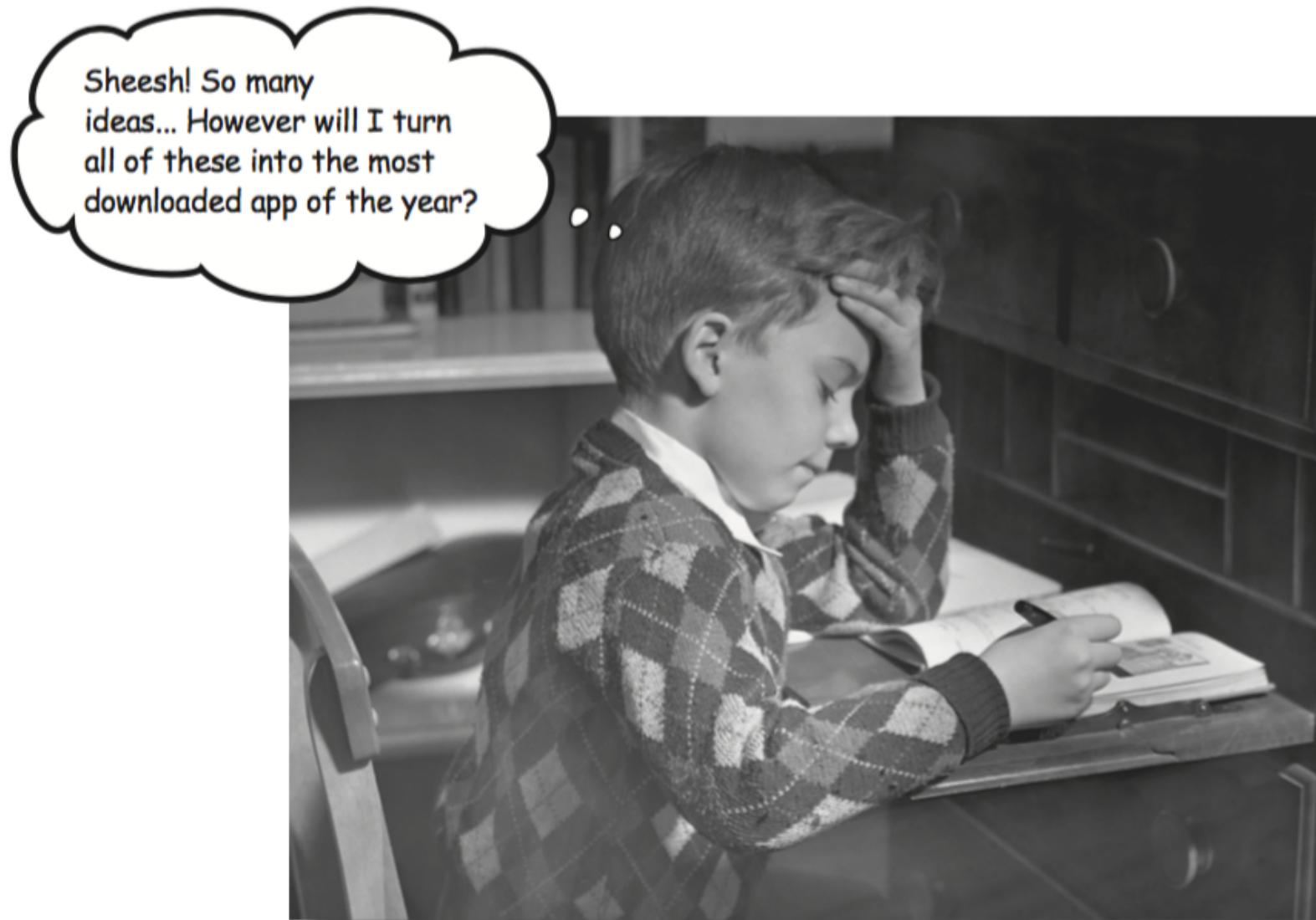


## 7

# Listas y adaptadores



# Cada aplicación empieza con algunas ideas

Show details of each drink.

Show a list of all our stores.

Display a menu showing all the food you can buy.

Display a list of the drinks we sell.

Show details of an item of food.

Display the address and opening times of each store.

Display a start screen with a list of options.

# Categorización de las ideas

Display a start screen with a list of options.

Display a menu showing all the food you can buy.

Show a list of all our stores.

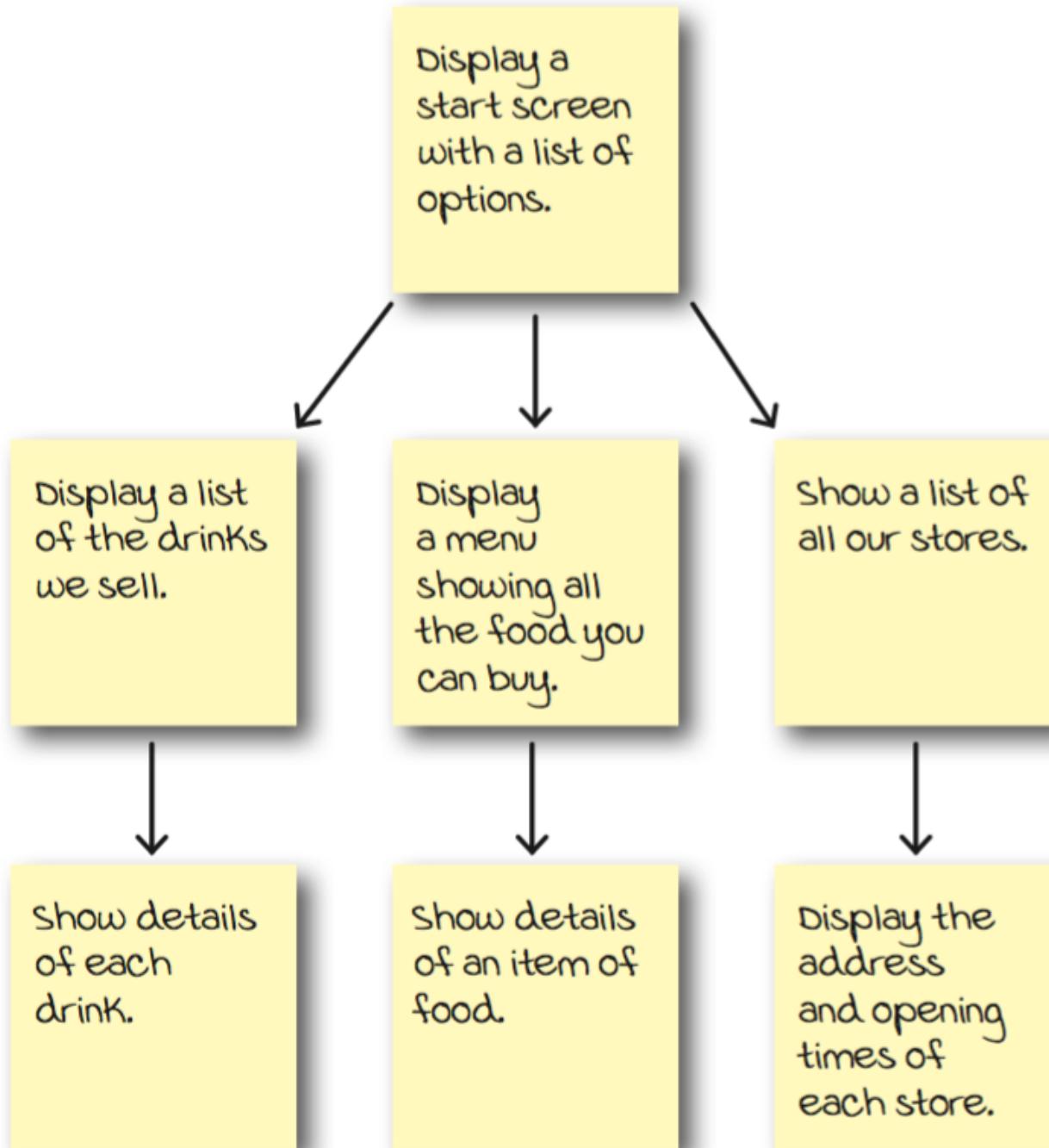
Display a list of the drinks we sell.

Show details of each drink.

Show details of an item of food.

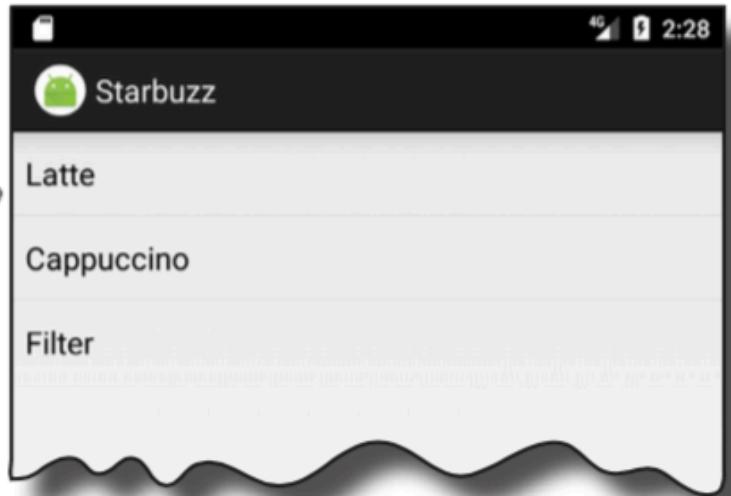
Display the address and opening times of each store.

# Navegando por las actividades

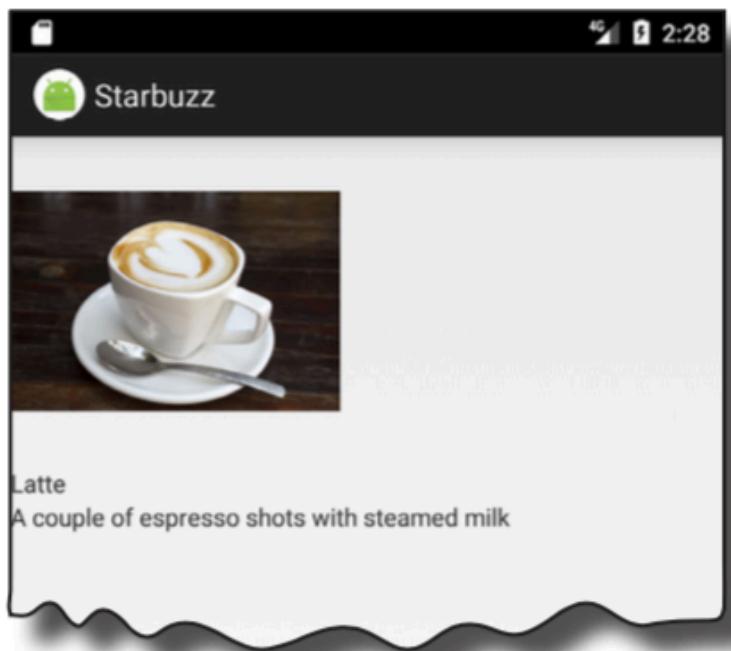


# ListView para navegar por las actividades

This is a list view  
containing a list  
of drinks.



If you click on the Latte option in the list view, you get shown the details for the latte.



