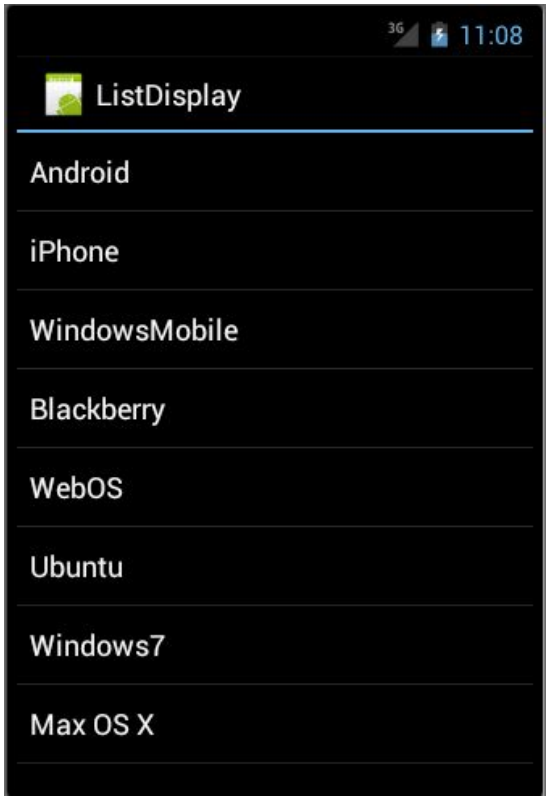
Apartado 9:
ListViews

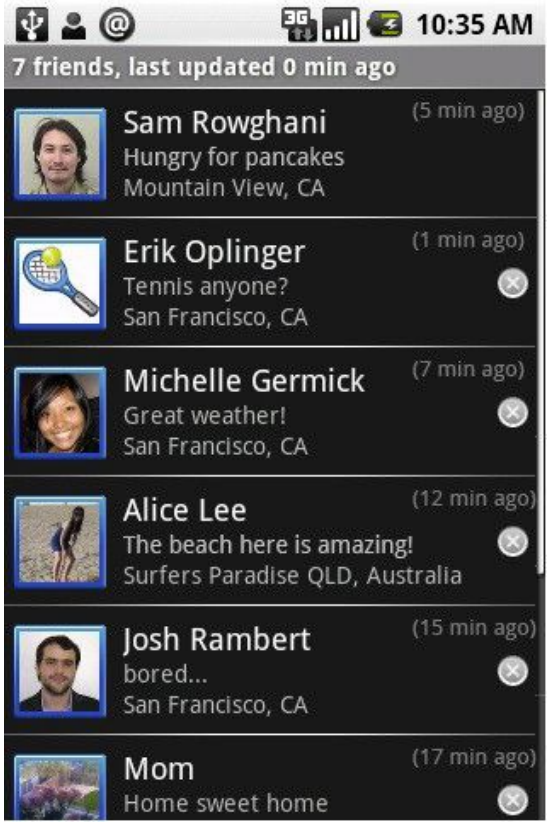
Autores:
Víctor M. Rivas Santos / Miguel Á. García Cumbreras
(Antonio Rueda Ruiz)

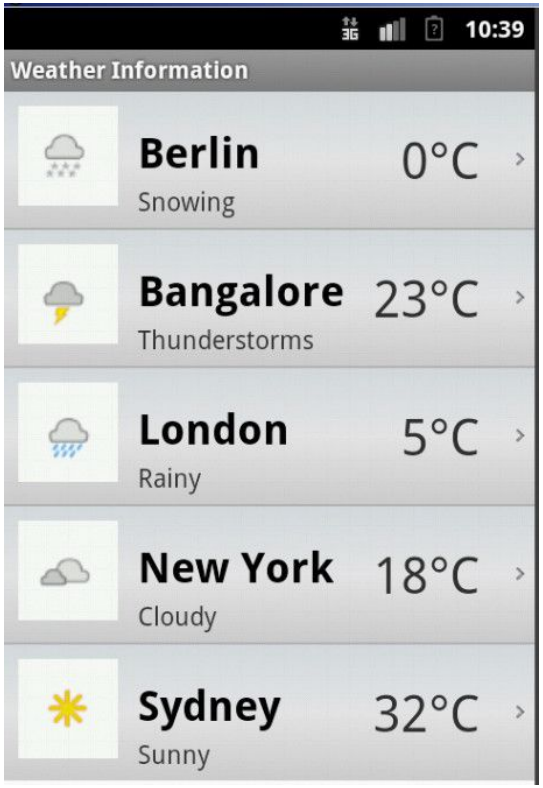
ListView

- Vista que muestra un conjunto de items en una lista vertical
- Muy usados (y también complejos) en la actualidad
- Alto grado de flexibilidad y personalización

ListView


11:08
ListDisplay
Android
iPhone
WindowsMobile
Blackberry
WebOS
Ubuntu
Windows7
Max OS X


10:35 AM
7 friends, last updated 0 min ago
Sam Rowghani
Hungry for pancakes
Mountain View, CA
(5 min ago)
Erik Oplinger
Tennis anyone?
San Francisco, CA
(1 min ago)
Michelle Germick
Great weather!
San Francisco, CA
(7 min ago)
Alice Lee
The beach here is amazing!
Surfers Paradise QLD, Australia
(12 min ago)
Josh Rambert
bored...
San Francisco, CA
(15 min ago)
Mom
Home sweet home
(17 min ago)


