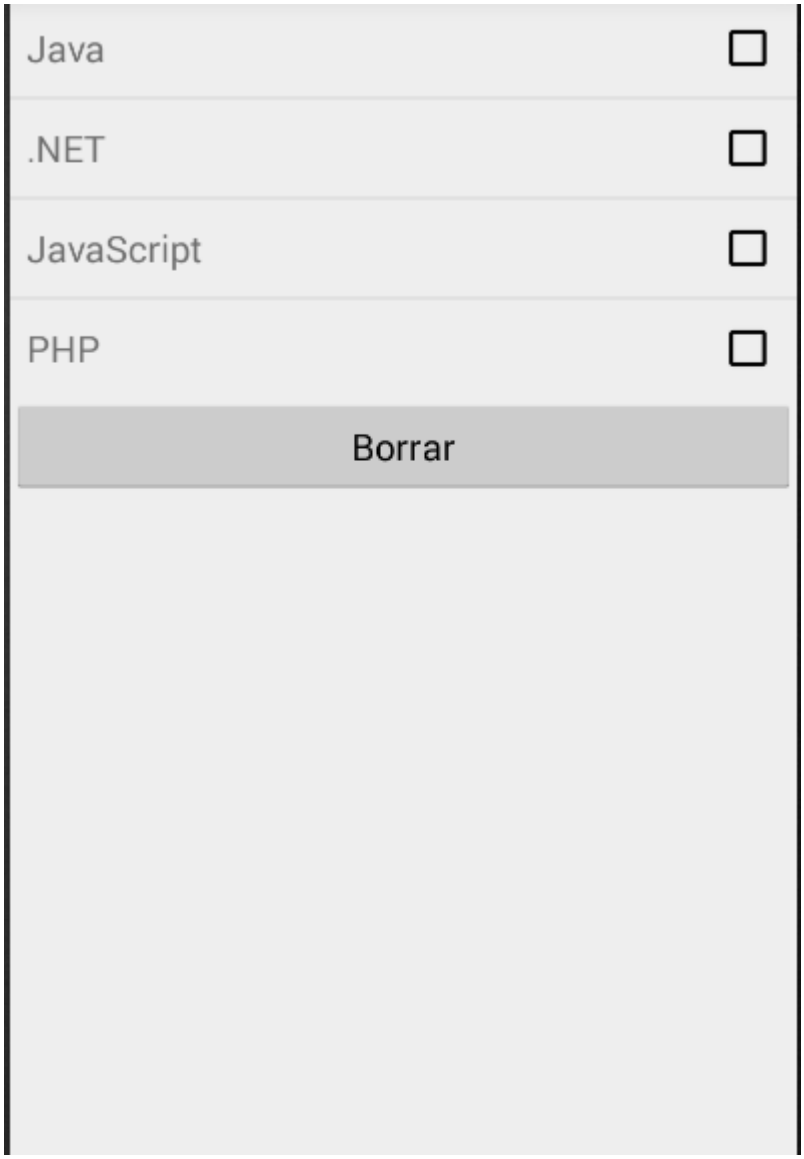
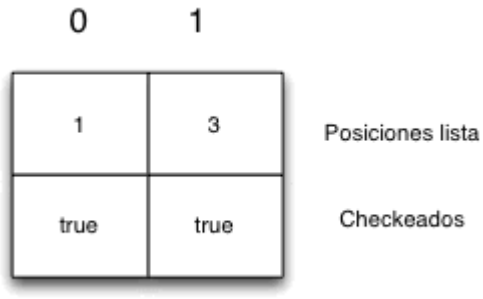
Hoy voy a hablar de Android SparseBooleanArray. Si hay una tecnología Java que a veces es realmente complicada de entender es Android. ¿Para que sirven los SparseBooleanArray?. Pues para algo teoricamente muy sencillo como es seleccionar los elementos seleccionados de un ListView. El problema es que en algunas ocasiones queremos eliminar los elementos seleccionados.



Estructura del SparseBooleanArray

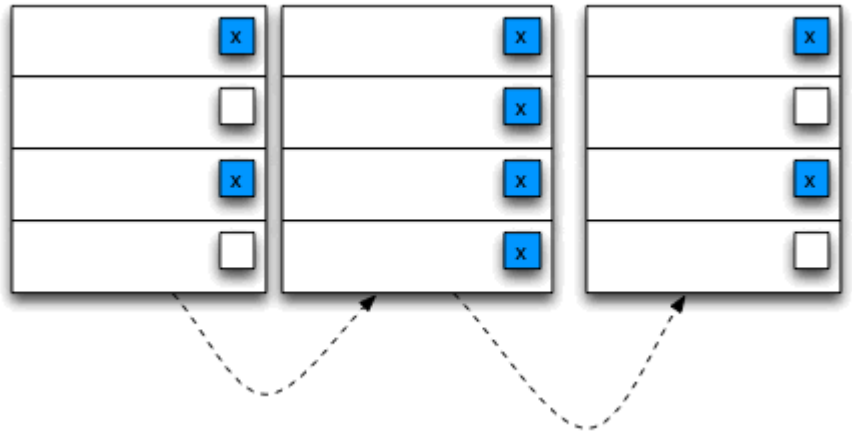
¿Cómo funciona exactamente esta clase? . El SparseBooleanArray es una clase optimizada para Android que almacena los elementos que hemos pulsado de nuestra lista. Por ejemplo si hemos pulsado dos elementos :

El SparseBooleanArray creará la siguiente estructura:

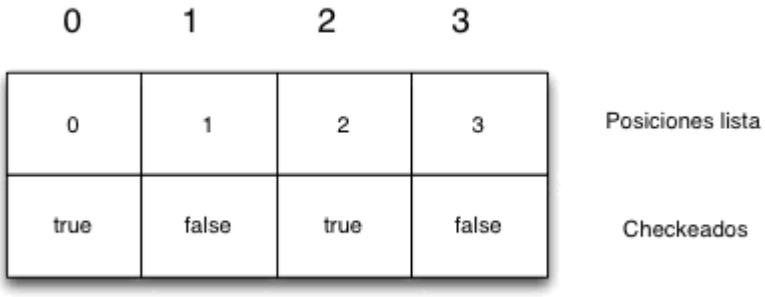


La estructura dispondrá de dos items y cada uno de ellos almacena la posición real en la lista del elemento seleccionado. Ahora bien si seleccionamos todos los elementos y luego

deseleccionamos dos para volver a la posición original.



Sorprendentemente el SparseBooleanArray no tendrá la misma estructura que antes hemos visto sino que tendrá la siguiente:



Esto genera muchos problemas a la hora de trabajar con él. Ademas en nuestro caso si recorremos el Array y eliminamos elementos la longitud del Array varía. He optado por recorrer el SparseArray realizar los chequeos oportunos y crear una nueva lista con los elementos a eliminar.

```
package arquitecturajava.com.android033borrarvarios;
```

```
import android.app.Activity;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.util.SparseBooleanArray;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckedTextView;
import android.widget.EditText;
import android.widget.ListView;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends Activity {

 List<String> listaDatos;
 ListView lista;
 ArrayAdapter<String> adaptador;
 Button miboton;

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

```
 lista= (ListView)findViewById(R.id.lista);

 miboton=(Button)findViewById(R.id.miboton);
 adaptador= new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_multiple_choice,listaDatos);
 lista.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
 lista.setAdapter(adaptador);

 miboton.setOnClickListener(new View.OnClickListener() {
 @Override
 public void onClick(View v) {

 SparseBooleanArray marcados = lista.getCheckedItemPositions();
 List<String> listaElegidos= new ArrayList<String>();

 for (int i = 0; i < marcados.size(); i++) {

 int posicionReal = marcados.keyAt(i);
 boolean esBorrable = marcados.get(posicionReal);
 if(esBorrable==true) {
 listaElegidos.add(adaptador.getItem(posicionReal));
 }
 }
 listaDatos.removeAll(listaElegidos);
 adaptador.notifyDataSetChanged();
 lista.clearChoices();
 }

 });

 }
```

```
 private void iniciarLista() {

 listaDatos= new ArrayList<String>();
 listaDatos.add("Java");
 listaDatos.add(".NET");
 listaDatos.add("JavaScript");
 listaDatos.add("PHP");
 }
}
```

Hecho esto utilizo el ArrayList y su método removeAll para eliminar los elementos que previamente fueron seleccionados. El siguiente paso es actualizar el adaptador y la lista.

Otros artículos relacionados :Android Intents , Android Eventos , Android R