050000;
 public static final int hello_world=0x7f050002;
 public static final int hola_mundo=0x7f050003;
 }
 public static final class style {
 public static final int AppBaseTheme=0x7f060000;

 public static final int AppTheme=0x7f060001;
 }
}
```

Se trata de una clase llena de variables estáticas en las que se identifica cada tipo de recurso . Como podemos ver la cadena de hello_world esta en la sección de Strings y el id texto1 en la sección de ids. ¿Pero para que sirve esta clase exactamente?. Al final todo es mas sencillo de lo que parece Android usa ficheros xml para definir varias estructuras (cadenas de texto,layouts, estilos etc).



Todas estas estructuras XML serán leidas cuando la aplicación arranque …. pero como nos encontramos en una plataforma movil el trabajo con XML directo podría ser demoledor y lento . Por lo tanto Android despues de leer el fichero XML carga todas las estructuras que se han solicitado en memoria y mantiene el fichero R como referencia directa a los recursos que se han cargado de esta manera el acceso será directo y sencillo para el programador.



Usando el fichero R

Vamos a ver un sencillo bloque de código en el cual usamos el fichero R para localizar el TextView de la aplicación y cambiar su contenido . Para ello modificaremos el fichero de recursos de strings.xml y añadiremos una nueva cadena “hola_mundo”.

```
<resources>

<string name="app_name">HolaMundo</string>
 <string name="action_settings">Settings</string>
 <string name="hello_world">Hello world!</string>
 <string name="hola_mundo">Hola Mundo</string>

</resources>
```

Hecho esto modificaremos el programa principal para hacer uso del fichero R y cambiar el texto por defecto de la aplicación.

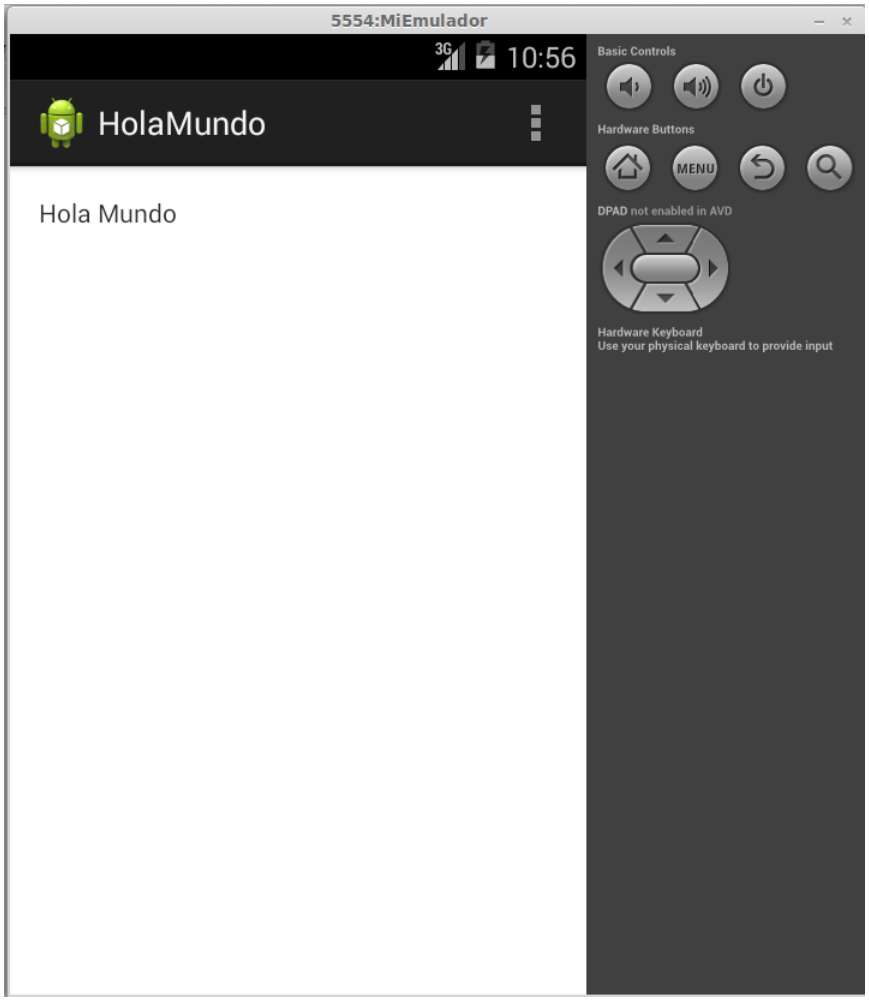
```
package com.arquitecturajava.holamundo;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {

@Override
 protected void onCreate(Bundle savedInstanceState) {
```

```
 super.onCreate(savedInstanceState);

 setContentView(R.layout.activity_main);
 TextView texto1= (TextView) findViewById(R.id.texto1);
 texto1.setText(R.string.hola_mundo);
 }
 @Override
 public boolean onCreateOptionsMenu(Menu menu) {
 // Inflate the menu; this adds items to the action bar if it is
present.
 getMenuInflater().inflate(R.menu.main, menu);
 return true;
 }

}
```

El código fundamental son estas dos lineas

```
TextView texto1= (TextView) findViewById(R.id.texto1);
 texto1.setText(R.string.hola_mundo);
```

El método findViewById nos localiza el TextView usando el fichero R y el bloque de identificadores y luego simplemente usamos el fichero R y su acceso a las cadenas para cambiar el contenido. el programa mostrará lo siguiente.



Como podemos ver el fichero R es necesario conocerlo incluso para operaciones triviales .