# Construyendo la aplicación starbuzz

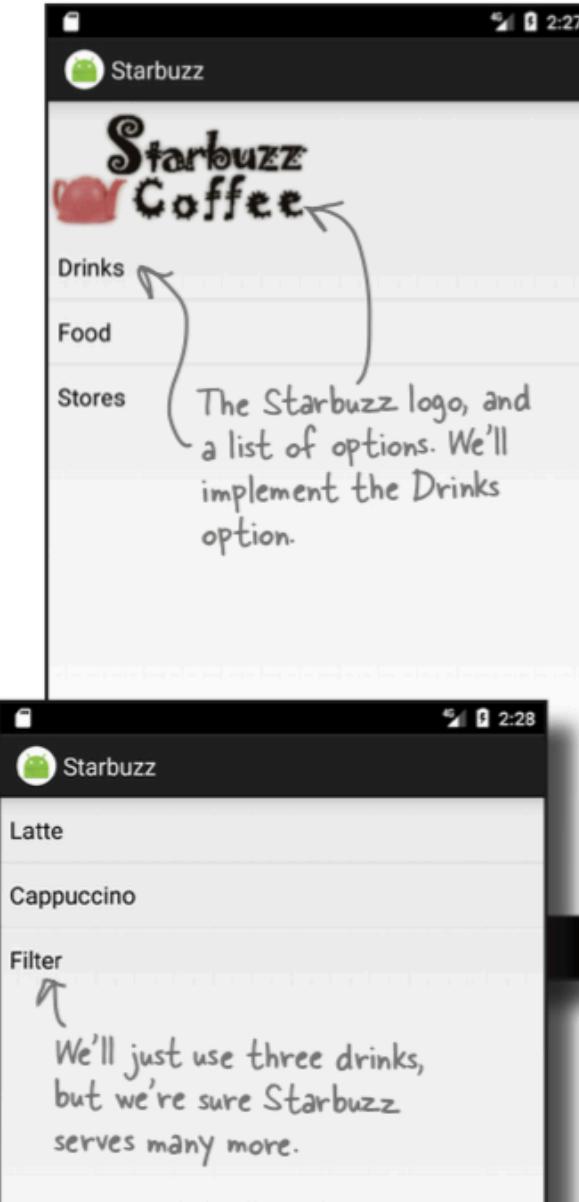
## The top-level activity

When the user launches the app, she will be presented with the top-level activity, the main entry point of the app. This activity includes an image of the Starbuzz logo, and a navigational list containing entries for Drinks, Food, and Stores.

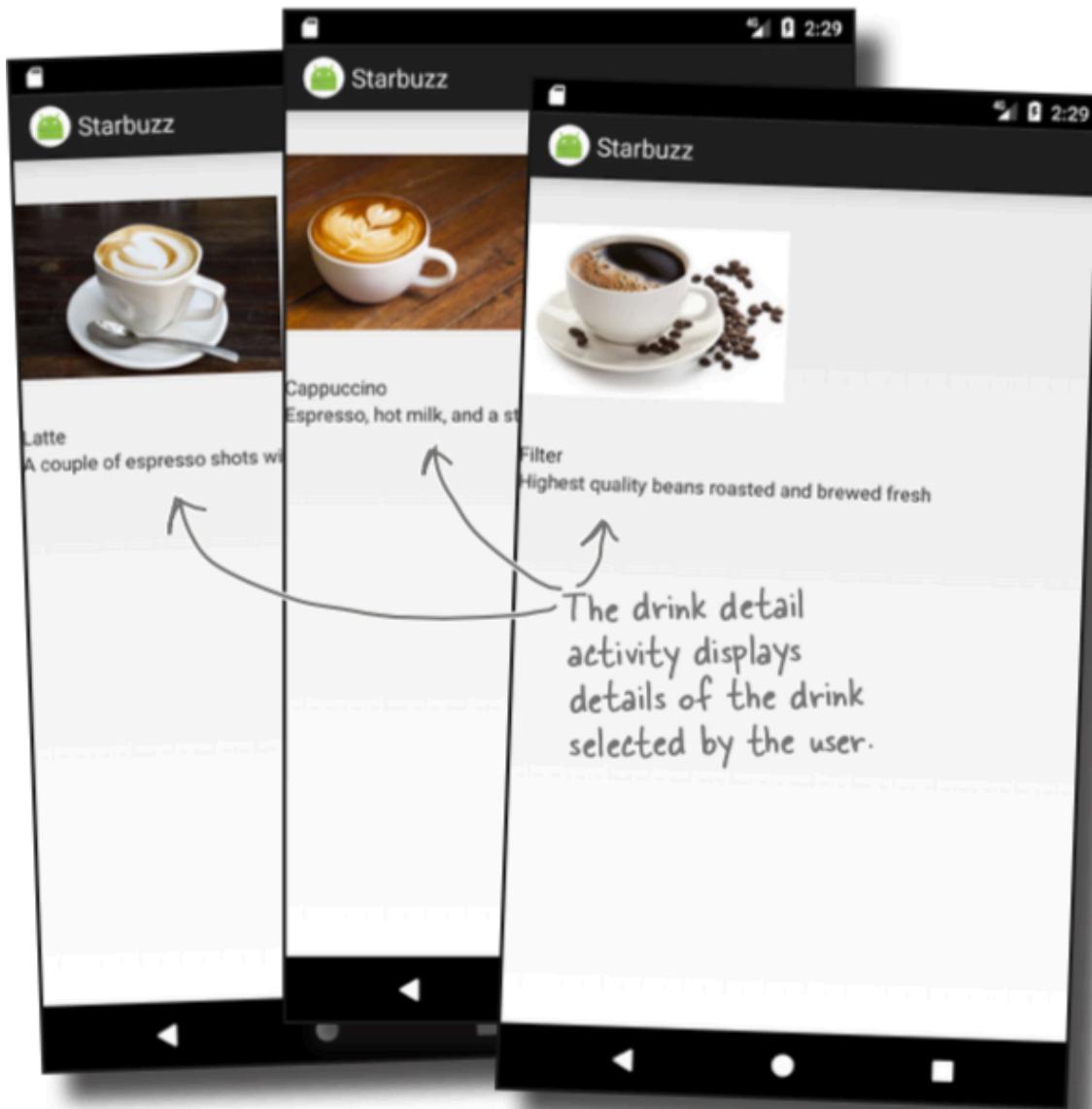
When the user clicks on an item in the list, the app uses her selection to navigate to a separate activity. As an example, if the user clicks on Drinks, the app starts a category activity relating to drinks.

## The drink category activity

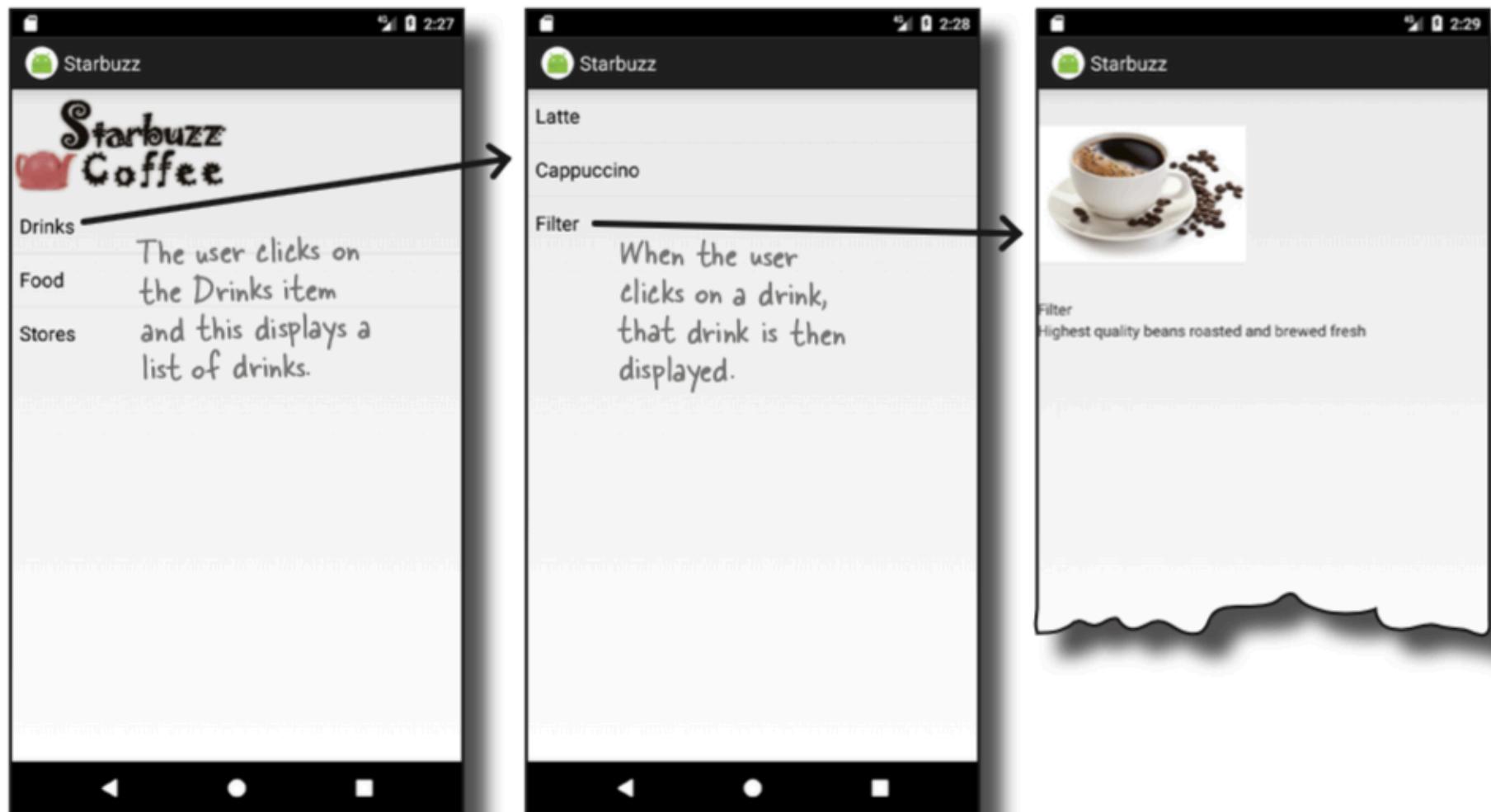
This activity is launched when the user chooses Drinks from the navigational list in the top-level activity. The activity displays a list of all the drinks that are available at Starbuzz. The user can click on one of these drinks to see more details of it.



