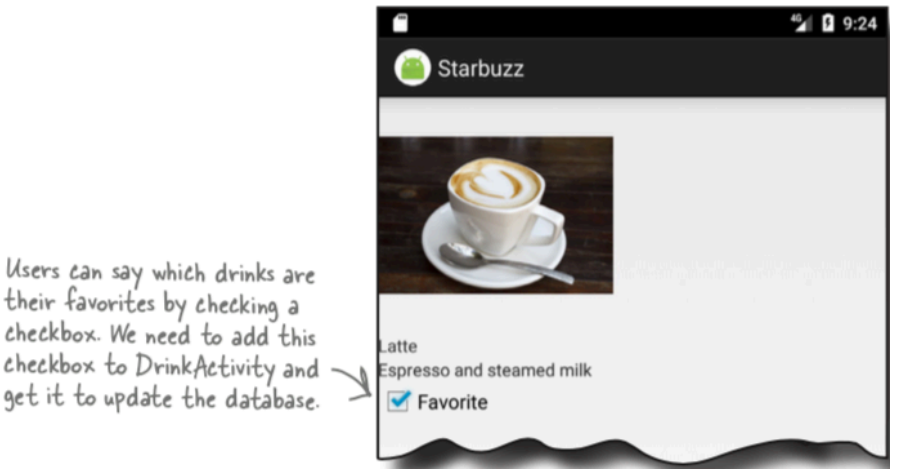
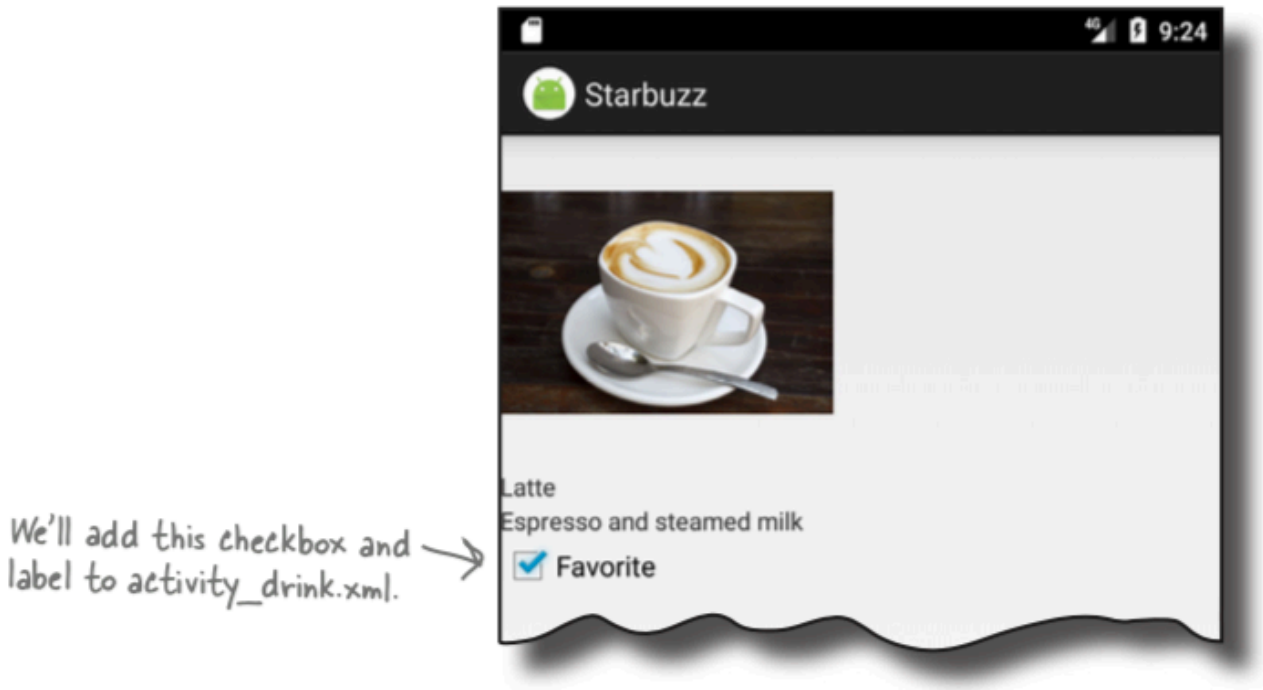
10
Cursores y tareas asíncronas

Actualizando la base de datos



Agregando favoritos a DrinkActivity

1. **Update DrinkActivity's layout to add a checkbox and text label.**



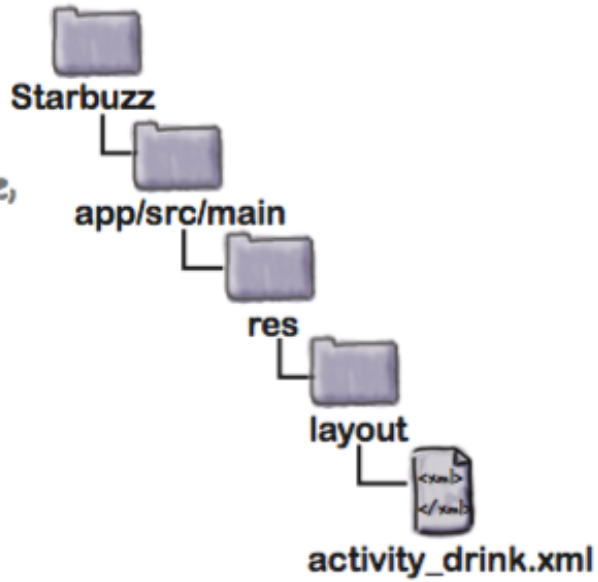
2. **Display the value of the FAVORITE column in the checkbox.**
To do this, we'll need to retrieve the value of the FAVORITE column from the Starbuzz database.

3. **Update the FAVORITE column when the checkbox is clicked.**
We'll update the FAVORITE column with the value of the checkbox so that the data in the database stays up to date.

Agregando favoritos a DrinkActivity

```
<LinearLayout ...>
    <ImageView android:id="@+id/photo"
        ... />

    <TextView android:id="@+id/name"
        ... />

    <TextView android:id="@+id/description"
        ... />

    <CheckBox android:id="@+id/favorite"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/favorite"
        android:onClick="onFavoriteClicked"/>
</LinearLayout>
```

These are the photo, name, and description views we added when we first created the activity.



The checkbox has an ID of favorite.

We need to give the checkbox a label.

When the checkbox is clicked, the onFavoriteClicked() method will get called.

```
protected void onCreate(Bundle savedInstanceState) {
    ...
    SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
    SQLiteDatabase db = starbuzzDatabaseHelper.getReadableDatabase();
    Cursor cursor = db.query ("DRINK",
                    new String[] {"NAME", "DESCRIPTION", "IMAGE_RESOURCE_ID", "FAVORITE"},
                    "_id = ?",
                    new String[] {Integer.toString(drinkNo)},
                    null, null,null);

    //Move to the first record in the Cursor
    if (cursor.moveToFirst()) {
        //Get the drink details from the cursor
        String nameText = cursor.getString(0);
        String descriptionText = cursor.getString(1);
        int photoId = cursor.getInt(2);
        boolean isFavorite = (cursor.getInt(3) == 1);
    ...
        //Populate the favorite checkbox
        CheckBox favorite = (CheckBox)findViewById(R.id.favorite);
        favorite.setChecked(isFavorite);
    ...
}
```

Add the FAVORITE column to the cursor.

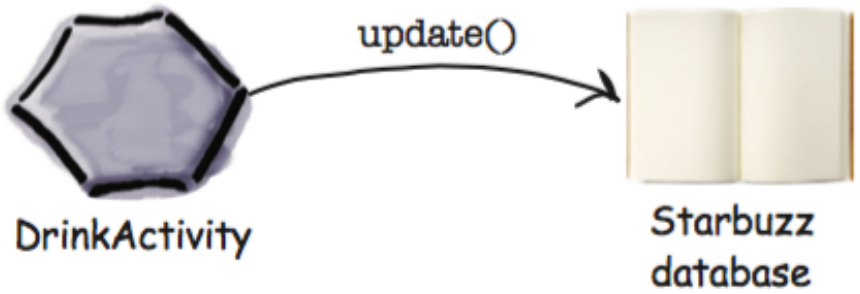
Get the value of the FAVORITE column. It's stored in the database as 1 for true, 0 for false.

Set the value of the favorite checkbox.

Agregando una nueva columna al cursor

Respondiendo a clicks para actualizar la base de datos

```
database.update(String table,
                ContentValues values,
                String whereClause,
                String[] whereArgs);
```



El código de DrinkActivity


```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.view.View;
import android.widget.CheckBox;
import android.content.ContentValues;

public class DrinkActivity extends Activity {

    public static final String EXTRA_DRINKID = "drinkId";

    @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);
```

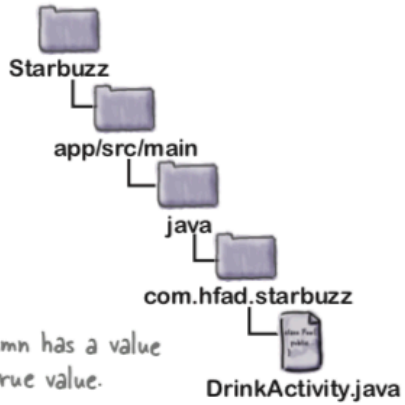
We're using these extra classes, so you need to import them.

```
//Create a cursor
SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
try {
    SQLiteDatabase db = starbuzzDatabaseHelper.getReadableDatabase();
    Cursor cursor = db.query("DRINK",
            new String[]{"NAME", "DESCRIPTION", "IMAGE_RESOURCE_ID", "FAVORITE"},
            "_id = ?",
            new String[]{Integer.toString(drinkId)},
            null, null, null);

    //Move to the first record in the Cursor
    if (cursor.moveToFirst()) {
        //Get the drink details from the cursor
        String nameText = cursor.getString(0);
        String descriptionText = cursor.getString(1);
        int photoId = cursor.getInt(2);
        boolean isFavorite = (cursor.getInt(3) == 1);

        //Populate the drink name
        TextView name = (TextView) findViewById(R.id.name);
        name.setText(nameText);

        //Populate the drink description
        TextView description = (TextView) findViewById(R.id.description);
        description.setText(descriptionText);

        //Populate the drink image
        ImageView photo = (ImageView) findViewById(R.id.photo);
        photo.setImageResource(photoId);
        photo.setContentDescription(nameText);

        //Populate the favorite checkbox
        CheckBox favorite = (CheckBox)findViewById(R.id.favorite);
        favorite.setChecked(isFavorite);
    }
```

Add the FAVORITE column to the cursor.

If the FAVORITE column has a value of 1, this indicates a true value.

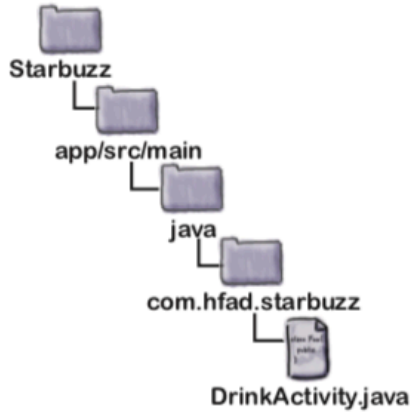
If the drink is a favorite, put a checkmark in the favorite checkbox.



El código de DrinkActivity

El código de DrinkActivity

```
        cursor.close();
        db.close();
    } catch (SQLiteException e) {
        Toast toast = Toast.makeText(this,
                "Database unavailable",
                Toast.LENGTH_SHORT);
        toast.show();
    }
}
```




```
//Update the database when the checkbox is clicked
public void onFavoriteClicked(View view){
    int drinkId = (Integer) getIntent().getExtras().get(EXTRA_DRINKID);

    //Get the value of the checkbox
    CheckBox favorite = (CheckBox) findViewById(R.id.favorite);
    ContentValues drinkValues = new ContentValues();
    drinkValues.put("FAVORITE", favorite.isChecked());

    //Get a reference to the database and update the FAVORITE column
    SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
    try {
        SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
        db.update("DRINK",
                  drinkValues,
                  "_id = ?",
                  new String[] {Integer.toString(drinkId)});
        db.close();
    } catch(SQLiteException e) {
        Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
        toast.show();
    }
}
}
```

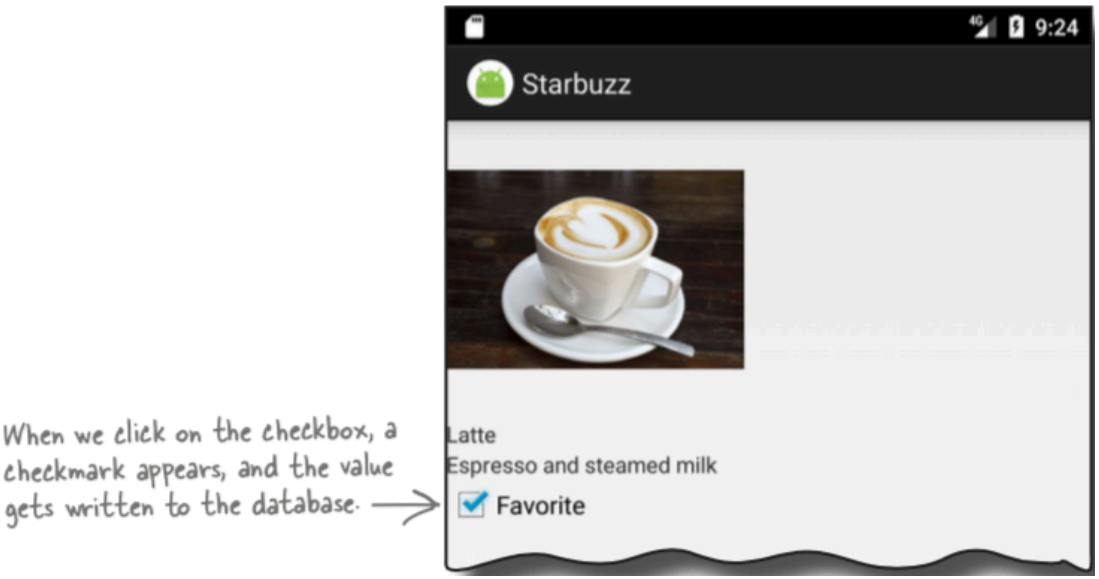
Add the value of the favorite checkbox to the drinkValues ContentValues object.

Update the drink's FAVORITE column in the database to the value of the checkbox.

Display a message if there's a problem with the database.

Prueba


9:23
Starbuzz
Latte
Espresso and steamed milk
Favorite
Here's the new checkbox we added with its label.