10:39
Weather Information
Berlin
Snowing
0°C
Bangalore
Thunderstorms
23°C
London
Rainy
5°C
New York
Cloudy
18°C
Sydney
Sunny
32°C

ListAdapters

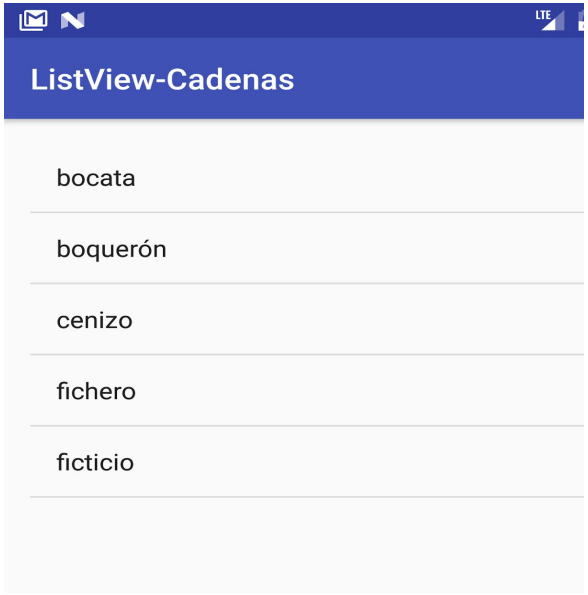
- Para que un *ListView* pueda mostrar una serie de items…
- ...estos deben haber sido generados por un *ListAdapter*
- El *ListAdapter* realiza dos funciones:
  1) Hace de intermediario con el proveedor de datos
  2) Genera un View para cada item
- Existen dos *ListAdapter* predefinidos:
  - *ArrayAdapter* → para mostrar datos obtenidos de un array o List
  - *CursorAdapter/SimpleCursorAdapter* → para mostrar datos obtenidos normalmente de una consulta de base de datos

Layouts para items

- En android.R.layout están predefinidos algunos layouts
  - simple_list_item_1
  - simple_list_item_2
  - simple_list_item_activated_1
  - simple_list_item_activated_2
  - simple_list_item_checked
  - simple_list_item_single_choice
  - simple_list_item_multiple_choice

simple_list_item_1

```
private void updateListView() {
    ListView lvPalabras;
    lvPalabras=(ListView) findViewById(R.id.lvPalabras);
    String[] palabras = { "fichero"
                        , "ficticio"
                        , "boquerón"
                        , "bocata"
                        , "cenizo" };
    Arrays.sort( palabras );
    lvPalabras.setAdapter(
              new ArrayAdapter( this
        , android.R.layout.simple_list_item_1
        , palabras )
                );
}
```

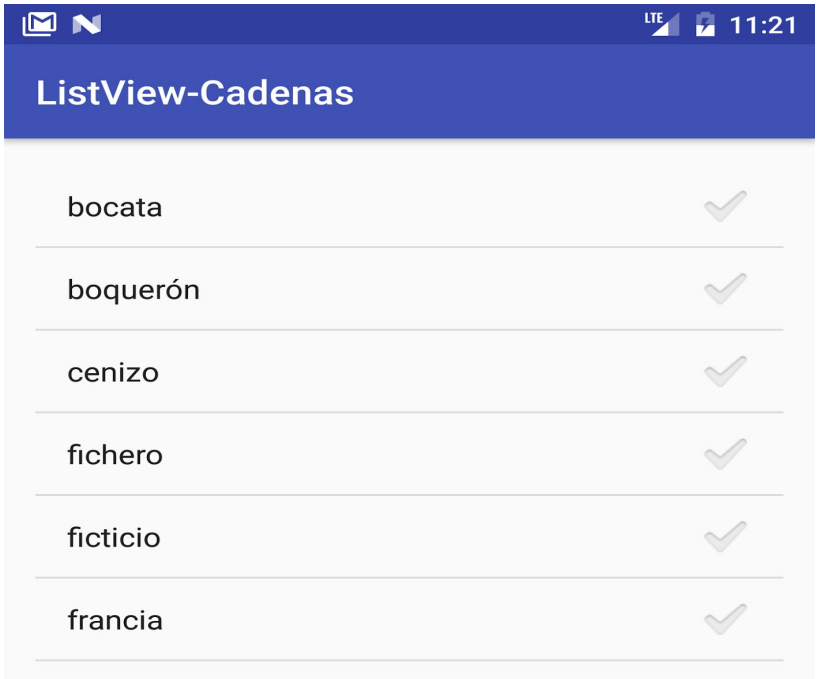


simple_list_item_1 (II)

```
private void updateListView_single_list() {
    ListView lvPalabras;
    lvPalabras=(ListView) findViewById(R.id.lvPalabras);
    final ArrayList<String> palabras = new ArrayList<String>() {{
        add( "fichero" );
        add( "cenicero" );
        add( "bombilla" );
        add( "cable" );
        add( "canasta" );
        add( "bombero" );
        add( "ficticio" );
        add( "caramelo");
    }};

    Collections.sort( palabras );
    lvPalabras.setAdapter( new ArrayAdapter( this
            , android.R.layout.simple_list_item_single_choice
            , palabras ));
}
```

simple_list_item_checked

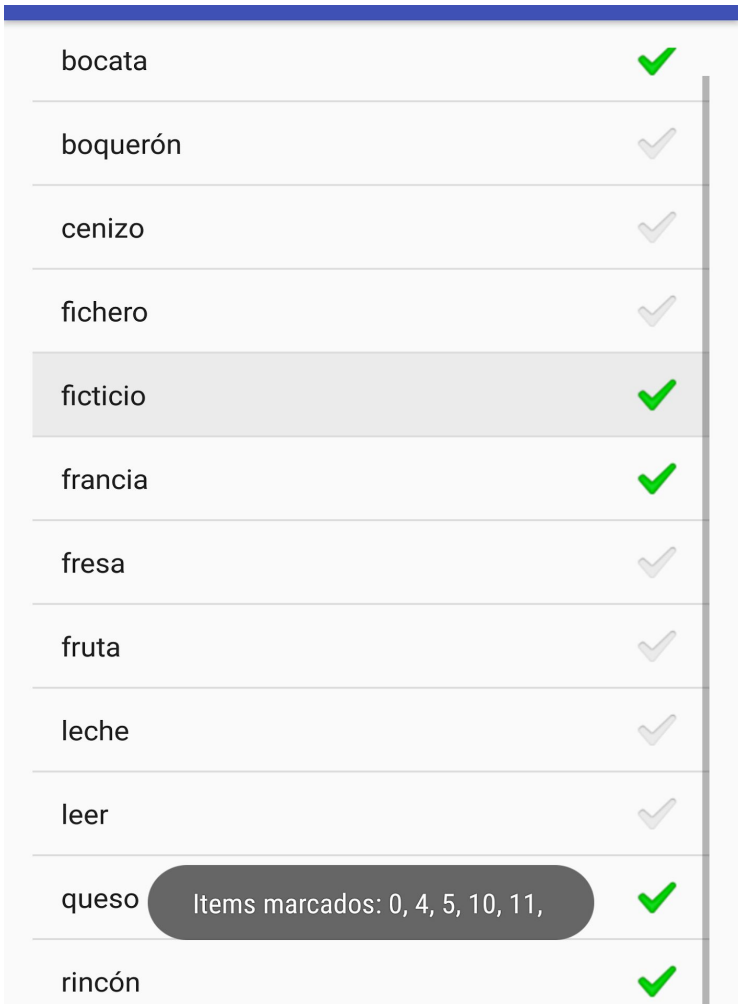
```
lvPalabras.setAdapter( new ArrayAdapter( this
        , android.R.layout.simple_list_item_checked
        , palabras ));
```



simple_list_item_checked (II)

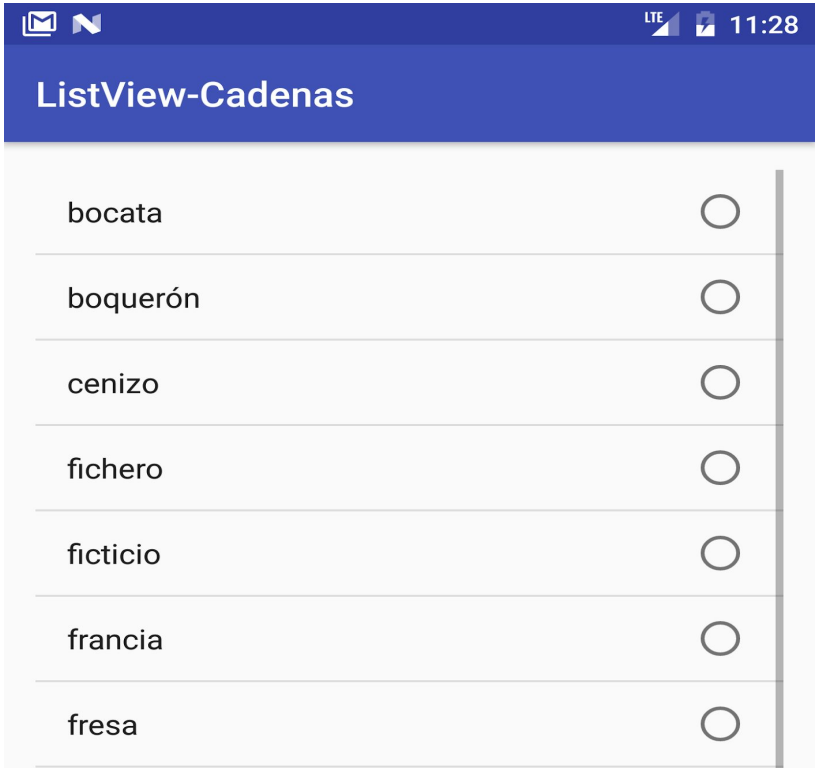
```
lvPalabras.setAdapter( new ArrayAdapter( this
        , android.R.layout.simple_list_item_checked
        , palabras ));
lvPalabras.setChoiceMode(AbsListView.CHOICE_MODE_MULTIPLE);
// OTRAS OPCIONES:
// CHOICE_MODE_NONE, CHOICE_MODE_SINGLE
lvPalabras.setOnItemClickListener(
  new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int position, long arg3) {
     SparseBooleanArray checked = lvPalabras.getCheckedItemPositions();
      String msg="Items marcados: ";
      for( int i=0; i<lvPalabras.getCount(); ++i ) {
          msg+=(checked.get(i))?i+", ":"";
      }
      Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
    }
});
```

simple_list_item_checked (III)