# La actividad para los detalles de la bebida



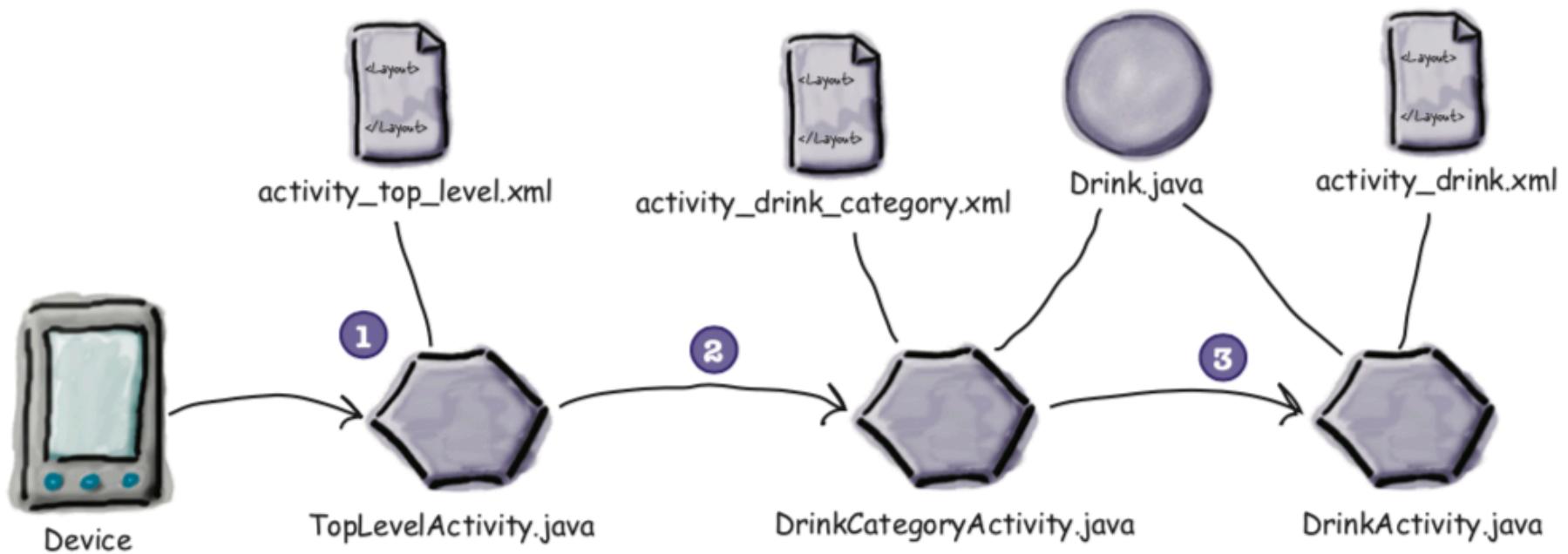
# Navegando por las actividades



# Estructura de la aplicación

- 1 When the app gets launched, it starts activity `TopLevelActivity`.**  
This activity uses layout `activity_top_level.xml`. The activity displays a list of options for Drinks, Food, and Stores.
- 2 The user clicks on Drinks in `TopLevelActivity`, which launches activity `DrinkCategoryActivity`.**  
This activity uses layout `activity_drink_category.xml` and displays a list of drinks. It gets information about the drinks from the `Drink.java` class file.
- 3 The user clicks on a drink in `DrinkCategoryActivity`, which launches activity `DrinkActivity`.**  
The activity uses layout `activity_drink.xml`. This activity also gets details about the drinks from the `Drink.java` class file.

# Estructura de la aplicación



# Pasos para construir la aplicación

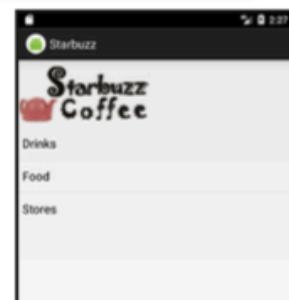
## 1 Add the Drink class and image resources.

This class contains details of the available drinks, and we'll use images of the drinks and Starbuzz logo in the app.



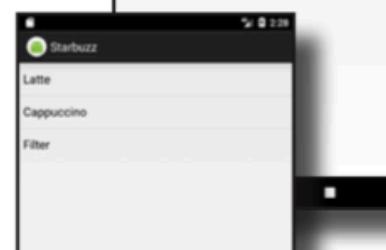
## 2 Create TopLevelActivity and its layout.

This activity is the entry point for the app. It needs to display the Starbuzz logo and include a navigational list of options. TopLevelActivity needs to launch DrinkCategoryActivity when the Drinks option is clicked.



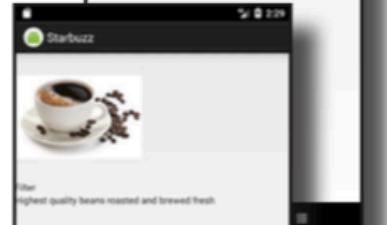
## 3 Create DrinkCategoryActivity and its layout.

This activity contains a list of all the drinks that are available. When a drink is clicked, it needs to launch DrinkCategoryActivity.



## 4 Create DrinkActivity and its layout.

This activity displays details of the drink the user clicked on in DrinkCategoryActivity.



# La clase Drink

```
package com.hfad.starbuzz;

public class Drink {
    private String name;
    private String description;
    private int imageResourceId;

    //drinks is an array of Drinks
    public static final Drink[] drinks = {
        new Drink("Latte", "A couple of espresso shots with steamed milk",
                  R.drawable.latte),
        new Drink("Cappuccino", "Espresso, hot milk, and a steamed milk foam",
                  R.drawable.cappuccino),
        new Drink("Filter", "Highest quality beans roasted and brewed fresh",
                  R.drawable.filter)
    };

    //Each Drink has a name, description, and an image resource
    private Drink(String name, String description, int imageResourceId) {
        this.name = name;
        this.description = description;
        this.imageResourceId = imageResourceId;
    }
}
```

These are images of the drinks. We'll add these next.

Each Drink has a name, description, and image resource ID. The image resource ID refers to drink images we'll add to the project on the next page.

drinks is an array of three Drinks.

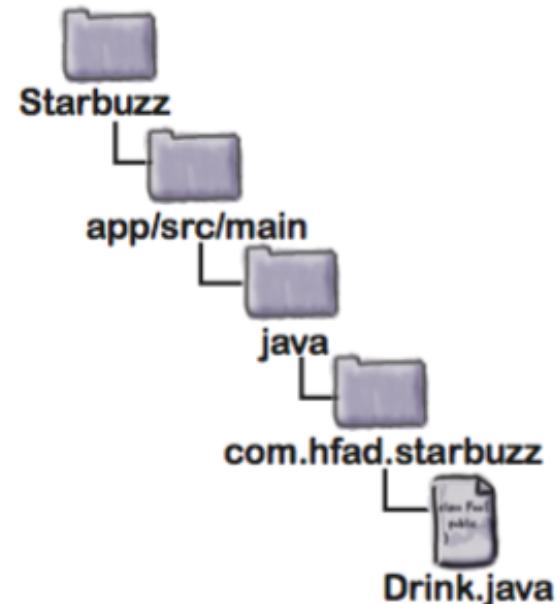
The Drink constructor

# La clase Drink

```
public String getDescription() {  
    return description;  
}  
  
public String getName() {  
    return name;  
}  
  
public int getImageResourceId() {  
    return imageResourceId;  
}  
  
public String toString() {  
    return this.name;  
}
```

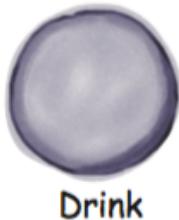
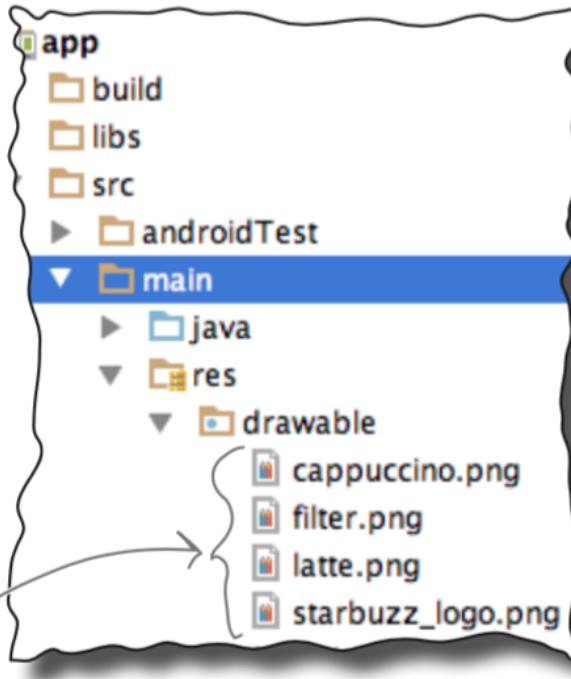
These are getters for the private variables.

The String representation of a Drink is its name.



# Los archivos de imágenes

Here are the four image files. You add them to Android Studio by dragging them to the drawable folder.



name: "Latte"

description: "A couple of espresso shots with steamed milk"

imageResourceId: R.drawable.latte

The image latte.png  
is given an ID of  
R.drawable.latte.

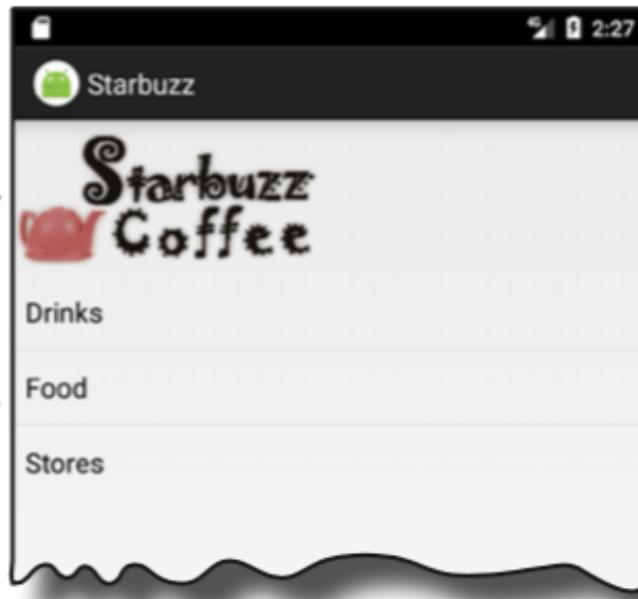


R.drawable.latte

# EL nivel mas alto contiene una imagen y una lista

This is the Starbuzz logo.  
We added this image to the →  
project on the previous page.

We'll add these items as →  
a static list of options;  
then we'll make each list  
item clickable.



# EL nivel mas alto contiene una imagen y una lista

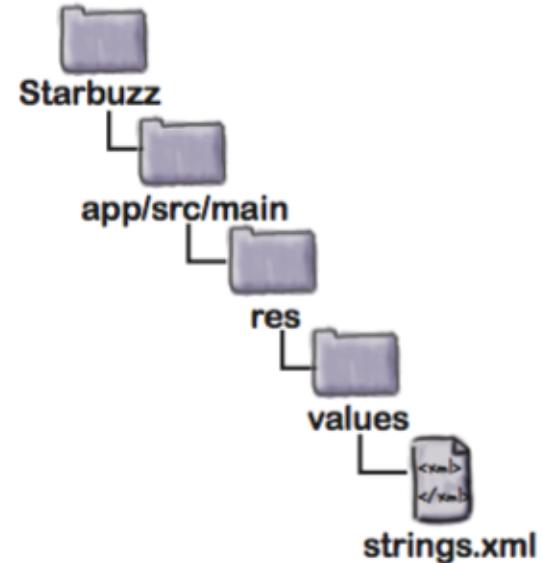
```
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="100dp"  
    android:src="@drawable/starbuzz_logo"  
    android:contentDescription="@string/starbuzz_logo" />
```

These are the dimensions we want the image to have.

The source of the image is the starbuzz\_logo.png file we added to the app.

Adding a content description makes your app more accessible.

```
<resources>  
    ...  
    <string name="starbuzz_logo">Starbuzz logo</string>  
</resources>
```



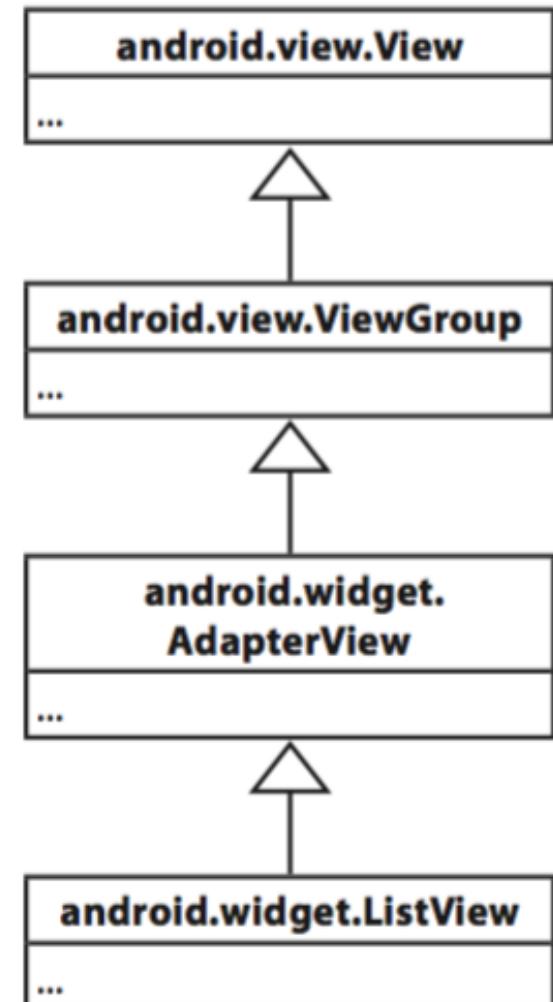
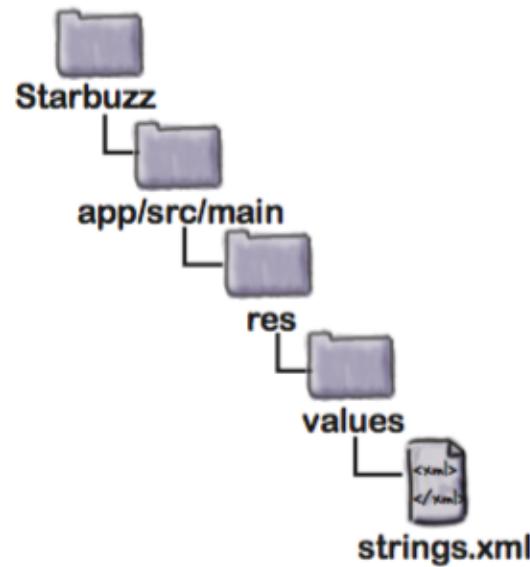
# Utilizando ListView para mostrar opciones

```
<ListView  
    android:id="@+id/list_options"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:entries="@array/options" />
```

This defines the list view.

The values in the list view are defined by the options array.

```
<resources>  
    ...  
    <string-array name="options">  
        <item>Drinks</item>  
        <item>Food</item>  
        <item>Stores</item>  
    </string-array>  
</resources>
```



# Utilizando arreglos para poblar un ListView

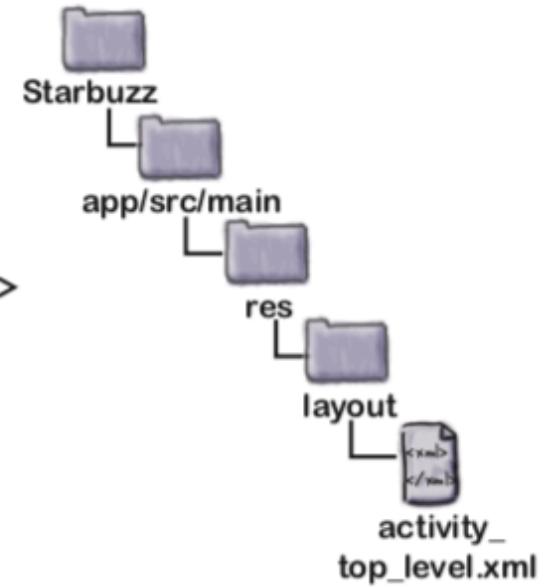


The `entries` attribute populates the `ListView` with values from the `options` array. Each item in the `ListView` is a `TextView`.

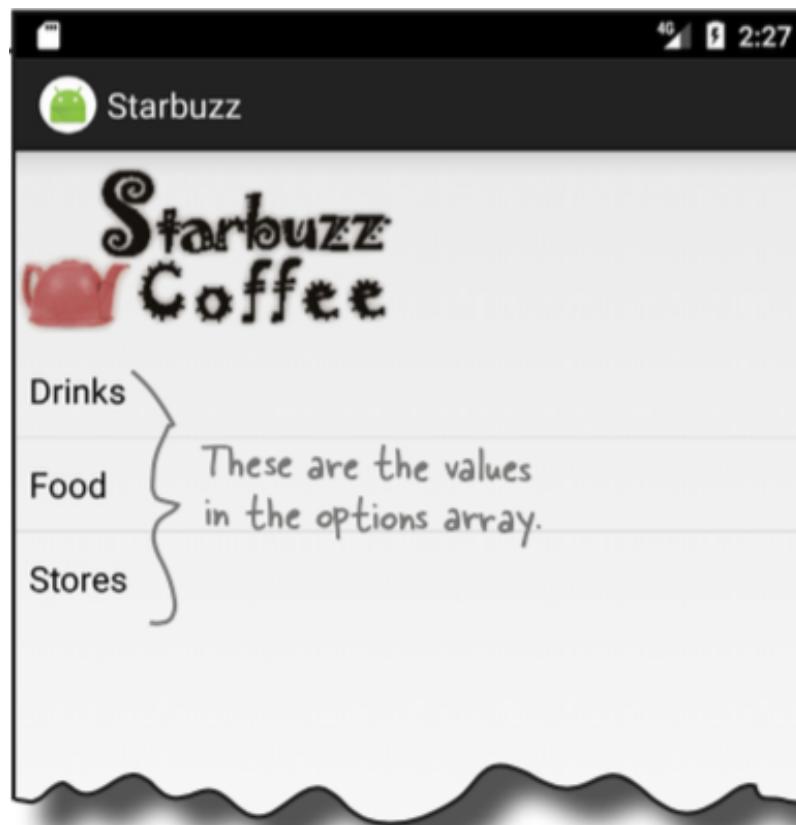


# EL código completo del layout

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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:orientation="vertical" ← We're using a linear layout with a vertical  
        orientation. This will display our list view  
        directly underneath the Starbuzz logo.  
    tools:context="com.hfad.starbuzz.TopLevelActivity" >  
  
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="100dp"  
    android:src="@drawable/starbuzz_logo"  
    android:contentDescription="@string/starbuzz_logo" />  
  
<ListView  
    android:id="@+id/list_options"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:entries="@array/options" />  
  
</LinearLayout>
```



# Prueba



# Respondiendo a eventos

```
AdapterView.OnItemClickListener itemClickListener = new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> listView, View itemView, int position, long id) {  
        if (position == 0) {  
            Intent intent = new Intent(TopLevelActivity.this, DrinkCategoryActivity.class);  
            startActivity(intent);  
        }  
    };  
};
```

Drinks is the first item in the list view, so it's at position 0.

OnItemClickListener is a nested class within the AdapterView class. A ListView is a subclass of AdapterView.

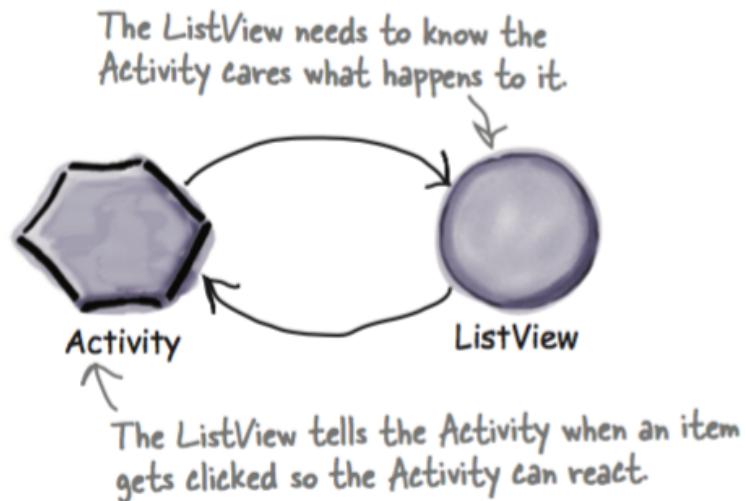
The view that was clicked (in this case, the list view).

These parameters give you info about which item was clicked in the list view, such as the item's view and its position.

The intent is coming from TopLevelActivity.

It needs to launch DrinkCategoryActivity.

# Respondiendo a eventos



```
AdapterView.OnItemClickListener itemClickListener = new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> listView,  
        ...  
    }  
};  
ListView listView = (ListView) findViewById(R.id.list_options);  
listView.setOnItemClickListener(itemClickListener);
```

This is the listener we just created.

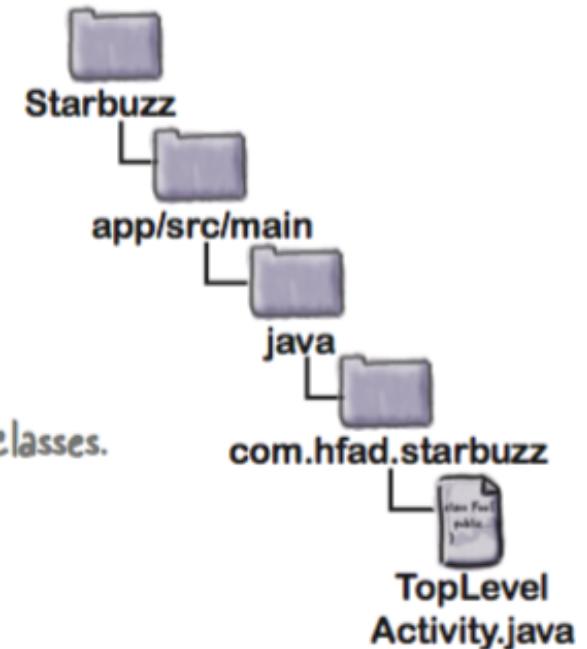
# El código completo de la actividad

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ListView;
import android.view.View;
```



We're using these extra classes.



# El código completo de la actividad

```
public class TopLevelActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_top_level);  
        //Create an OnItemClickListener  
        AdapterView.OnItemClickListener itemClickListener =  
            new AdapterView.OnItemClickListener() {  
                public void onItemClick(AdapterView<?> listView,  
                    View v,  
                    int position,  
                    long id) {  
                    if (position == 0) {  
                        Intent intent = new Intent(TopLevelActivity.this,  
                            DrinkCategoryActivity.class);  
                        startActivity(intent);  
                    }  
                };  
                //Add the listener to the list view  
                ListView listView = (ListView) findViewById(R.id.list_options);  
                listView.setOnItemClickListener(itemClickListener);  
            }  
}
```

*Create the listener.*

*Implement its onItemClick() method.*

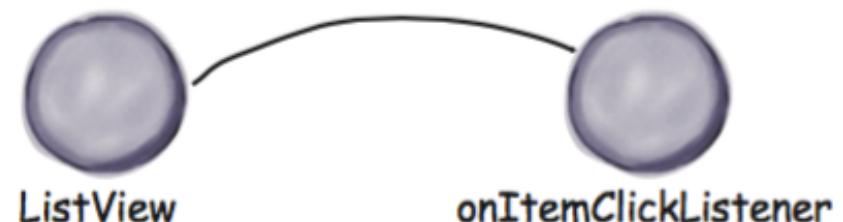
*Launch DrinkCategoryActivity if the user clicks on the Drinks item. We'll create this activity next, so don't worry if Android Studio says it doesn't exist.*

*Add the listener to the list view.*

# Navegando por las actividades

1

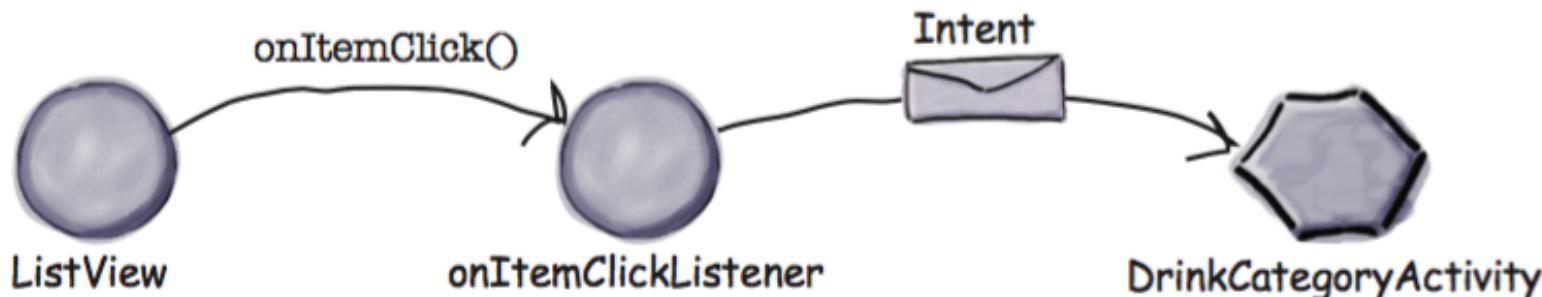
The `onCreate()` method in `TopLevelActivity` creates an `onItemClickListener` and links it to the activity's `ListView`.



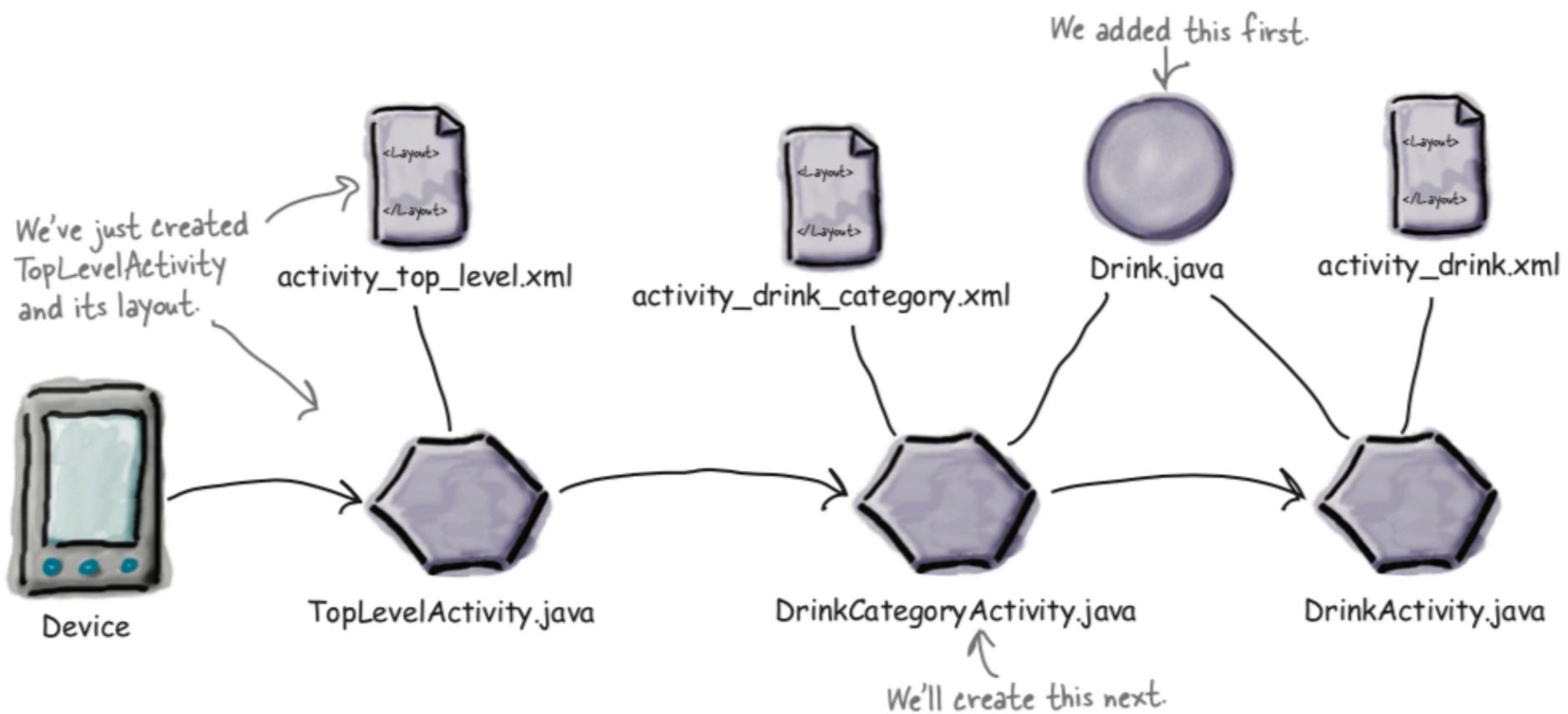
2

When the user clicks on an item in the list view, the `onItemClickListener`'s `onItemClick()` method gets called.

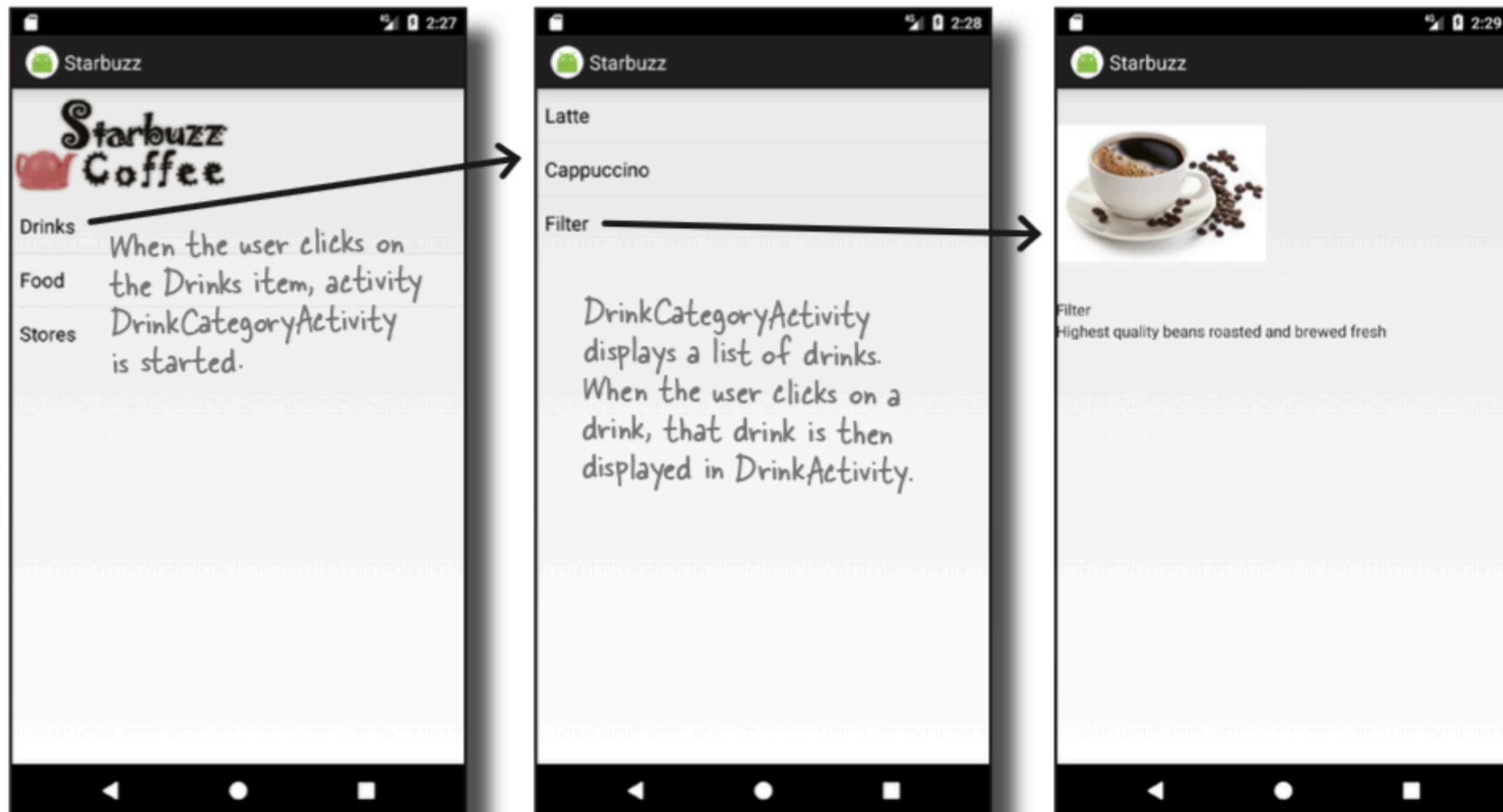
If the Drinks item is clicked, the `onItemClickListener` creates an intent to start `DrinkCategoryActivity`.



# Hacia donde vamos



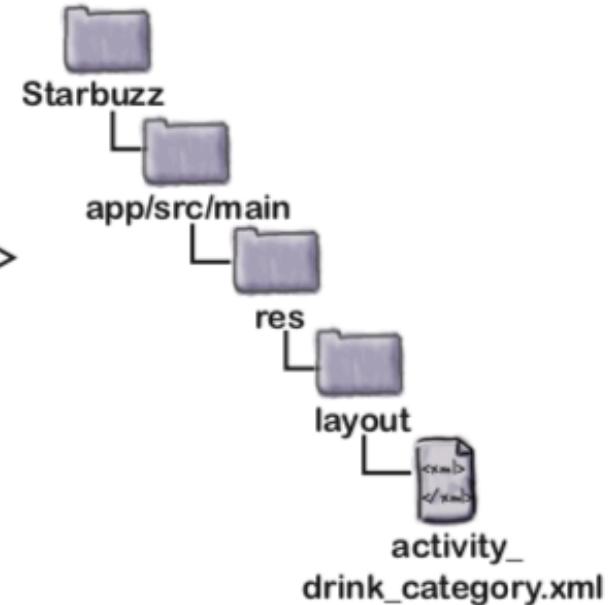
# Una actividad categoría muestra datos de una sola categoría



# Actualizando activity\_drink\_category.xml

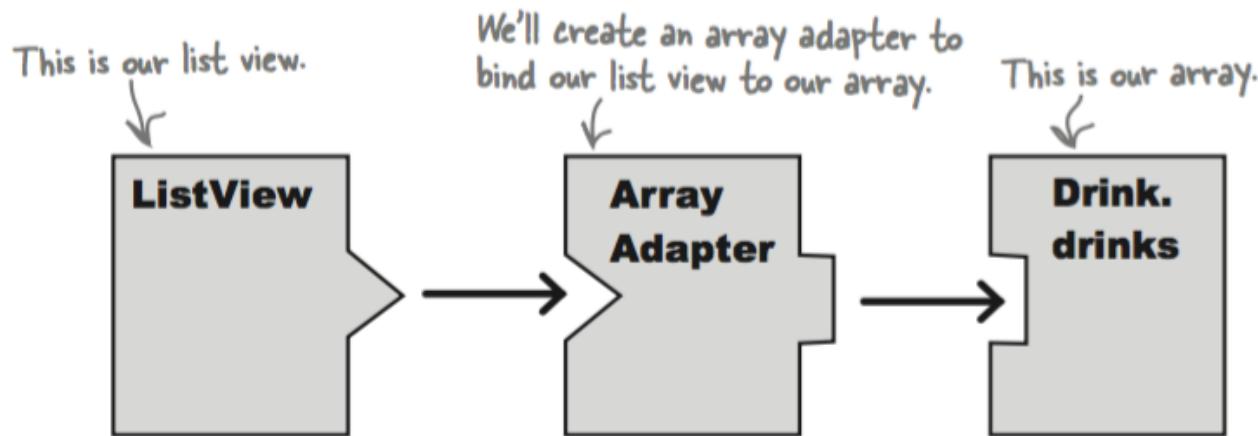
```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    tools:context="com.hfad.starbuzz.DrinkCategoryActivity">  
  
    <ListView  
        android:id="@+id/list_drinks"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />  
  
</LinearLayout>
```

This layout only needs  
to contain a ListView.



# Conectando un ListView con un arreglo

An adapter acts as a bridge between a view and a data source. An **ArrayAdapter** is a type of adapter that specializes in working with arrays.



# Agregando el ArrayAdapter a la clase DrinkCategoryActivity

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class DrinkCategoryActivity extends Activity {

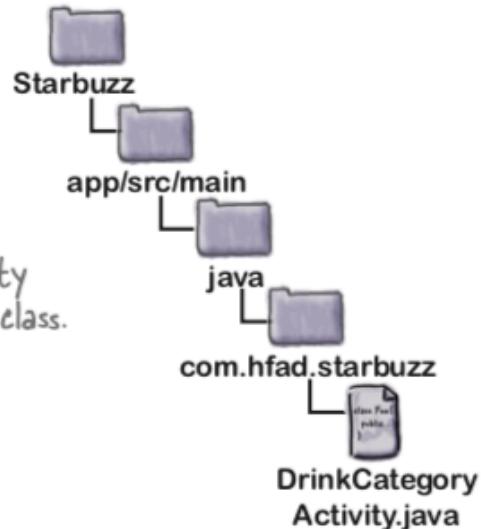
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drink_category);

        ArrayAdapter<Drink> listAdapter = new ArrayAdapter<>(
            this,
            android.R.layout.simple_list_item_1,
            Drink.drinks);
        ListView listDrinks = (ListView) findViewById(R.id.list_drinks);
        listDrinks.setAdapter(listAdapter);
    }
}
```

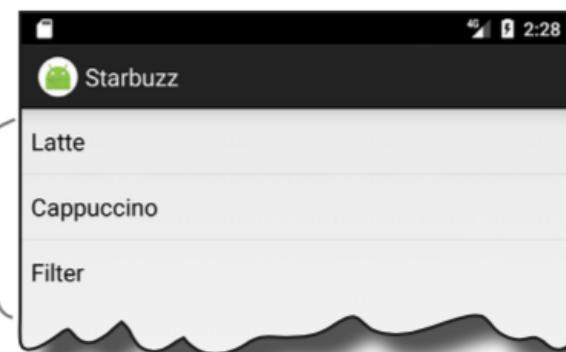
We're using these classes so we need to import them.

Make sure your activity extends the Activity class.

This populates the list view with data from the drinks array.

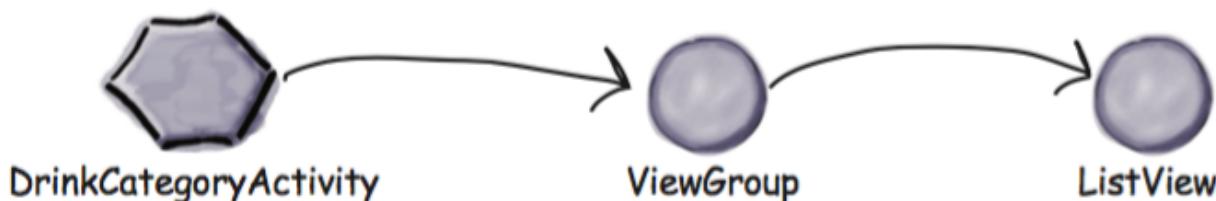


These are the drinks from the drinks array. Each row in the list view is a single text view, each one displaying a separate drink.



# Que pasa al ejecutar el código

- 1 When the user clicks on the Drinks option, `DrinkCategoryActivity` is launched. As `DrinkCategoryActivity` is a list activity, it has a default layout containing a single `ListView` object. This layout is created behind the scenes in Java code, so it's not defined by XML.



- 2 `DrinkCategoryActivity` creates an `ArrayAdapter<Drink>`, an array adapter that deals with arrays of `Drink` objects.

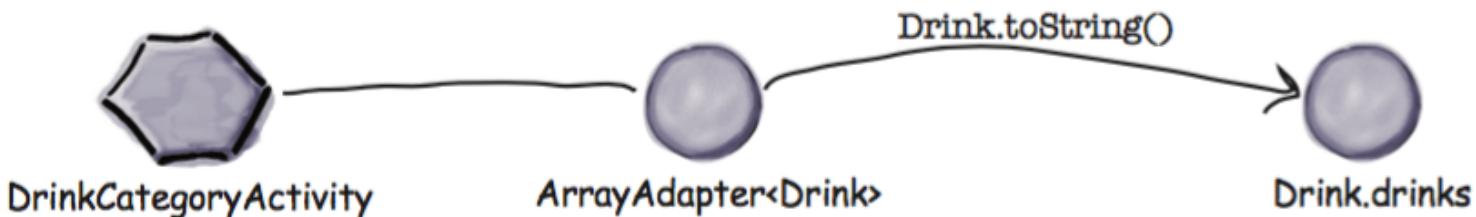


# Que pasa al ejecutar el código

3

**The array adapter's source is the drinks array in the Drink class.**

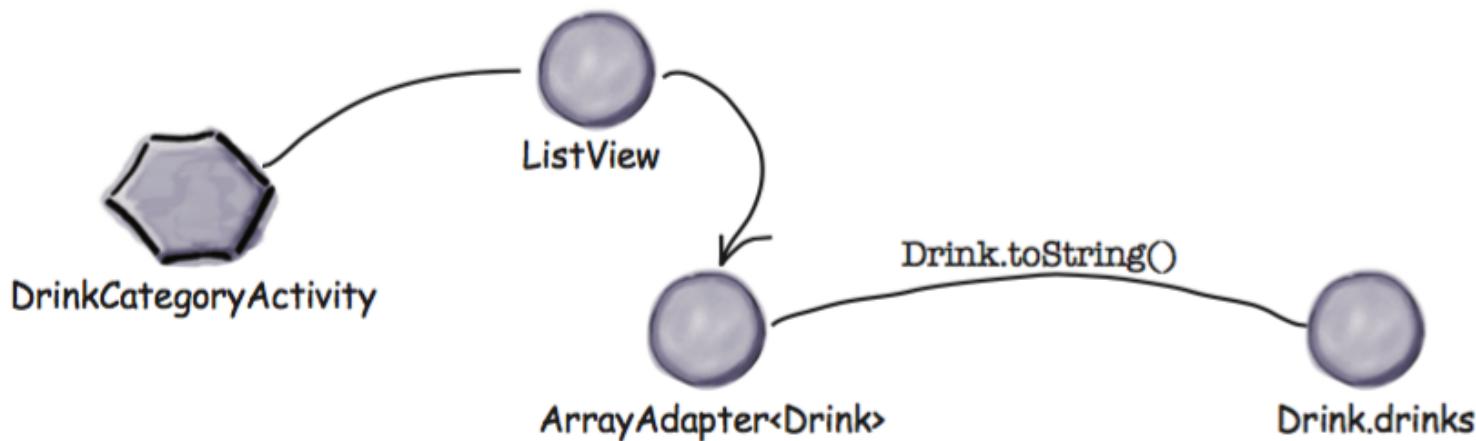
It uses the `Drink.toString()` method to return the name of each drink.



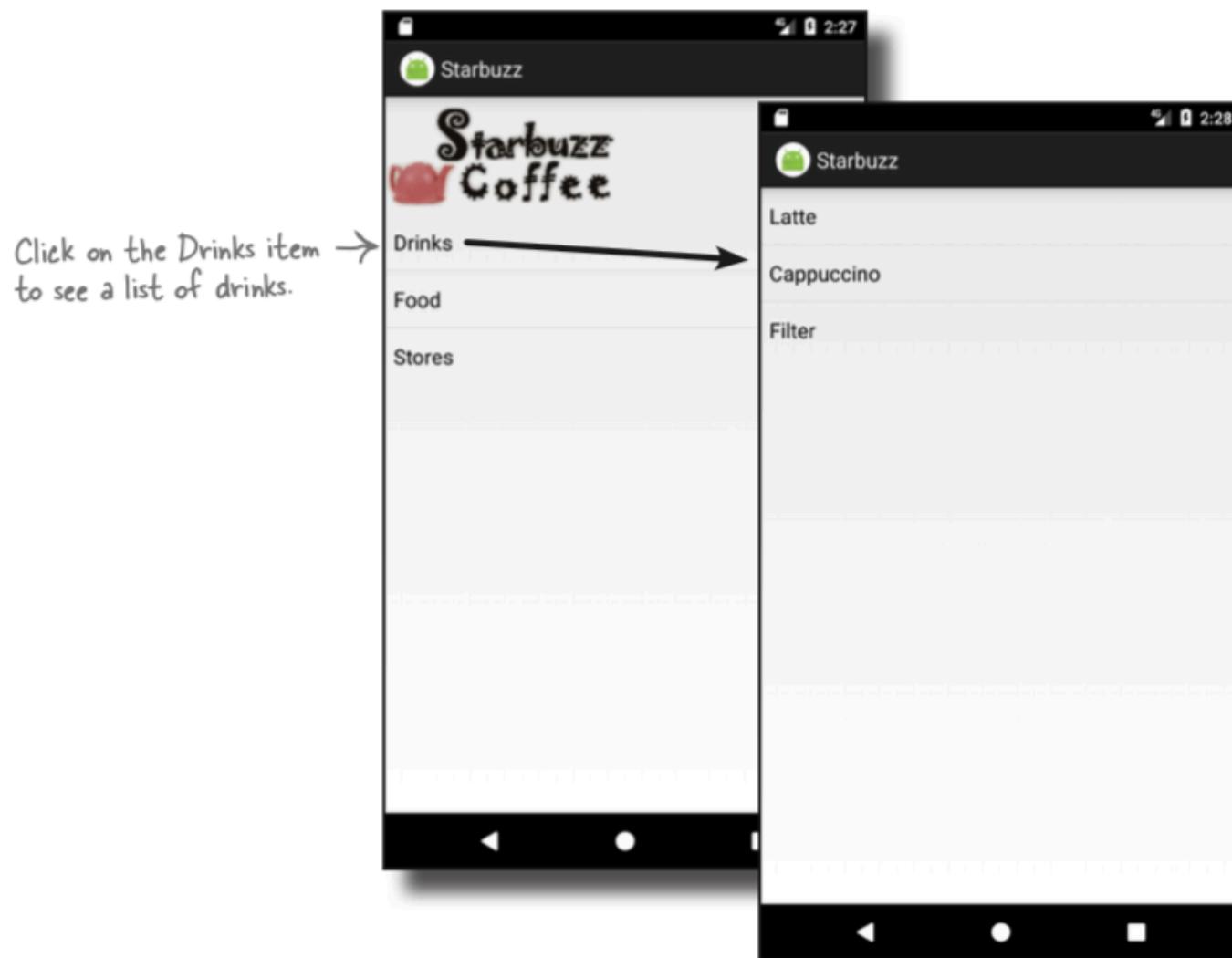
4

**DrinkCategoryActivity makes the ListView use the array adapter using the `setAdapter()` method.**

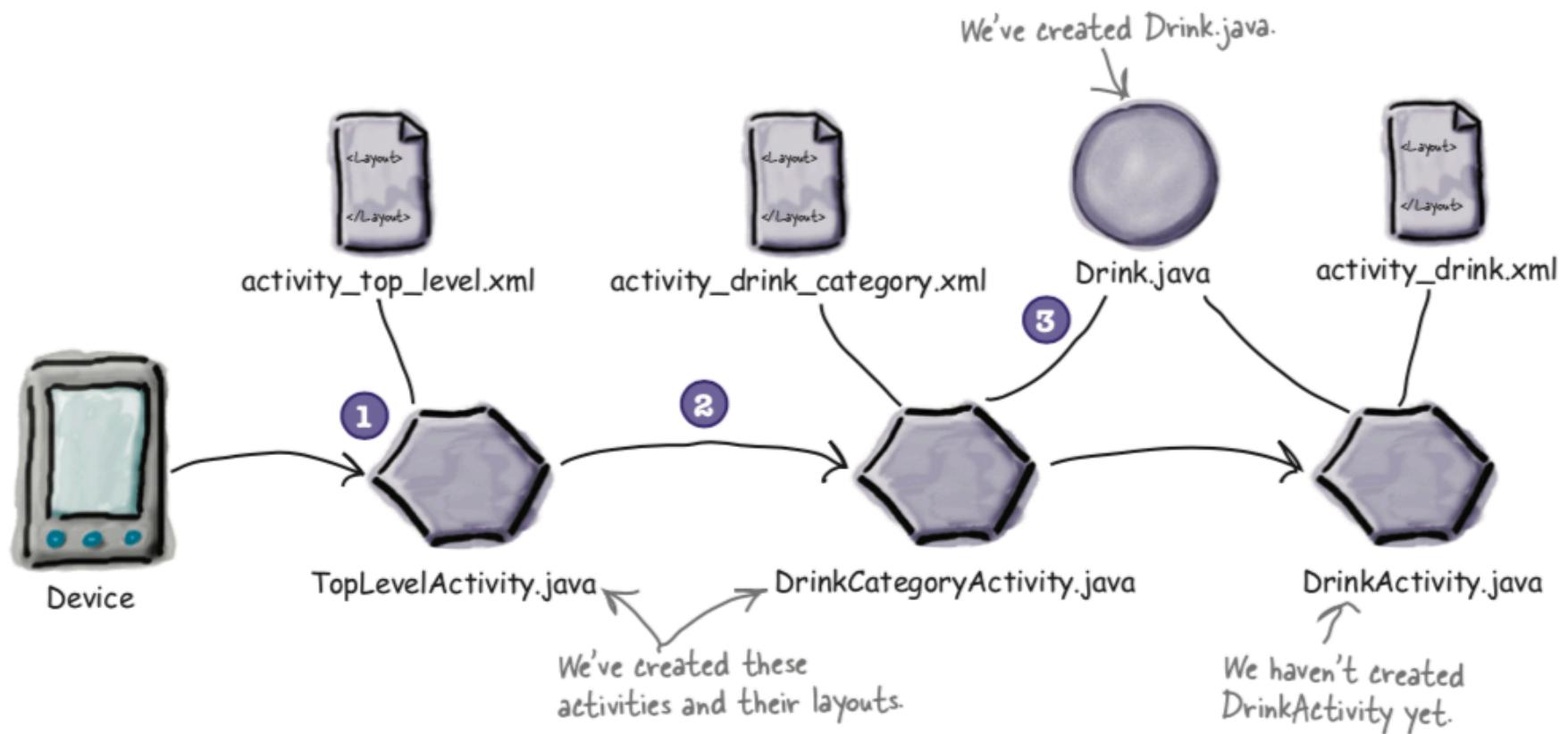
The list view uses it to display a list of the drink names.



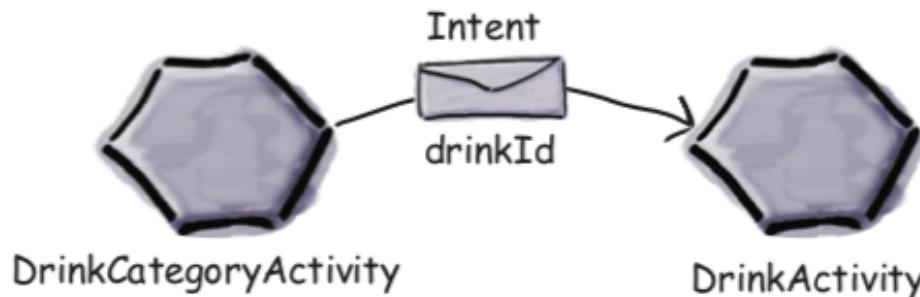
# Prueba



# Hacia donde vamos



# Pasando datos a una actividad utilizando el método onListItemClick()



DrinkCategoryActivity needs to start DrinkActivity.



```
Intent intent = new Intent(DrinkCategoryActivity.this, DrinkActivity.class);
intent.putExtra(DrinkActivity.EXTRA_DRINKID, (int) id); ← Add the ID of the item that
startActivity(intent);                                was clicked to the intent.
```



We're using a constant for the name of the extra information in the intent so that we know DrinkCategoryActivity and DrinkActivity are using the same String. We'll add this constant to DrinkActivity when we create the activity.

This is the index of the drink in the drinks array.

# El código completo de la clase DrinkCategoryActivity

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.view.View;
import android.content.Intent;
import android.widget.AdapterView;
```

We're using these extra classes  
so we need to import them.



# El código completo de la clase DrinkCategoryActivity

```
public class DrinkCategoryActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drink_category);
        ArrayAdapter<Drink> listAdapter = new ArrayAdapter<>(
            this,
            android.R.layout.simple_list_item_1,
            Drink.drinks);
        ListView listDrinks = (ListView) findViewById(R.id.list_drinks);
        listDrinks.setAdapter(listAdapter);

        //Create the listener
        AdapterView.OnItemClickListener itemClickListener =
            new AdapterView.OnItemClickListener() {
                public void onItemClick(AdapterView<?> listDrinks,
                    View itemView,
                    int position,
                    long id) {
                    This gets called when an item ↑
                    in the list view is clicked.

                    //Pass the drink the user clicks on to DrinkActivity
                    Intent intent = new Intent(DrinkCategoryActivity.this,
                        DrinkActivity.class); ←
                    intent.putExtra(DrinkActivity.EXTRA_DRINKID, (int) id);
                    startActivity(intent);
                }
            };
        //Assign the listener to the list view
        listDrinks.setOnItemClickListener(itemClickListener);
    }
}
```

Annotations and handwritten notes:

- Create a listener to listen for clicks.* (Annotation pointing to the `OnItemClickListener` declaration)
- This gets called when an item in the list view is clicked.* (Annotation pointing to the `onItemClick` method)
- When the user clicks on a drink, pass its ID to DrinkActivity and start it.* (Annotation pointing to the code block where `Intent` is created)
- We'll add DrinkActivity next, so don't worry if Android Studio says it doesn't exist.* (Annotation pointing to the `startActivity(intent)` call)

# Agregando la actividad DrinkActivity

To create the activity, select the `com.hfad.starbuzz` package in the `app/src/main/java` folder, then go to File→New...→Activity→Empty Activity. Name the activity “DrinkActivity”, name the layout “activity\_drink”, make sure the package name is `com.hfad.starbuzz`, and **uncheck the Backwards Compatibility (AppCompat) checkbox**. Then replace the contents of `activity_drink.xml` with this:

Make sure you create the new activity.

If prompted for the activity's source language, select the option for Java.

# Una actividad de detalle muestra datos de un sólo registro

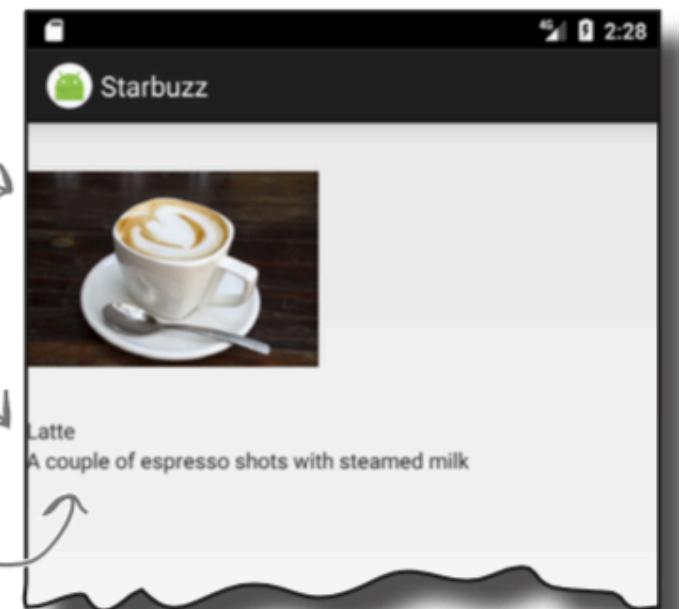
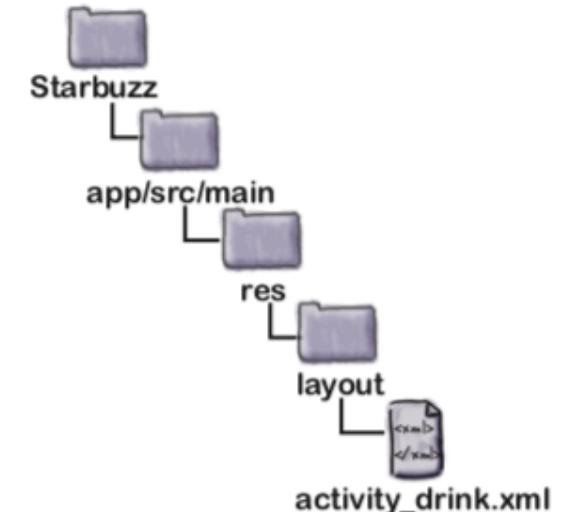
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.hfad.starbuzz.DrinkActivity" >

    <ImageView
        android:id="@+id/photo"
        android:layout_width="190dp"
        android:layout_height="190dp" />

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```



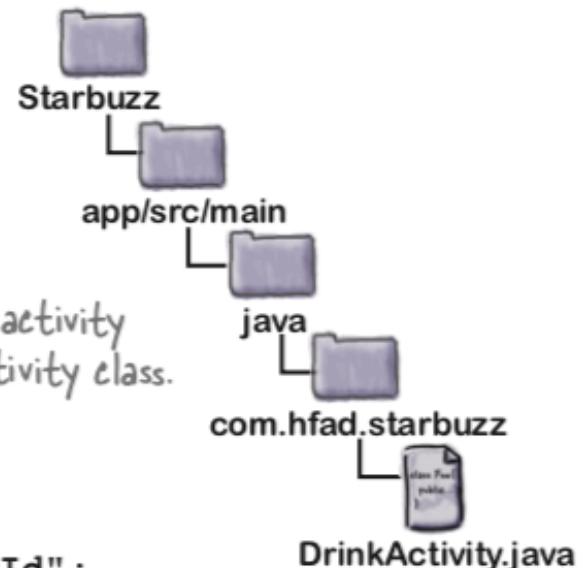
# El código de la clase DrinkActivity

```
package com.hfad.starbuzz;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
public class DrinkActivity extends Activity {  
  
    public static final String EXTRA_DRINKID = "drinkId";  
}
```

We're using these classes so we need to import them.

Make sure your activity extends the Activity class.

Add EXTRA\_DRINKID as a constant.

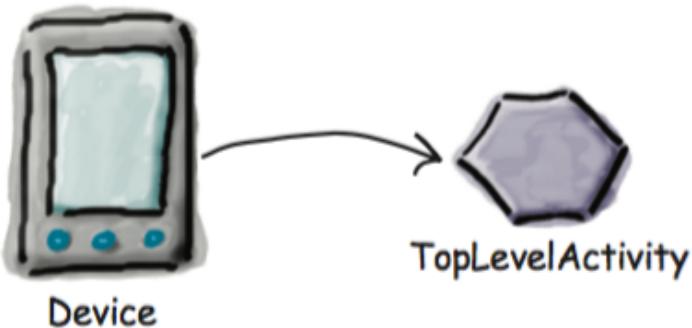


```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_drink);  
  
    //Get the drink from the intent  
    int drinkId = (Integer) getIntent().getExtras().get(EXTRA_DRINKID);  
    Drink drink = Drink.drinks[drinkId]; ← Use the drinkId to get details of  
                                              the drink the user chose.  
    //Populate the drink name  
    TextView name = (TextView) findViewById(R.id.name);  
    name.setText(drink.getName());  
  
    //Populate the drink description  
    TextView description = (TextView) findViewById(R.id.description);  
    description.setText(drink.getDescription());  
  
    //Populate the drink image  
    ImageView photo = (ImageView) findViewById(R.id.photo);  
    photo.setImageResource(drink.getImageResourceId());  
    photo.setContentDescription(drink.getName());  
}  
}
```

El código de la clase DrinkActivity

# Que pasa al ejecutar la aplicación

- 1 When the user starts the app, it launches TopLevelActivity.



- 2 The `onCreate()` method in `TopLevelActivity` creates an `onItemClickListener` and links it to the activity's `ListView`.

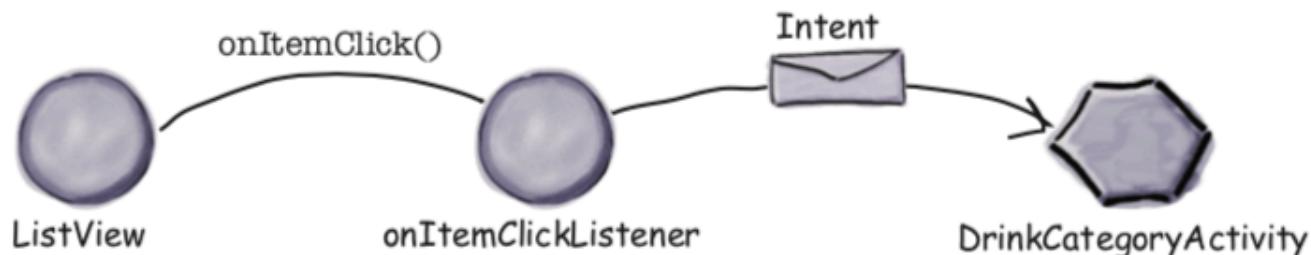


# Que pasa al ejecutar la aplicación

3

**When the user clicks on an item in the ListView, the onItemClickListener's onItemClick() method gets called.**

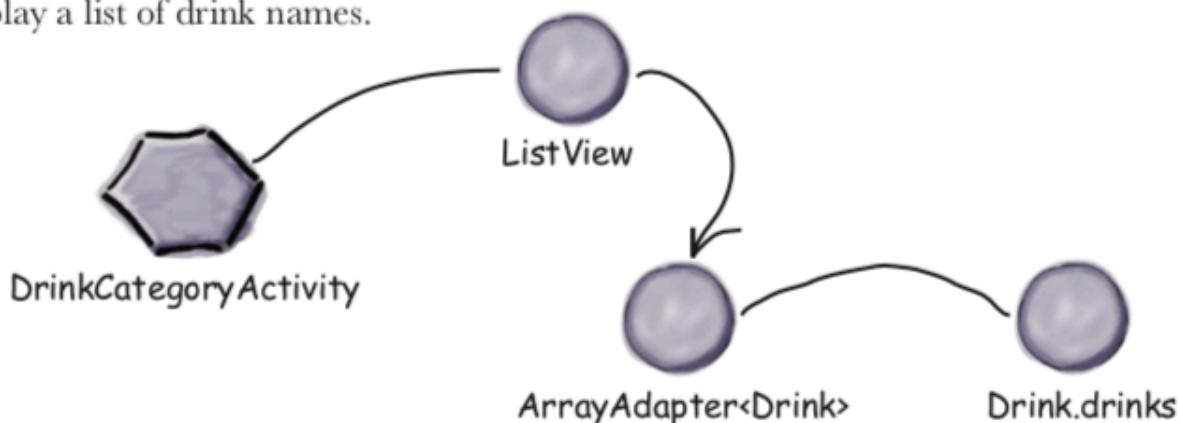
If the Drinks item was clicked, the onItemClickListener creates an intent to start DrinkCategoryActivity.



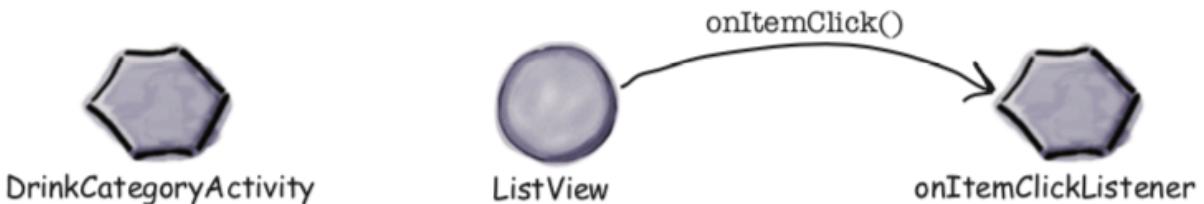
4

**DrinkCategoryActivity displays a single ListView.**

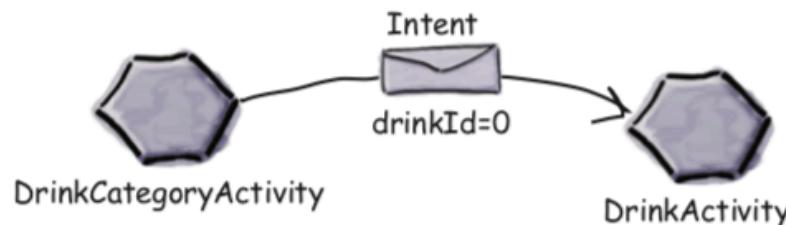
The DrinkCategoryActivity list view uses an `ArrayAdapter<Drink>` to display a list of drink names.



- 5 When the user chooses a drink from DrinkCategoryActivity's ListView, onItemClickListener's onItemClick() method gets called.

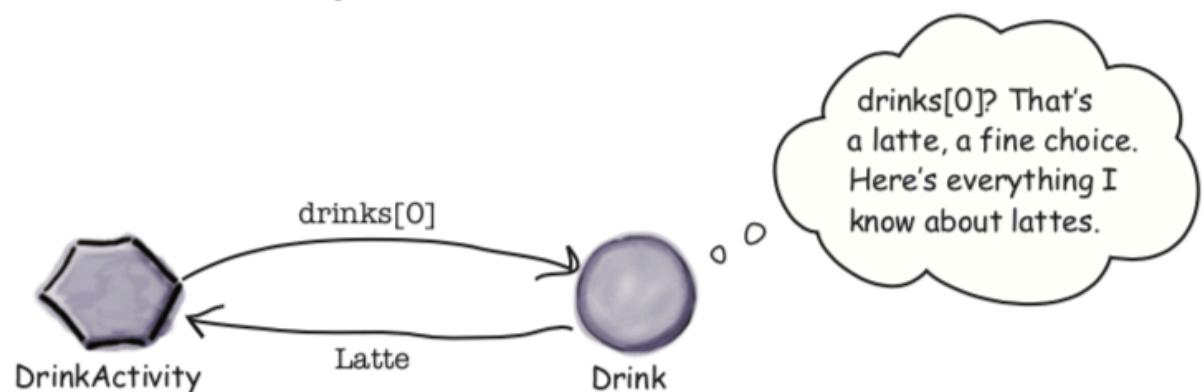


- 6 The onItemClick() method creates an intent to start DrinkActivity, passing along the drink ID as extra information.



- 7 **DrinkActivity launches.**

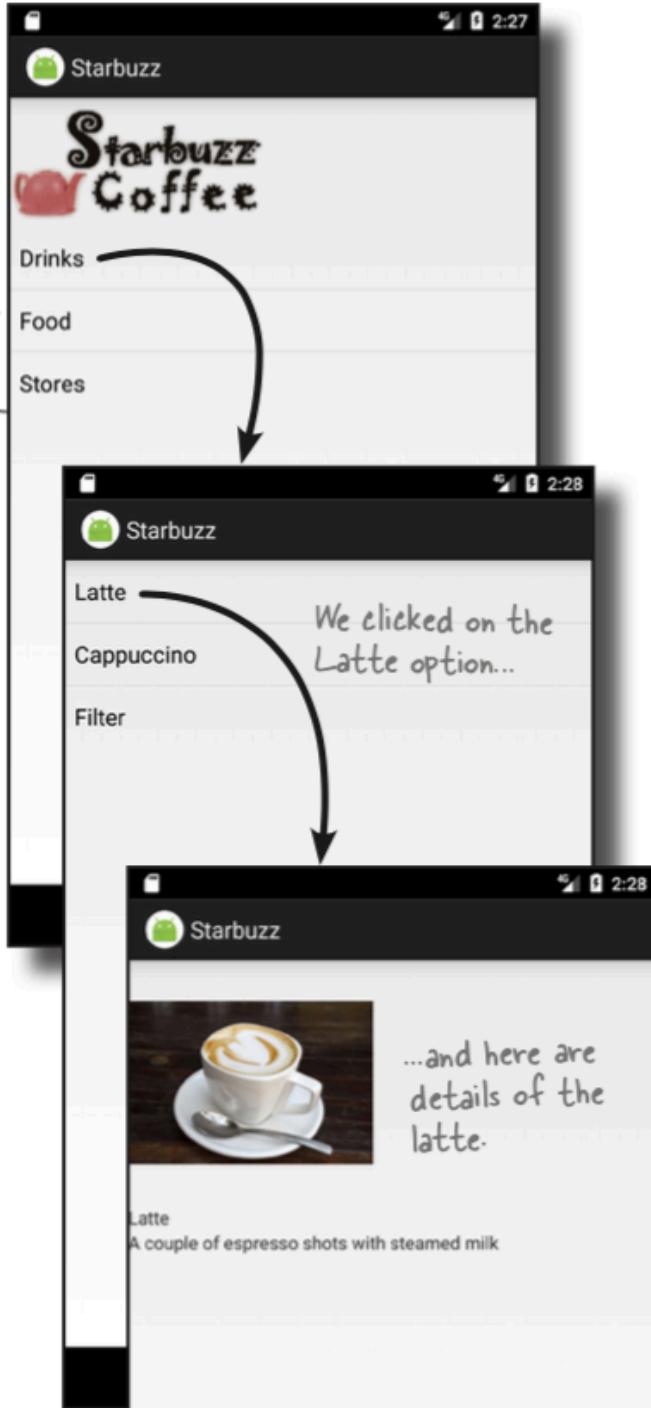
It retrieves the drink ID from the intent, and gets details for the correct drink from the Drink class. It uses this information to update its views.



Que pasa al ejecutar la aplicación

# Prueba

We've implemented the Drinks part  
of the app. The other items won't  
do anything if you click on them.





## BULLET POINTS

- Sort your ideas for activities into top-level activities, category activities, and detail/edit activities. Use the category activities to navigate from the top-level activities to the detail/edit activities.
- A list view displays items in a list. Add it to your layout using the `<ListView>` element.
- Use `android:entries` in your layout to populate the items in your list views from an array defined in `strings.xml`.
- An adapter acts as a bridge between an `AdapterView` and a data source. `ListsViews` and `Spinners` are both types of `AdapterView`.
- An `ArrayAdapter` is an adapter that works with arrays.
- Handle click events on Buttons using `android:onClick` in the layout code. Handle click events elsewhere by creating a listener and implementing its click event.