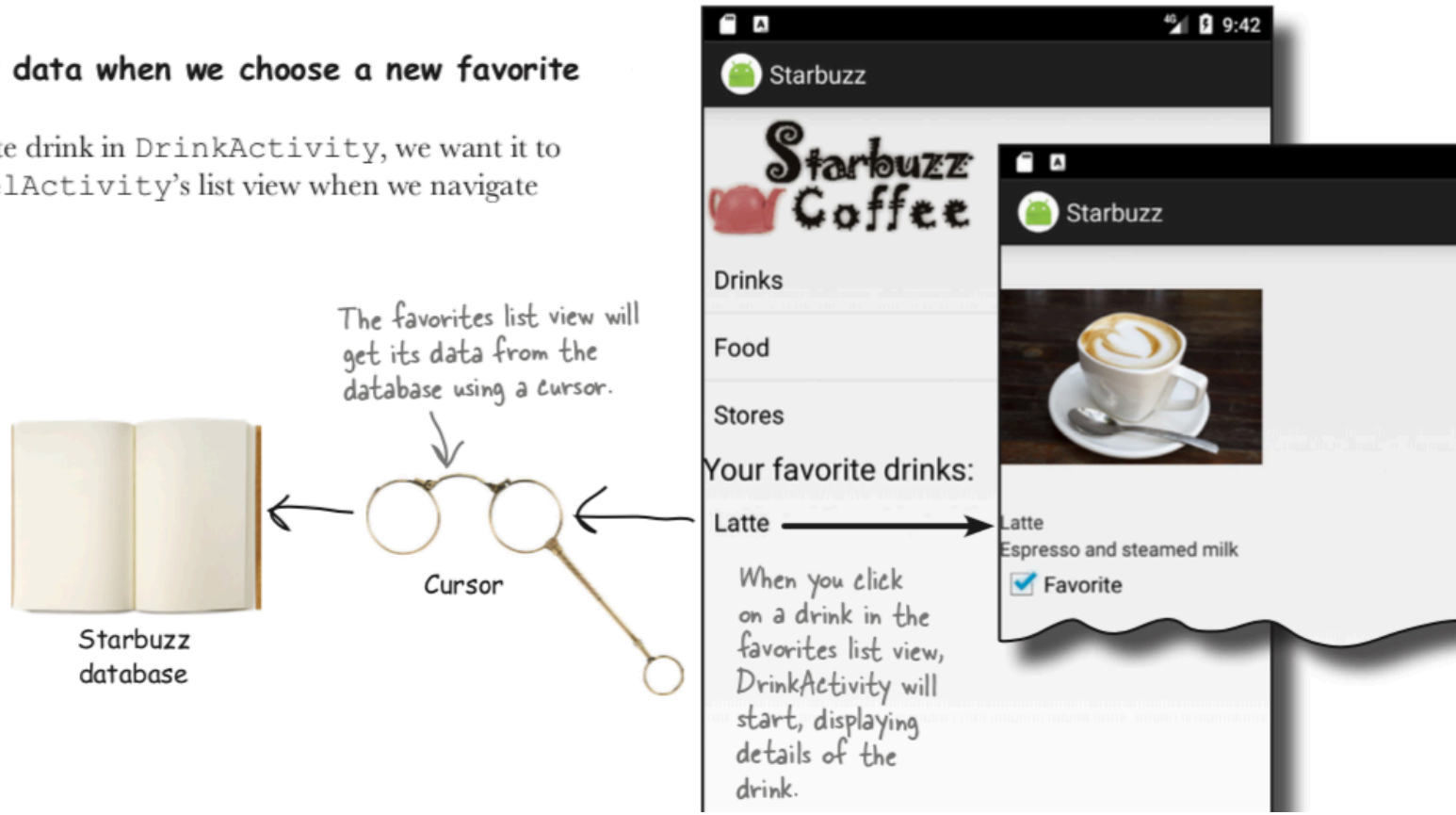

9:24
Starbuzz
Latte
Espresso and steamed milk
Favorite
When we click on the checkbox, a checkmark appears, and the value gets written to the database.

Mostrando los favoritos en TopLevelActivity

1. **Add a list view and text view to TopLevelActivity's layout.**

2. **Populate the list view and get it to respond to clicks.**
We'll create a new cursor that retrieves the user's favorite drinks from the database, and attach it to the list view using a cursor adapter. We'll then create an `onItemClickListener` so that we can get `TopLevelActivity` to start `DrinkActivity` when the user clicks on one of the drinks.

3. **Refresh the list view data when we choose a new favorite drink.**
If we choose a new favorite drink in `DrinkActivity`, we want it to be displayed in `TopLevelActivity`'s list view when we navigate back to it.



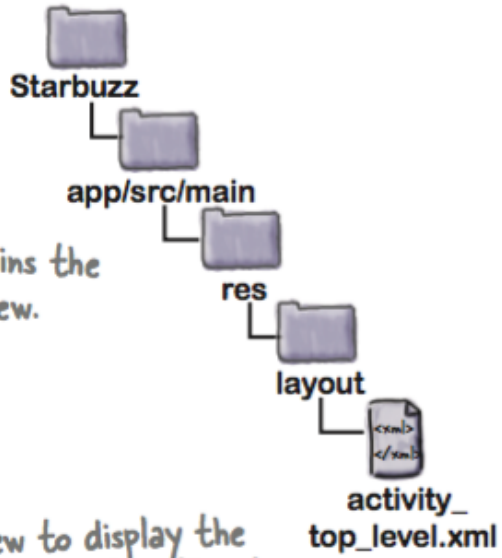
Mostrando las bebidas favoritas en activity_top_level.xml

```
<LinearLayout ... >
    <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" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/favorites" />

    <ListView
        android:id="@+id/list_favorites"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

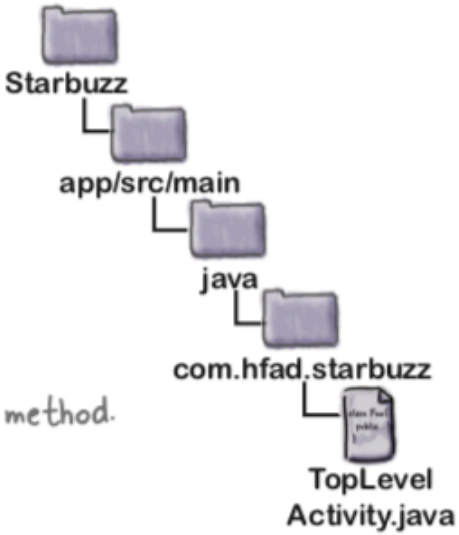
The layout already contains the Starbuzz logo and list view.

We'll add a text view to display the text "Your favorite drinks". We'll put this in a string called favorites.

The list_favorites ListView will display the user's favorite drinks.



Refactorizando TopLevelActivity




```
package com.hfad.starbuzz;
...
public class TopLevelActivity extends Activity {

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

    private void setupOptionsListView() {
        //Create an OnItemClickListener
        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);
                }
            }
        };
        //Add the listener to the list view
        ListView listView = (ListView) findViewById(R.id.list_options);
        listView.setOnItemClickListener(itemClickListener);
    }
}
```

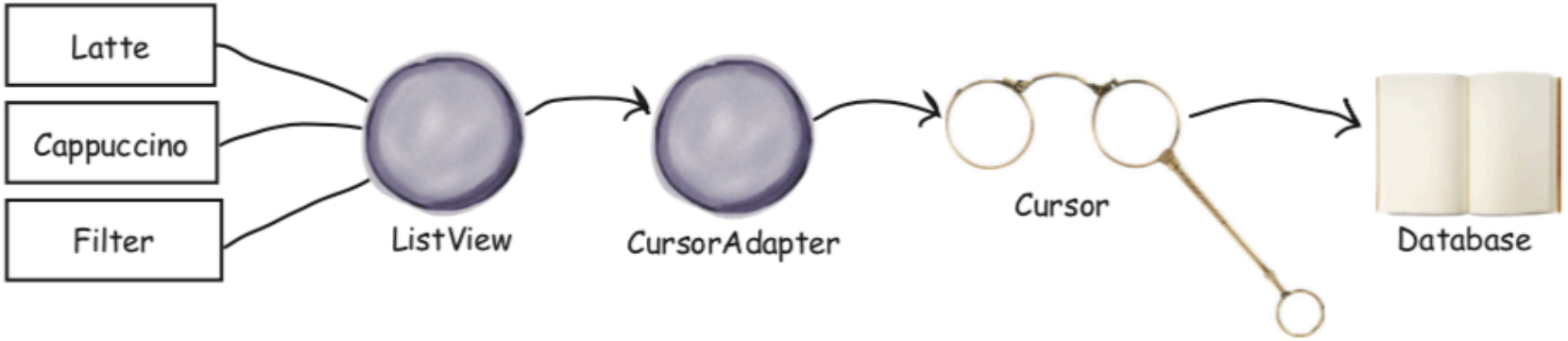
Call the new setupOptionsListView() method.

All of this code was in the onCreate() method. We're putting it in a new method to make the code tidier.

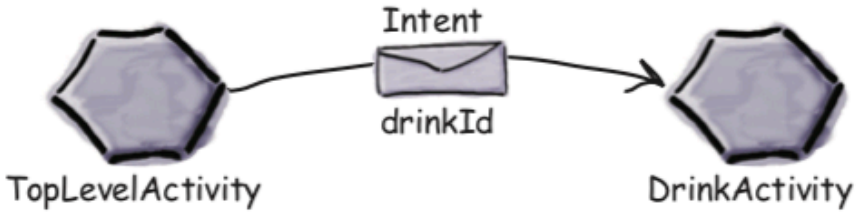
If the Drink option in the list_options list view is clicked, start DrinkCategoryActivity.

Cambios en TopLevelActivity

1 **Populate the list_favorites list view using a cursor.**
The cursor will return all drinks where the FAVORITE column has been set to `1`—all drinks that the user has flagged as being a favorite. Just as we did in our code for `DrinkCategoryActivity`, we can connect the cursor to the list view using a cursor adapter.



2 **Create an onItemClickListener so that the list_favorites list view can respond to clicks.**
If the user clicks on one of their favorite drinks, we can create an intent that starts `DrinkActivity`, passing it the ID of the drink that was clicked. This will show the user details of the drink they've just chosen.



```
private void setupFavoritesListView() {
    //Populate the list_favorites ListView from a cursor
    ListView listFavorites = (ListView) findViewById(R.id.list_favorites);
    try{
        SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
        db = starbuzzDatabaseHelper.getReadableDatabase();
        favoritesCursor = db.query("DRINK",
                new String[] { "_id", "NAME"},
                "FAVORITE = 1",
                null, null, null, null);
        CursorAdapter favoriteAdapter =
                new SimpleCursorAdapter(TopLevelActivity.this,
                        android.R.layout.simple_list_item_1,
                        favoritesCursor,
                        new String[]{"NAME"},
                        new int[]{android.R.id.text1}, 0);
        listFavorites.setAdapter(favoriteAdapter);
    } catch(SQLiteException e) {
        Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
        toast.show();
    }

    //Navigate to DrinkActivity if a drink is clicked
    listFavorites.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> listView, View v, int position, long id) {
            Intent intent = new Intent(TopLevelActivity.this, DrinkActivity.class);
            intent.putExtra(DrinkActivity.EXTRA_DRINKID, (int)id);
            startActivity(intent);
        }
    });
}
```

Get the list_favorites list view.

Create a cursor that gets the values of the _id and NAME columns where FAVORITE=1.

Get the names of the user's favorite drinks.

Create a new cursor adapter.

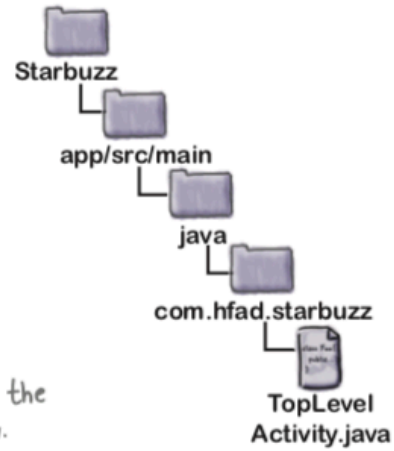
Use the cursor in the cursor adapter.

Display the names of the drinks in the list view.

Display a message if there's a problem with the database.

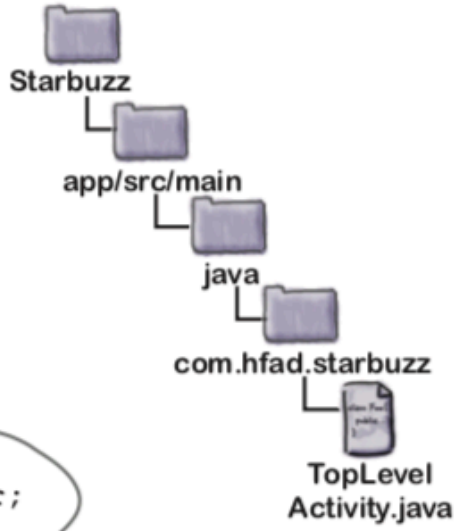
This will get called if an item in the list view is clicked.

If the user clicks on one of the items in the list_favorites list view, create an intent to start DrinkActivity, including the ID of the drink as extra information.



El método setupFavoritesListView()

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.widget.AdapterView;
import android.widget.ListView;
import android.view.View;
import android.database.Cursor;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteDatabase;
import android.widget.SimpleCursorAdapter;
import android.widget.CursorAdapter;
import android.widget.Toast;
```



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

```
public class TopLevelActivity extends Activity {

    private SQLiteDatabase db;
    private Cursor favoritesCursor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_top_level);
        setupOptionsListView();
        setupFavoritesListView();
    }
```

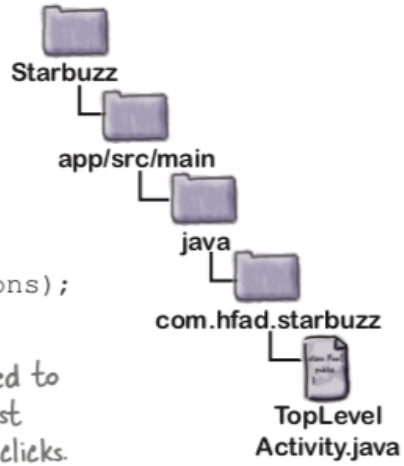
We're adding the database and cursor as private variables so that we can access them in the setUpFavoritesListView() and onDestroy() methods.

Call the setupFavoritesListView() method from the onCreate() method.

El nuevo código de la actividad de nivel más alto

We don't need to change this method.

```
private void setupOptionsListView() {
    //Create an OnItemClickListener
    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);
            }
        }
    };

    //Add the listener to the list view
    ListView listView = (ListView) findViewById(R.id.list_options);
    listView.setOnItemClickListener(itemClickListener);
}
```



This is the method we created to populate the list_favorites list view and make it respond to clicks.

```
private void setupFavoritesListView() {
    //Populate the list_favorites ListView from a cursor
    ListView listFavorites = (ListView) findViewById(R.id.list_favorites);
    try{
        SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
        db = starbuzzDatabaseHelper.getReadableDatabase();
        favoritesCursor = db.query("DRINK",
                new String[] { "_id", "NAME"},
                "FAVORITE = 1",
                null, null, null, null);
```

Get a reference to the database.

The list_favorites list view will use this cursor for its data.

El nuevo código de la actividad de nivel más alto

El nuevo código de la actividad de nivel más alto

```
        CursorAdapter favoriteAdapter =
                new SimpleCursorAdapter(TopLevelActivity.this,
                        android.R.layout.simple_list_item_1,
                        favoritesCursor,
                        new String[]{"NAME"},
                        new int[]{android.R.id.text1}, 0);
        listFavorites.setAdapter(favoriteAdapter);
    } catch(SQLiteException e) {
        Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
        toast.show();
    }

    //Navigate to DrinkActivity if a drink is clicked
    listFavorites.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> listView, View v, int position, long id) {
            Intent intent = new Intent(TopLevelActivity.this, DrinkActivity.class);
            intent.putExtra(DrinkActivity.EXTRA_DRINKID, (int)id);
            startActivity(intent);
        }
    });
}

//Close the cursor and database in the onDestroy() method
@Override
public void onDestroy(){
    super.onDestroy();
    favoritesCursor.close();
    db.close();
}
}
```

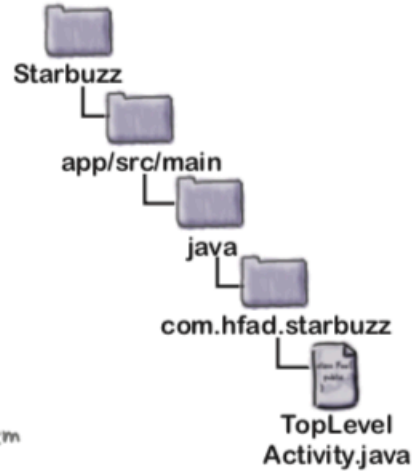
Use the cursor in a cursor adapter.

Set the curser adapter to the list view.

Get the list_favorites list view to respond to clicks.

Start DrinkActivity, passing it the ID of the drink that was clicked on.

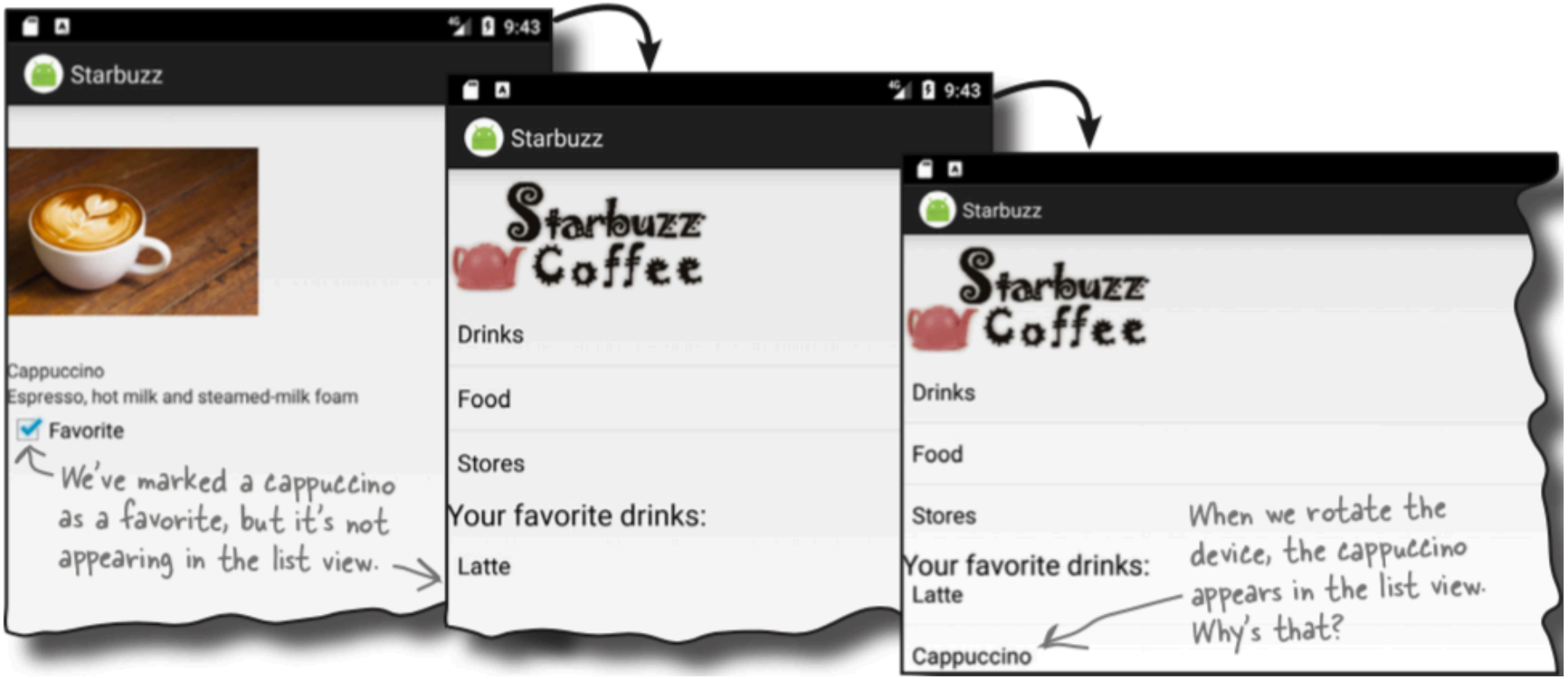
The onDestroy() method gets called just before the activity is destroyed. We'll close the cursor and database in this method, as we no longer need them if the activity's being destroyed.



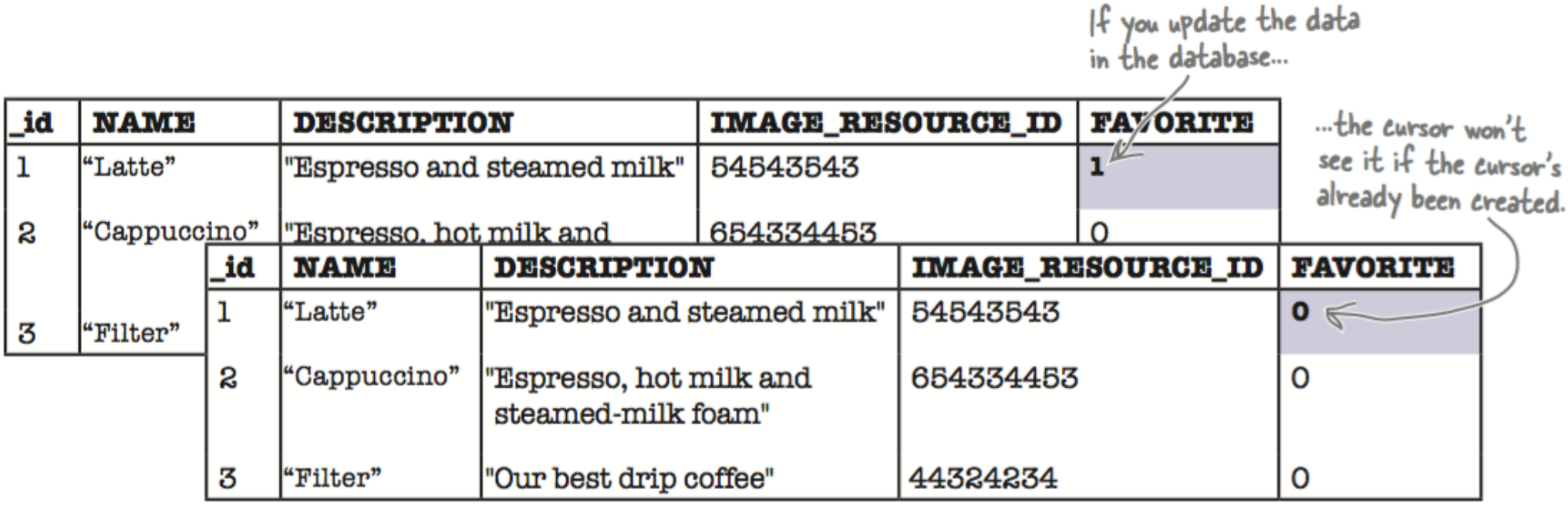
Prueba


Here's the new list_favorites list view we created. It displays a latte, as we marked this as a favorite drink earlier in the chapter.
9:42
Starbuzz
Starbuzz Coffee
Drinks
Food
Stores
Your favorite drinks:
Latte
9:42
Starbuzz
Latte
Espresso and steamed milk
Favorite
When we click on the drink, its details are displayed.

El cursor no se refresca automáticamente



El cursor no se refresca automáticamente



_id	NAME	DESCRIPTION	IMAGE_RESOURCE_ID	FAVORITE
1	"Latte"	"Espresso and steamed milk"	54543543	1
2	"Cappuccino"	"Espresso, hot milk and	654334453	0
3	"Filter"			

_id	NAME	DESCRIPTION	IMAGE_RESOURCE_ID	FAVORITE
1	"Latte"	"Espresso and steamed milk"	54543543	0
2	"Cappuccino"	"Espresso, hot milk and steamed-milk foam"	654334453	0
3	"Filter"	"Our best drip coffee"	44324234	0

Cambiando el cursor con changeCursor()

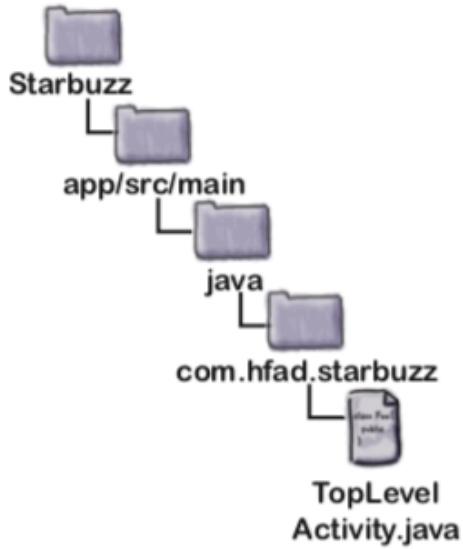
```
//Create the new cursor
StarbuzzDatabaseHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
SQLiteDatabase db = starbuzzDatabaseHelper.getReadableDatabase();
Cursor cursor = db.query("DRINK",
                new String[] { "_id", "NAME"},
                "FAVORITE = 1",
                null, null, null, null);

//Get the CursorAdapter used by the ListView
ListView listFavorites = (ListView)findViewById(R.id.list_favorites);
CursorAdapter adapter = (CursorAdapter) listFavorites.getAdapter();

//Change the cursor used by the CursorAdapter to the new one we just created
adapter.changeCursor(cursor);
```

You create a new cursor in exactly the same way you did before.

You get the ListView's adapter using the getAdapter() method.

Change the cursor used by the cursor adapter to the new one.

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.widget.AdapterView;
import android.widget.ListView;
import android.view.View;
import android.database.Cursor;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteDatabase;
import android.widget.SimpleCursorAdapter;
import android.widget.CursorAdapter;
import android.widget.Toast;

public class TopLevelActivity extends Activity {

    private SQLiteDatabase db;
    private Cursor favoritesCursor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_top_level);
        setupOptionsListView();
        setupFavoritesListView();
    }
```

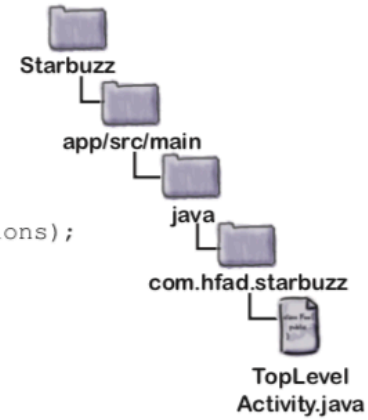


You don't need to change any of the code on this page.

El código revisado de TopLevelActivity

```
private void setupOptionsListView() {
    //Create an OnItemClickListener
    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);
            }
        }
    };

    //Add the listener to the list view
    ListView listView = (ListView) findViewById(R.id.list_options);
    listView.setOnItemClickListener(itemClickListener);
}

private void setupFavoritesListView() {
    //Populate the list_favorites ListView from a cursor
    ListView listFavorites = (ListView) findViewById(R.id.list_favorites);
    try{
        SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
        db = starbuzzDatabaseHelper.getReadableDatabase();
        favoritesCursor = db.query("DRINK",
                                   new String[] { "_id", "NAME"},
                                   "FAVORITE = 1",
                                   null, null, null, null);

        CursorAdapter favoriteAdapter =
                new SimpleCursorAdapter(TopLevelActivity.this,
                        android.R.layout.simple_list_item_1,
                        favoritesCursor,
                        new String[]{"NAME"},
                        new int[]{android.R.id.text1}, 0);
        listFavorites.setAdapter(favoriteAdapter);
```

You don't need to change any of the code on this page.



El código revisado de TopLevelActivity

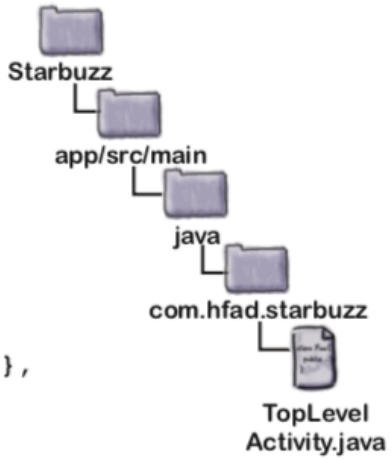
```
        } catch(SQLiteException e) {
            Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
            toast.show();
        }

        //Navigate to DrinkActivity if a drink is clicked
        listFavorites.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> listView, View v, int position, long id) {
                Intent intent = new Intent(TopLevelActivity.this, DrinkActivity.class);
                intent.putExtra(DrinkActivity.EXTRA_DRINKID, (int)id);
                startActivity(intent);
            }
        });
    }

    @Override
    public void onRestart() {
        super.onRestart();
        Cursor newCursor = db.query("DRINK",
                                    new String[] { "_id", "NAME"},
                                    "FAVORITE = 1",
                                    null, null, null, null);
        ListView listFavorites = (ListView) findViewById(R.id.list_favorites);
        CursorAdapter adapter = (CursorAdapter) listFavorites.getAdapter();
        adapter.changeCursor(newCursor);
        favoritesCursor = newCursor;
    }
    //Close the cursor and database in the onDestroy() method
    @Override
    public void onDestroy(){
        super.onDestroy();
        favoritesCursor.close();
        db.close();
    }
}
```

Add the onRestart() method. This will get called when the user navigates back to TopLevelActivity.

Create a new version of the cursor.

Switch the cursor being used by the list_favorites list view to the new cursor.

Change the value of favoritesCursor to the new cursor so we can close it in the activity's onDestroy() method.



El código revisado de TopLevelActivity

Prueba


10:13
Starbuzz
Starbuzz Coffee
Drinks
Food
Stores
Your favorite drinks:
Latte
Cappuccino
The list_favorites ListView initially contains a latte and a cappuccino.
10:13
Starbuzz
Latte
Espresso and steamed milk
Favorite
When we click on a latte, its details are displayed.
10:14
Starbuzz
Latte
Espresso and steamed milk
Favorite
We then uncheck the checkbox to indicate the drink is no longer a favorite.
10:14
Starbuzz
Starbuzz Coffee
Drinks
Food
Stores
Your favorite drinks:
Cappuccino
When we return to TopLevelActivity, the latte is no longerlisted in the list_favorites list view.

Utiizando hilos (threads) para acelerar el funcionamiento de la base de datos

- **The main event thread**
  This is the real workhorse in Android. It listens for intents, it receives touch messages from the screen, and it calls all of the methods inside your activities.
- **The render thread**
  You don't normally interact with this thread, but it reads a list of requests for screen updates and then calls the low-level graphics hardware to repaint the screen and make your app look pretty.
- **All of the other thread that you create**

Que código va en cada hilo?

```
//Update the database when the checkbox is clicked
public void onFavoriteClicked(View view){
```

```
int drinkNo = (Integer)getIntent().getExtras().get(EXTRA_DRINKNO);
CheckBox favorite = (CheckBox)findViewById(R.id.favorite);
ContentValues drinkValues = new ContentValues();
drinkValues.put("FAVORITE", favorite.isChecked());
```

```
SQLiteOpenHelper starbuzzDatabaseHelper =
                        new StarbuzzDatabaseHelper(DrinkActivity.this);
try {
    SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
    db.update("DRINK", drinkValues,
                        "_id = ?", new String[] {Integer.toString(drinkNo)});
    db.close();
} catch(SQLiteException e) {
```

```
    Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
    toast.show();
  }
}
```

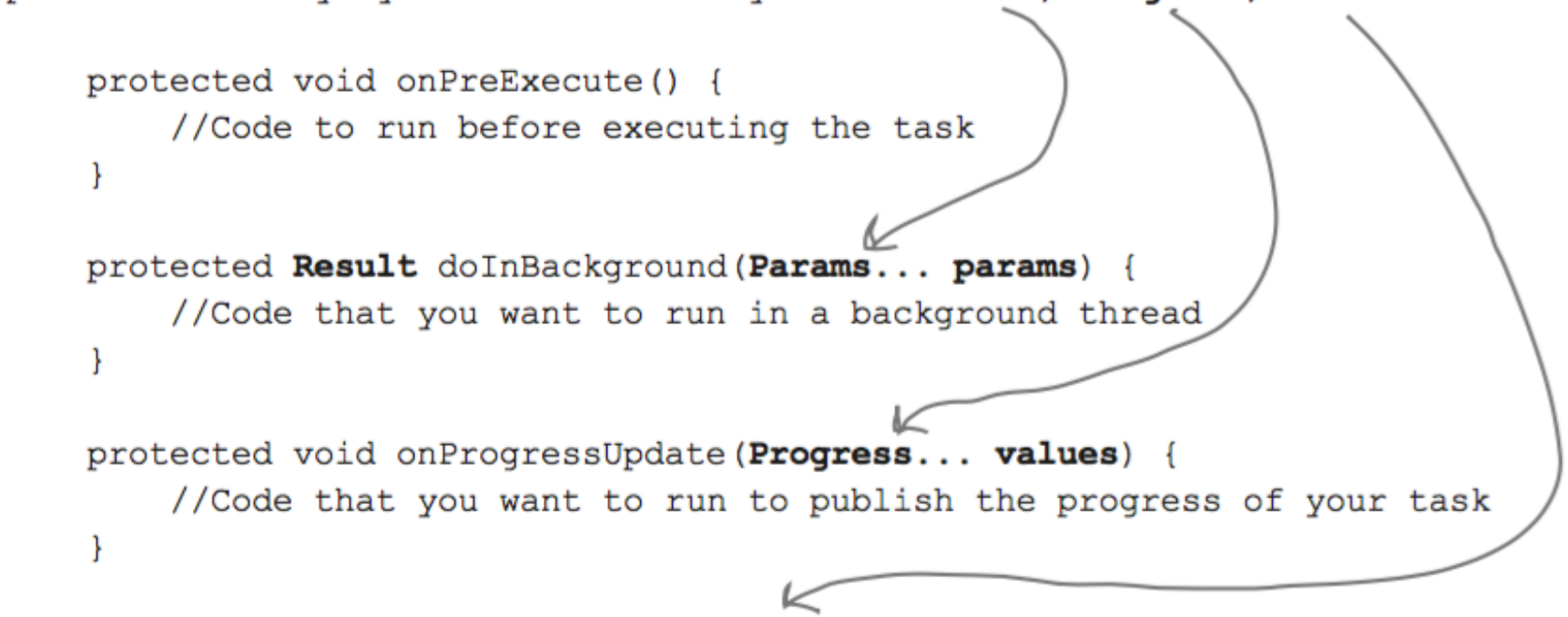
1 **Code that needs to be run before the database code**
The first few lines of code gets the value of the favorite checkbox, and puts it in the `drinkValues ContentValues` object. This code must be run before the database code.

2 **Database code that needs to be run on a background thread**
This updates the DRINK table.

3 **Code that needs to be run after the database code**
If the database is unavailable, we want to display a message to the user. This must run on the main event thread.

AsyncTask realiza tareas asíncronas

```
private class MyAsyncTask extends AsyncTask<Params, Progress, Result>

    protected void onPreExecute() {
        //Code to run before executing the task
    }

    protected Result doInBackground(Params... params) {
        //Code that you want to run in a background thread
    }

    protected void onProgressUpdate(Progress... values) {
        //Code that you want to run to publish the progress of your task
    }

    protected void onPostExecute(Result result) {
        //Code that you wan to run when the task is complete
    }
}
```



La clase AsyncTask

```
private class UpdateDrinkTask extends AsyncTask<Integer, Void, Boolean> {
    ...
    protected Boolean doInBackground(Integer... drinks) {
        ...
    }

    protected void onPostExecute(Boolean... success) {
        ...
    }
}
```

El método onPreExecute()

```
private class UpdateDrinkTask extends AsyncTask<Params, Progress, Result> {

    ContentValues drinkValues;

    protected void onPreExecute() {
        CheckBox favorite = (CheckBox)findViewById(R.id.favorite);
        drinkValues = new ContentValues();
        drinkValues.put("FAVORITE", favorite.isChecked());
    }

    ...

}
```

Before we run the database code, we need to get the value of the favorite checkbox.

```
private class UpdateDrinkTask extends AsyncTask<Integer, Progress, Boolean> {

    ContentValues drinkValues;

    ...

    protected Boolean doInBackground(Integer... drinks) {
        int drinkNo = drinks[0];
        SQLiteOpenHelper starbuzzDatabaseHelper =
                                    new StarbuzzDatabaseHelper(DrinkActivity.this);
        try {
            SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
            db.update("DRINK", drinkValues,
                                    "_id = ?", new String[] {Integer.toString(drinkNo)});
            db.close();
            return true;
        } catch(SQLiteException e) {
            return false;
        }
    }

    ...
}
```

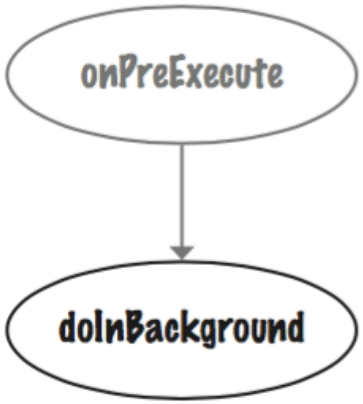
You change this to Integer to match the parameter of the doInBackground() method.

You change this to Boolean to match the return type of the doInBackground() method.

This code runs in a background thread.

This is an array of Integers, but we'll just include one item, the drink ID.

The update() method uses the drinkValues object that the onPreExecute() method created.

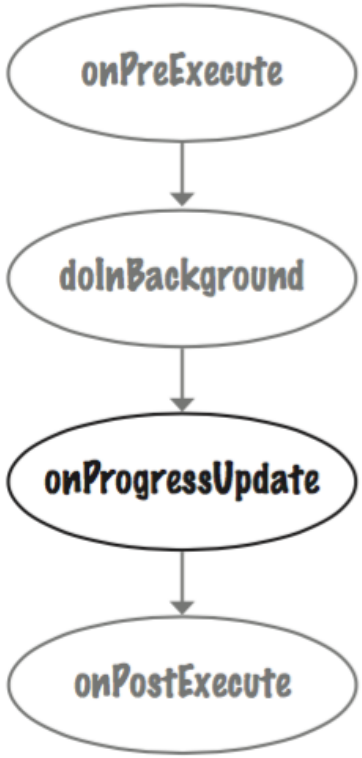


El método doInBackground()

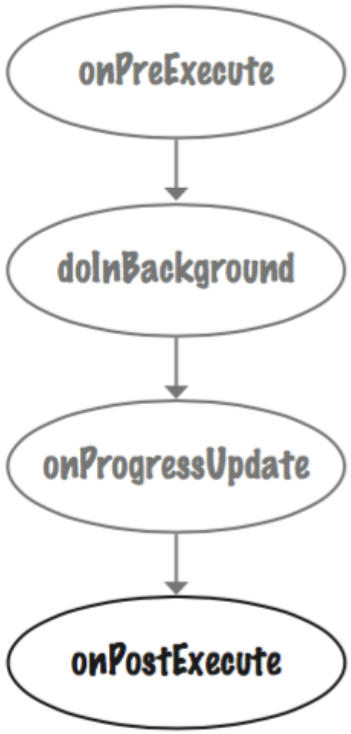
El método onProgressUpdate()

```
protected Boolean doInBackground(Integer... count) {
    for (int i = 0; i < count; i++) {
        publishProgress(i);
    }
}

protected void onProgressUpdate(Integer... progress) {
    setProgress(progress[0]);
}
```

This calls the onProgressUpdate() method, passing in a value of i.



El método onPostExecute()




```
private class UpdateDrinkTask extends AsyncTask<Integer, Void, Boolean> {

    ...

    protected void onPostExecute(Boolean success) {
        if (!success) {
            Toast toast = Toast.makeText(DrinkActivity.this,
                                         "Database unavailable", Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
```

This is Boolean, as our doInBackground() method returns a Boolean.

Pass the Toast the DrinkActivity context.