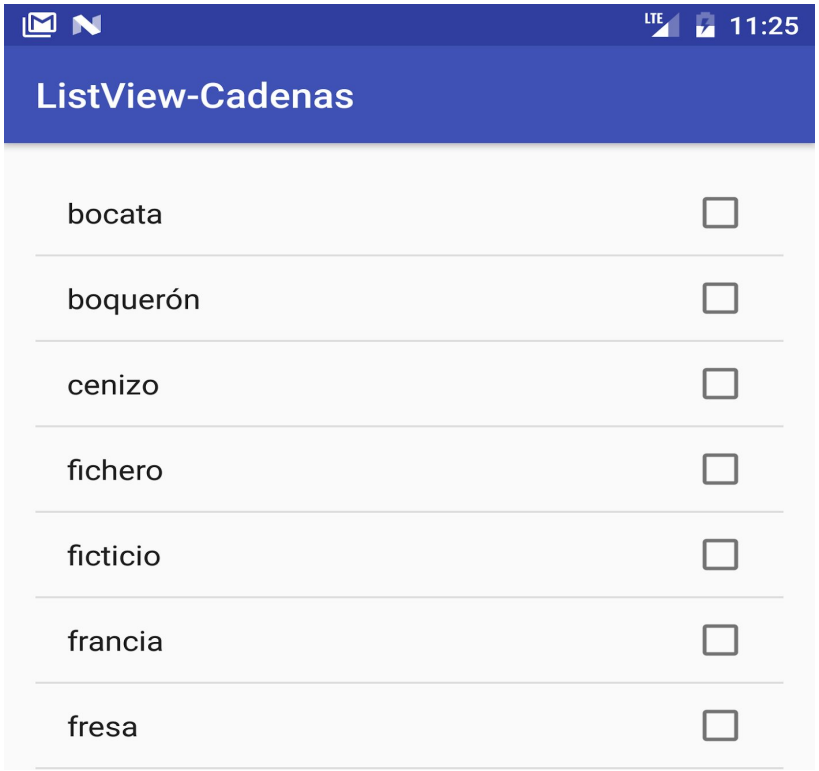
simple_list_item_single_choice

```
lvPalabras.setAdapter( new ArrayAdapter( this
        , android.R.layout.simple_list_item_single_choice
        , palabras ));
```



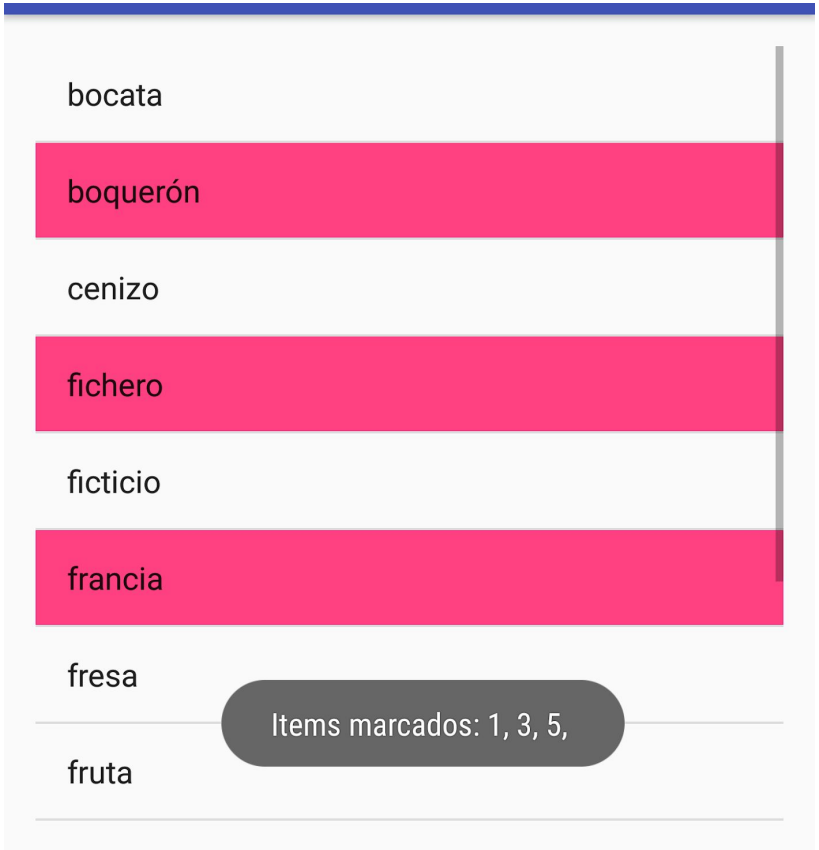
simple_list_item_multiple_choice

```
lvPalabras.setAdapter( new ArrayAdapter( this
        , android.R.layout.simple_list_item_multiple_choice
        , palabras ));
```



