Retornar valor desde una Actividad Secundaria en Android

Buenas tardes, en este tutorial vamos a aprender a retornar un valor desde una actividad secundaria a una actividad primaria.

- ## FirstActivity.java

En esta actividad vamos a disponer de una interfaz gráfica muy simple, la cual veremos a continuación:

Una vez tenemos esta GUI, vamos a adentrarnos en el código de nuestra Activity, el cual se encuentra en la dirección src/nombre-del-paquete/FirstActivity.java. Su código es el siguiente:

```
1
2
3   public class FirstActivity extends Activity {
4       int request_code = 1;
5       private TextView tvTexto;
6
7       @Override
8       protected void onCreate(Bundle savedInstanceState) {
9           super.onCreate(savedInstanceState);
10          setContentView(R.layout.activity_first);
11          tvTexto = (TextView)findViewById(R.id.tvTexto);
12      }
13
14      public void onClick(View v){
15          Intent i = new Intent(this, SecondActivity.class);
16          startActivityForResult(i, request_code);
17      }
18
19      @Override
20      protected void onActivityResult(int requestCode, int resultCode, Intent data)
21          // TODO Auto-generated method stub
22          if ((requestCode == request_code) && (resultCode == RESULT_OK)){
23              tvTexto.setText(data.getDataString());
24          }
25      }
26
  }
```

Como podemos ver, tenemos una variable de instancia de tipo int llamada request_code, la cual será usada mas tarde en el método onActivityResult para comprobar lo que ha llegado.

El **TextView** será el encargado de mostrar el texto que seleccionaremos en la segunda Actividad.

En el método onClick, el cual se ejecuta al pulsar en el **Button**, vemos que no usamos la llamada mediante el método startActivity, sino que usamos el método **startActivityForResult()**. Este método inicia una actividad de la cual queremos retornar un valor a la actividad que la ha llamado.

- # SecondActivity.java

En este actividad vamos a hacer algo simple. Será la encargada de cargar un **ListView** con cadenas de texto, y cuando pulsemos sobre un elemento, se cerrará la Actividad y se nos volverá a mostrar la FirstActivity, con el TextView actualizado.

Su GUI es fácil y no requiere mucho trabajo,

Nos debería quedar algo así:

Una vez tenemos su UI, vamos a ver el código de la Actividad, que se encuentra en el mismo lugar que nuestra otra actividad, en la carpeta src/nombre-del-paquete/SecondActivity.java.

```
1  public class SecondActivity extends Activity {
2      private ListView lvString;
3
4      @Override
5      protected void onCreate(Bundle savedInstanceState) {
6          super.onCreate(savedInstanceState);
7          setContentView(R.layout.activity_second);
8
9          lvString = (ListView)findViewById(R.id.lv_Strings);
           ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.l
   cargarListView());
```

```
10          lvString.setAdapter(adapter);
11
12          lvString.setOnItemClickListener(new OnItemClickListener(){
13
14              @Override
15              public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
16              long arg3) {
17                  // TODO Auto-generated method stub
18                  String cad = (String)lvString.getAdapter().getItem(arg2);
19                  Intent data = new Intent();
20                  data.setData(Uri.parse(cad));
21                  setResult(RESULT_OK, data);
22                  finish();
23              }
24
25          });
26
27      }
28
29      private List<String> cargarListView(){
30          List<String> listaStrings = new ArrayList<String>();
31          for (int i = 0 ; i < 10; i++){
32              String cad = "String " + i + " de la Segunda Actividad";
33              listaStrings.add(cad);
34          }
35          return listaStrings;
36      }
37  }
38
39
```

Lo importante en esta actividad es el código que está dentro del onCLickListener del **ListView**. Cuando pulsemos sobre un elemento, se va a crear un Intent, el cual tendrá el String que hemos seleccionado. Una vez esté cargado, vamos a llamar al método **setResult()**, el cual tiene la constante RESULT_OK, es decir, que el resultado que enviamos es correcto. Si por el contrario quisieramos retroceder a la actividad anterior sin enviar nada, podríamos establecer RESULT_CANCELLED. Como segundo argumento, recibe el Intent con los datos que hemos cargado en el. Una vez hecho esto, llamamos al método **finish()**, el cual finalizará la actividad y volverá a la primera actividad.

- onActivityResult()

Este método el cual se encuentra en la primera actividad, es el encargado de gestionar el Intent que hemos recibido de la segunda actividad.

```
1
2   @Override
3   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
4       // TODO Auto-generated method stub
5       if ((requestCode == request_code) && (resultCode == RESULT_OK)){
6           tvTexto.setText(data.getDataString());
7       }
8   }
```

Este método trae consigo los parámetros **requestCode**, **resultCode** y **data**. Los 2 primeros parámetros los usaremos para comprobar, es decir, si el requestCode es igual que el que definimos en la clase de la actividad, y si el resultCode tiene el mismo valor que la constante RESULT_OK, entonces extraemos los datos que trae consigo el Intent.

Como lo que guardamos en nuestro Intent era un objeto de la clase String, usamos el método getDataString() para extraerlo y asignarlo al **TextView**.

Ahora podemos ejecutar nuestra aplicación y ver como queda tras haber hecho todo lo anterior: