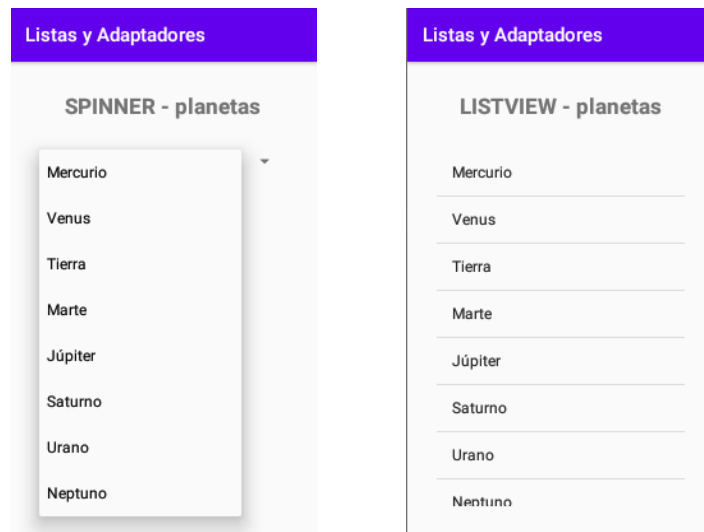


LISTADOS Y ADAPTADORES

1. INTRODUCCION

- Android dispone de diversas vistas que nos permiten seleccionar una opción dentro de una **lista de posibilidades**. Así, podremos utilizar listas desplegables (**Spinner**) o listas fijas (**ListView**), entre otros.



- Como siempre, elaborar y gestionar una lista simple (**ListView** o **Spinner**) es relativamente sencillo. Pero en ocasiones, según las necesidades de nuestra app, necesitamos personalizar el aspecto de cada elemento del listado:



- Para esto, la programación se hace más compleja y requiere el uso de un elemento adicional: el **adaptador** o **Adapter**.
- El adaptador es el elemento que sirve de conexión entre los datos que queremos visualizar y las vistas que los muestran



(Imagen de CABRERA RODRIGUEZ, J: "Programación multimedia y dispositivos móviles". Ed. Síntesis)

- La imagen muestra como un adaptador es un elemento que hace de intermediario entre una fuente de datos (array, fichero, BBDD) y una interfaz de usuario que muestra esos datos, por ejemplo un **Spinner** o una **ListView**.
- Es el **responsable de generar las vistas necesarias para mostrar los datos**. Por ejemplo, cada ítem de una lista puede tener sólo una línea de texto (como en las capturas iniciales) o estar formado por varios subcomponentes (por ejemplo, textos e imagen, en la lista personalizada anterior). En cada caso, es el adaptador el encargado de representar eso.
- Android proporciona por defecto varios tipos de adaptadores sencillos.
- El adaptador más sencillo es el **ArrayAdapter**, que proporciona datos a un Spinner o a una ListView a partir de un array de objetos de cualquier tipo.

2. SPINNER

- Es una lista **desplegable** en la que podemos seleccionar una opción.
- Un spinner se define con un elemento del mismo nombre en el archivo de layout, es decir, con una etiqueta **<Spinner>**:

```
<Spinner  
    android:id="@+id/spPlanetas"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

- Si los datos que queremos mostrar en el spinner son estáticos es posible definir la lista de valores como un recurso de tipo **string-array**. Para ello, primero creamos un nuevo fichero XML en la carpeta **/res/values** llamado, por ejemplo, **arrays.xml** e incluimos en él los valores seleccionables de la siguiente forma:

```

<resources>
    <string-array name="planetas">
        <item>Mercurio</item>
        <item>Venus</item>
        <item>Tierra</item>
        <item>Marte</item>
        <item>Júpiter</item>
        <item>Saturno</item>
        <item>Urano</item>
        <item>Neptuno</item>
    </string-array>
</resources>

```

Hay una etiqueta `<item>` por cada valor que queremos que aparezca en el spinner.

- También es válido añadir el **string-array** como un recurso más en el archivo **strings.xml** pero, si lo hacemos así, quedaría peor estructurado.
- Para facilitar la asociación entre estos datos y la vista de tipo spinner, podemos utilizar la propiedad **android:entries="@array/nombre_del_string-array"**:

```
android:entries="@array/planetas"
```

- Para obtener el **contenido del item seleccionado** y su **posición** en el Spinner se pueden emplear los métodos **getSelectedItem()** y **getSelectedItemId()**, de la clase **AdapterView**. Por ejemplo:

```
String planeta=spPlanetas.getSelectedItem().toString();
```

- La primera posición del spinner tiene el valor 0.
- El más común de los eventos lanzados por la vista Spinner es el que se produce cuando seleccionamos una opción de la lista desplegable: el evento **onItemSelected**
- Para realizar un escuchador o **listener** que responda al evento en cuestión se procederá de forma similar a lo visto para, por ejemplo, el evento **onClick** de un botón. Es decir, emplearíamos el método **setOnItemSelectedListener()**:

```

//escuchador del spinner
spPlanetas.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
        //operaciones pertinentes
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {

    }
});

```

- Al implementar el listener hay que sobrescribir los métodos **onItemSelected()** y **onNothingSelected()**.

- El método ***onItemSelected()*** es llamado cuando se selecciona un item.

Parámetros principales:

- ***adapterView***: El elemento (adaptador) donde se hizo la selección.
 - ***view***: La vista seleccionada dentro del AdapterView.
 - ***i***: La posición de la vista dentro del adaptador.
- El método ***onNothingSelected()*** no tiene utilidad para nosotros.
 - También podemos recuperar el elemento seleccionado utilizando el método ***getItemAtPosition()*** del parámetro ***adapterView*** que recibimos en el evento.

```
@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l)
{
    String eleccion1=spPlanetas.getSelectedItemAtPosition(i).toString();
    String eleccion2=adapterView.getItemAtPosition(i).toString();
    //resto de operaciones
}
```

- Podemos probar todo esto para obtener el ejemplo recogido en la primera imagen capturada.
- Lo que acabamos de ver es la manera más simple de gestionar un Spinner: utilizamos únicamente su definición en /res/layout/main.xml y un fichero de recursos en res/values/arrays.xml, con los elementos a mostrar. Pero, si queremos tener el control del contenido del Spinner e incluso modificar el aspecto con el que se muestra, debemos aprender más sobre los adaptadores.

Documentación en <http://developer.android.com/reference/android/widget/ArrayAdapter.html>

3. ADAPTADORES

3.1 ADAPTADOR CON ARRAY ESTATICO

- Los elementos que forman un spinner también pueden ser definidos **mediante código Java**. El ejemplo equivalente al anterior sería:

```
String[] arrayPlanetas= {"Mercurio", "Venus", "Tierra", "Marte", "Júpiter",
    "Saturno", "Urano", "Neptuno"};
```

- Los pasos siguientes son **crear el adaptador** que reconozca y pueda darles forma a dichos datos, y **asignar el adaptador al spinner**, mediante el método ***setAdapter()***:

```
//creamos el elemento adaptador
ArrayAdapter<String> adaptador = new ArrayAdapter<String>
    (this, android.R.layout.simple_spinner_item, arrayPlanetas);
//asignamos al Spinner el adaptador con los datos ya cargados
spPlanetas.setAdapter(adaptador);
```

- La clase ***ArrayAdapter*** tiene varios constructores:

Public constructors
<code>ArrayAdapter(Context context, int resource)</code> Constructor
<code>ArrayAdapter(Context context, int resource, int textViewResourceId)</code> Constructor
<code>ArrayAdapter(Context context, int resource, T[] objects)</code> Constructor.
<code>ArrayAdapter(Context context, int resource, int textViewResourceId, T[] objects)</code> Constructor.
<code>ArrayAdapter(Context context, int resource, List<T> objects)</code> Constructor
<code>ArrayAdapter(Context context, int resource, int textViewResourceId, List<T> objects)</code> Constructor

- En este ejemplo hemos utilizado el constructor con tres parámetros:
 - Un objeto de tipo Context, que referencia a la propia actividad (***this***).
 - La manera en que será mostrado el spinner. Android proporciona una serie de layouts predefinidos de los cuales los más habituales son:
 - ***simple_spinner_item***
 - ***simple_spinner_dropdown_item***
 - El array de objetos (strings) que queremos visualizar.
- Los layouts predefinidos, como “*simple_spinner_item*”, se pueden visualizar con aspecto diferente según el nivel de API.
- Podemos probar todo esto en un ejemplo similar al anterior en cuanto a funcionalidad, pero con este nuevo código que incluye el uso de un adaptador simple.