simple_list_item_activated_1

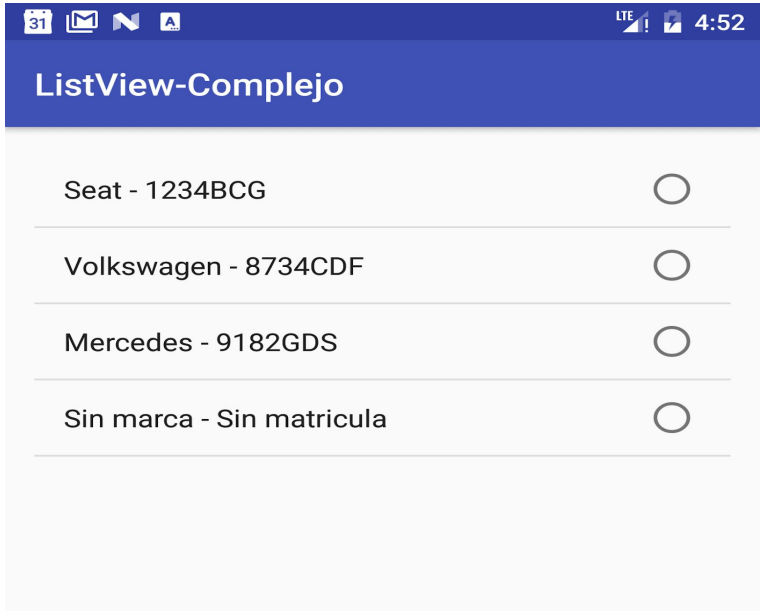
```
lvPalabras.setAdapter( new ArrayAdapter( this
        , android.R.layout.simple_list_item_activated_1
        , palabras ));
```



ListView para objetos: toString

- Sobrecargando el método *toString*, podemos usar los anteriores modos

```
private class Vehiculo {
   private String marca="Sin marca";
   private String matricula="Sin matricula";

   public Vehiculo(String marca
         , String matricula ) {
      this.marca = marca;
      this.matricula = matricula;
   }
   @Override
   public String toString() {
      return this.marca
                        +" - "
                        +this.matricula;
   }
}
```



EJERCICIO

App: mySocialLife

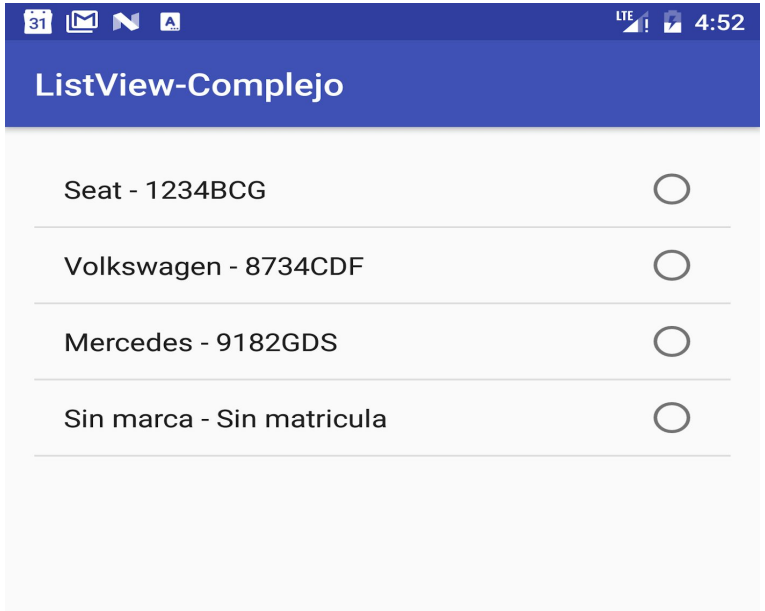
MTV: Practicar con ListViews simples

- Realiza una app en la que haya un ListView que muestre datos básicos del usuario en relación a sus redes sociales.
  - Cada ítem del ListView corresponderá a una red social,
  - La información que indicaremos será: *Nombre de la red – Nombre de usuario en dicha red*: por ejemplo:
    - +----------------------------+
    - | Facebook – victorrivassantos|
    - +----------------------------+
    - | Twitter - @vrsantos |
    - +----------------------------+
- Añade algún mecanismo (botones, radiobuttons…) que permita al usuario ir alternando entre las distintas formas de visualización descritas en las diapositivas anteriores.
- Sube a la plataforma el fichero en el que se implementen los métodos usados para conseguir dichas visualizaciones (que será el MainActivity.java con bastante probabilidad)

ListView para objetos: toString

- Sobrecargando el método *toString*, podemos usar los anteriores modos

```
private class Vehiculo {
   private String marca="Sin marca";
   private String matricula="Sin matricula";

   public Vehiculo(String marca
         , String matricula ) {
      this.marca = marca;
      this.matricula = matricula;
   }
   @Override
   public String toString() {
      return this.marca
                        +" - "
                        +this.matricula;
   }
}
```

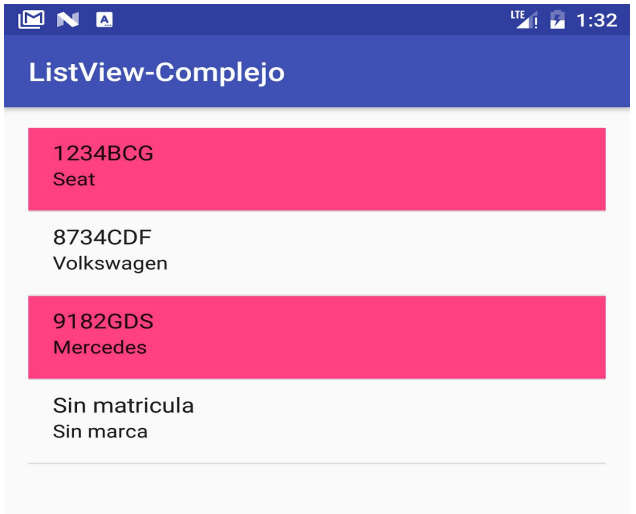


ListView para objetos: simple_list_item_2

```
lvVehiculos.setAdapter(new ArrayAdapter(this
                                        , android.R.layout.simple_list_item_2
                                        , android.R.id.text1
                                        , vehiculos)
{
   @Override
   public View getView(int position, View convertView, ViewGroup parent) {
      View view = super.getView(position, convertView, parent);


      ((TextView) view.findViewById(android.R.id.text1))
                                    .setText(vehiculos.get(position).getMatricula());


      ((TextView) view.findViewById(android.R.id.text2))
                                    .setText(vehiculos.get(position).getMarca());
      return view;
   } // Fin getView
}); // Fin setAdapter
```

ListView para objetos: simple_list_item_activated_2

```
lvVehiculos.setAdapter(new ArrayAdapter(this
                                , android.R.layout.simple_list_item_activated_2
                                , android.R.id.text1
                                , vehiculos){
   @Override
   public View getView(int position, View convertView, ViewGroup parent) {
      View view = super.getView(position, convertView, parent);


      ((TextView) view.findViewById(android.R.id.text1))
                              .setText(vehiculos.get(position).getMatricula());


      ((TextView) view.findViewById(android.R.id.text2))
                              .setText(vehiculos.get(position).getMarca());
      return view;
   } // Fin getView
}); // Fin setAdapter
```

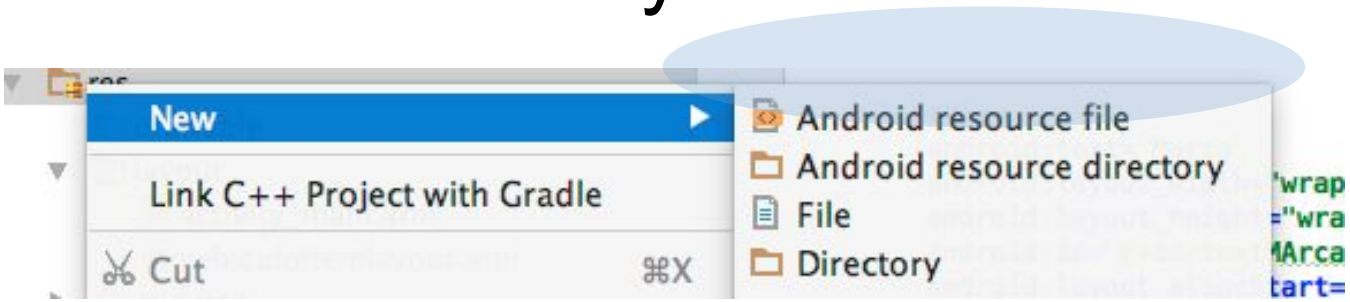


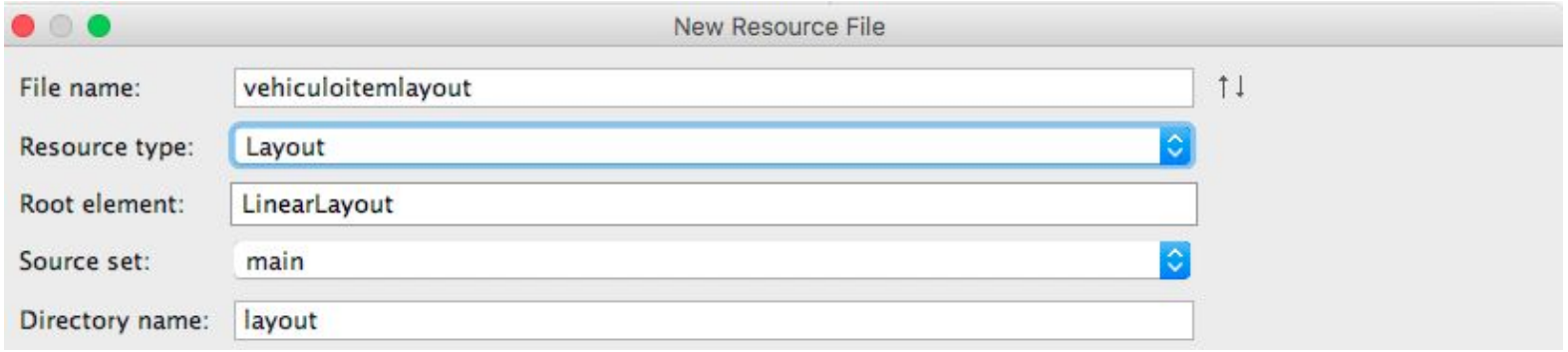
ListView para objetos: layouts personalizados

- Podemos crear un layout que va a dar forma a los items del *ListView*
- Una vez creado, debemos programar un adapter que utilice los elementos de ese *layout*

Layouts personalizados (a)

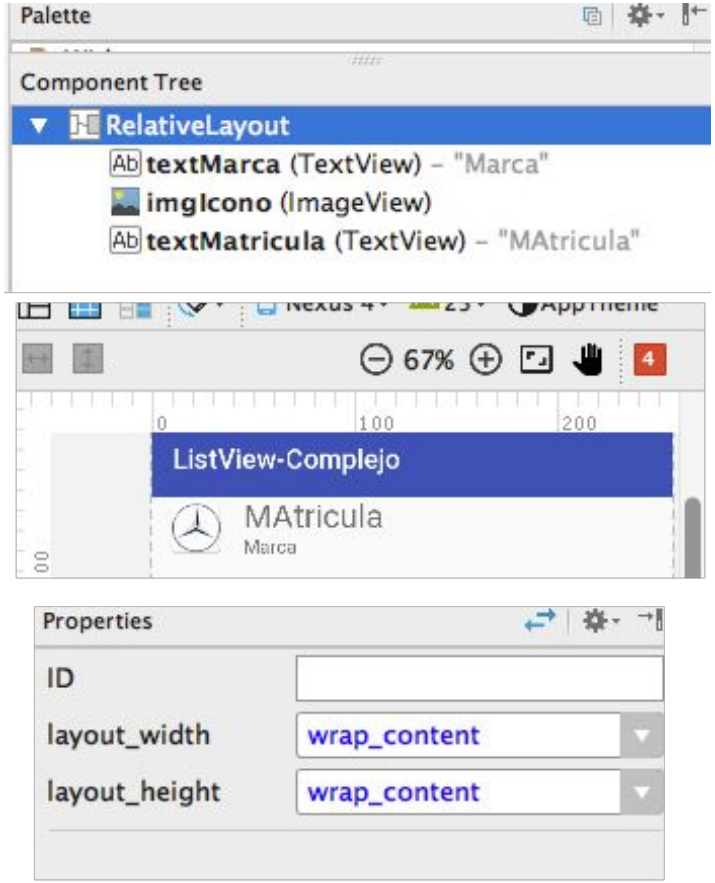
- Paso 1: Crear el layout





Layouts personalizados (b)

- Paso 1: Crear el layout (cont.)

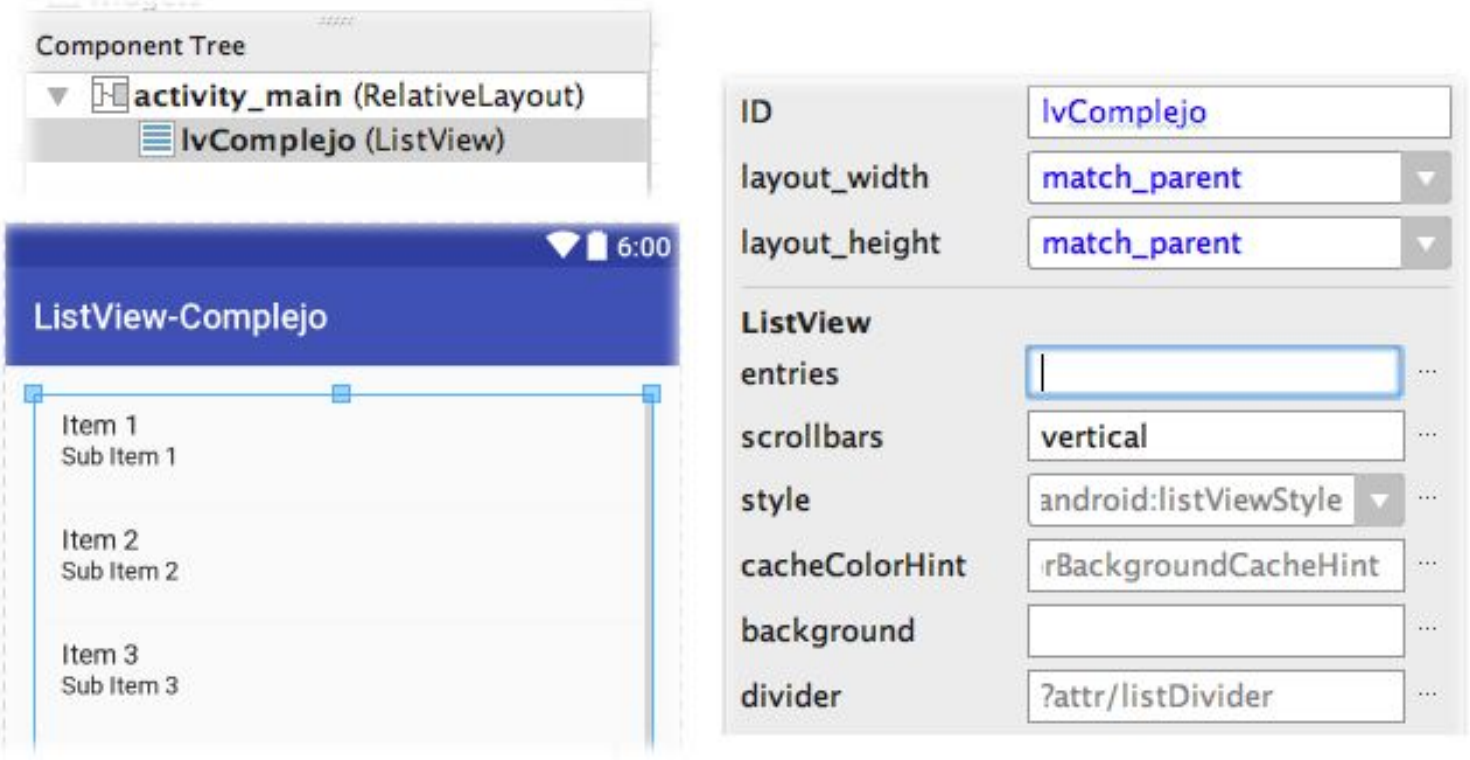


Layouts personalizados (c)

- Paso 2: Para crear las imágenes:
  - *New* > *Image Asset*
  - Elegir fichero png o vectorial (https://material.io/icons/)
  - Poner un nombre (p.ej.: *ic_seat*)
  - Se crearán dentro de res/mimap

Layouts personalizados (d)

- Paso 3: Crear el listview dentro de la activity



Layouts personalizados (e)

- Paso 4: Preparar la fuente de datos

```
private class Vehiculo {
  private String marca="Sin marca";
  private String matricula="Sin matricula";
  private int icono=android
        .R
        .mipmap
        .sym_def_app_icon;

public Vehiculo(String marca
        , String matricula
        , int icono) {
    this.marca = marca;
    this.matricula = matricula;
    this.icono = icono;
}

public String getMarca() {
    return marca;
}

public String getMatricula() {
    return matricula;
}

public int getIcono() {
    return icono;
}
```

```
private int populateVehiculos() {
    vehiculos=new ArrayList<Vehiculo>();

    vehiculos.add(
          new Vehiculo( "Seat"
                        , "1234BCG"
                        , R.mipmap.ic_seat) );

    vehiculos.add(
      new Vehiculo( "Volkswagen"
                        , "8734CDF"
                        , R.mipmap.ic_vw) );

    vehiculos.add(
          new Vehiculo( "Mercedes"
                        , "9182GDS"
                        , R.mipmap.ic_mercedes) );

    vehiculos.add( new Vehiculo() );

    return vehiculos.size();
}
```

Layouts personalizados (f)

- Paso 5: Instanciar elementos del listview

```
private void showLVVehiculos_Complejo() {
   lvVehiculos=(ListView) findViewById(R.id.lvComplejo);

   ArrayAdapter adaptadorVehiculos=
         new ArrayAdapter( this, R.layout.vehiculoitemlayout, vehiculos) {
            public View getView(int position
                                          , View convertView
                                          , ViewGroup parent) {
               LayoutInflater inflater = (LayoutInflater) getContext()
                     .getSystemService(getContext().LAYOUT_INFLATER_SERVICE);

               // Creamos la vista para cada fila
               View fila = inflater.inflate(R.layout.vehiculoitemlayout, parent, false);

               // Creamos cada uno de los widgets que forman una fila
               ImageView iconoView = (ImageView) fila.findViewById(R.id.imgIcono);
               TextView marcaView = (TextView) fila.findViewById(R.id.textMarca);
               TextView matriculaView = (TextView) fila.findViewById(R.id.textMatricula);

               // Establemcemos los valores que queremos que muestren los widgets
               iconoView.setImageResource(vehiculos.get(position).getIcono());
               marcaView.setText(vehiculos.get(position).getMarca());
               matriculaView.setText(vehiculos.get(position).getMatricula());
               return fila;
            }
   };
   lvVehiculos.setAdapter(adaptadorVehiculos);
```

Layouts personalizados (g)

- Paso 6: Definir color para item(s) seleccionado(s)


```
<!-- FICHERO: res/values/colors.xml →

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="colorAzulito">#8140FF</color>
</resources>
```

```
<!-- FICHERO: res/drawable/miscolores.xml -->
<!-- Sobre res/drawable: menú contextual > new > Drawable Resource File

<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_activated="true" android:drawable="@color/colorAzulito"/>
</selector>
```

Layouts personalizados (h)

- Paso 7: *onClickListener* es similar al visto para *checked*

```
lvVehiculos.setAdapter(adaptadorVehiculos);
lvVehiculos.setOnItemClickListener(
        new AdapterView.OnItemClickListener() {
          @Override
          public void onItemClick( AdapterView<?> adapterView
                                                  , View view
                                                  , int position
                                                  , long l ) {
         SparseBooleanArray checked = lvVehiculos.getCheckedItemPositions();
         String msg = "Items marcados: ";
         for (int i = 0; i < lvVehiculos.getCount(); ++i) {
            msg += (checked.get(i)) ? i + ", " : "";
         }
        Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
    }
});
```

EJERCICIO

App: mySocialLife 2

MTV: Practicar con ListViews complejos

- Modifica o realiza una nueva versión de *mySocialLife* en la cual gestiones tanta información como desees para cada una de las redes sociales, incluyendo al menos:
  - Logo de la red social
  - Nombre de usuario en ella
  - Última fecha y hora de conexión
- Nuevamente, permite que el usuario pueda alternar entre varias formas de presentar la información incluyendo una que corresponda a un layout personalizado.
- Sube a la plataforma un fichero PDF que contengan una captura de pantalla de la aplicación funcionando, y a continuación el código en Java que has implementado para usar el layout personalizado.

EJERCICIO

App: mySocialLife 3

MTV: Practicar con ListViews complejos

- Modifica o realiza una nueva versión de *mySocialLife 2* añadiendo un *checkbox* a cada item de modo que se destaquen los que están marcados NO por el color de fondo, sino por el estado del *checkbox*
- Como en el caso anterior, sube un fichero PDF con captura de pantalla y código en Java.

  Si lo deseas, puedes cambiar el *checkbox* por los iconos:

  - https://material.io/icons/#ic_check_box
  - https://material.io/icons/#ic_check_box_outline_blank