```
private class UpdateDrinkTask extends AsyncTask<Integer, Void, Boolean> {
    private ContentValues drinkValues;

    protected void onPreExecute() {
        CheckBox favorite = (CheckBox) findViewById(R.id.favorite);
        drinkValues = new ContentValues();
        drinkValues.put("FAVORITE", favorite.isChecked());
    }

    protected Boolean doInBackground(Integer... drinks) {
        int drinkId = drinks[0];
        SQLiteOpenHelper starbuzzDatabaseHelper =
                new StarbuzzDatabaseHelper(DrinkActivity.this);
        try {
            SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
            db.update("DRINK", drinkValues,
                      "_id = ?", new String[] {Integer.toString(drinkId)});
            db.close();
            return true;
        } catch(SQLiteException e) {
            return false;
        }
    }

    protected void onPostExecute(Boolean success) {
        if (!success) {
            Toast toast = Toast.makeText(DrinkActivity.this,
                    "Database unavailable", Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
```

We've defined drinkValues as a private variable, as it's used by the onExecute() and doInBackground() methods.

Before we run the database code, we need to put the value of the favorite checkbox in the drinkValues ContentValues object.

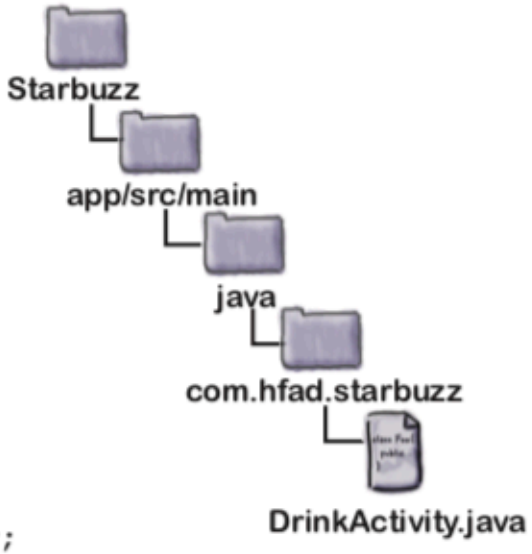
Our database code goes in the doInBackground() method.

After the database code has run in the background, check whether it ran successfully. If it didn't, display a message.

We have to put the code to display a message in the onPostExecute() method, as it needs to be run on the main event thread to update the screen.

Código completo de UpdateDrinkTask

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.view.View;
import android.widget.CheckBox;
import android.content.ContentValues;
import android.os.AsyncTask; ← We're using the AsyncTask class, so we need to import it.

public class DrinkActivity extends Activity {

    public static final String EXTRA_DRINKID = "drinkId";
```



El código de DrinkActivity

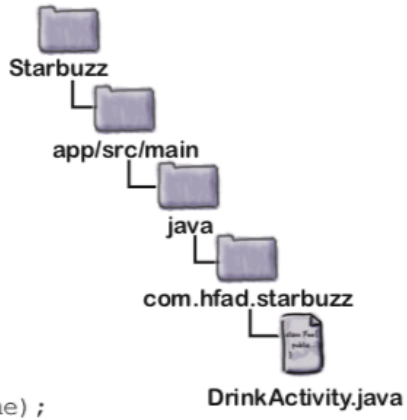
We don't need to change the onCreate() method, we're just showing it for completeness.

```
@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);

    //Create a cursor
    SQLiteOpenHelper starbuzzDatabaseHelper = new StarbuzzDatabaseHelper(this);
    try {
        SQLiteDatabase db = starbuzzDatabaseHelper.getReadableDatabase();
        Cursor cursor = db.query("DRINK",
                new String[]{"NAME", "DESCRIPTION", "IMAGE_RESOURCE_ID", "FAVORITE"},
                "_id = ?",
                new String[]{Integer.toString(drinkId)},
                null, null, null);
```

The code continues on the next page.

El código de DrinkActivity

```
        //Move to the first record in the Cursor
        if (cursor.moveToFirst()) {
            //Get the drink details from the cursor
            String nameText = cursor.getString(0);
            String descriptionText = cursor.getString(1);
            int photoId = cursor.getInt(2);
            boolean isFavorite = (cursor.getInt(3) == 1);

            //Populate the drink name
            TextView name = (TextView) findViewById(R.id.name);
            name.setText(nameText);

            //Populate the drink description
            TextView description = (TextView) findViewById(R.id.description);
            description.setText(descriptionText);

            //Populate the drink image
            ImageView photo = (ImageView) findViewById(R.id.photo);
            photo.setImageResource(photoId);
            photo.setContentDescription(nameText);

            //Populate the favorite checkbox
            CheckBox favorite = (CheckBox)findViewById(R.id.favorite);
            favorite.setChecked(isFavorite);
        }
        cursor.close();
        db.close();
    } catch (SQLiteException e) {
        Toast toast = Toast.makeText(this,
                "Database unavailable",
                Toast.LENGTH_SHORT);
        toast.show();
    }
}
```



None of the code on this page needs to change.

El código de DrinkActivity

```
//Update the database when the checkbox is clicked
public void onFavoriteClicked(View view){
    int drinkId = (Integer) getIntent().getExtras().get(EXTRA_DRINKID);

    //Get the value of the checkbox
    CheckBox favorite = (CheckBox) findViewById(R.id.favorite);
    ContentValues drinkValues = new ContentValues();
    drinkValues.put("FAVORITE", favorite.isChecked());

    //Get a reference to the database and update the FAVORITE column
    SQLiteOpenHelper starbuzzDatabaseHelper =
            new StarbuzzDatabaseHelper(this);
    try {
        SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
        db.update("DRINK",
                drinkValues,
                "_id = ?",
                new String[] {Integer.toString(drinkId)});
        db.close();
    } catch(SQLiteException e) {
        Toast toast = Toast.makeText(this, "Database unavailable", Toast.LENGTH_SHORT);
        toast.show();
    }

    new UpdateDrinkTask().execute(drinkId); ← Execute the task.
}
```

Delete all these lines of code, as we're now using an AsyncTask for these actions.

El código de DrinkActivity

```
//Inner class to update the drink.
private class UpdateDrinkTask extends AsyncTask<Integer, Void, Boolean> {
    private ContentValues drinkValues;

    protected void onPreExecute() {
        CheckBox favorite = (CheckBox) findViewById(R.id.favorite);
        drinkValues = new ContentValues();
        drinkValues.put("FAVORITE", favorite.isChecked());
    }

    protected Boolean doInBackground(Integer... drinks) {
        int drinkId = drinks[0];
        SQLiteOpenHelper starbuzzDatabaseHelper =
                new StarbuzzDatabaseHelper(DrinkActivity.this);
        try {
            SQLiteDatabase db = starbuzzDatabaseHelper.getWritableDatabase();
            db.update("DRINK", drinkValues,
                    "_id = ?", new String[] {Integer.toString(drinkId)});
            db.close();
            return true;
        } catch(SQLiteException e) {
            return false;
        }
    }

    protected void onPostExecute(Boolean success) {
        if (!success) {
            Toast toast = Toast.makeText(DrinkActivity.this,
                    "Database unavailable", Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
}
```

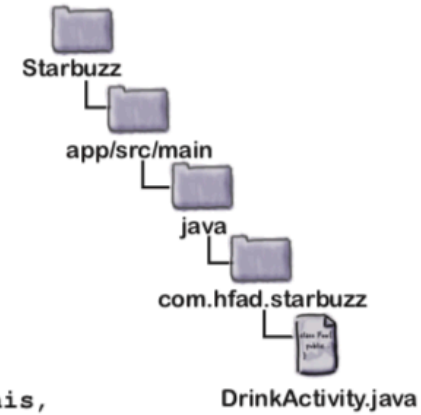
Add the AsyncTask to the activity as an inner class.

Before the database code runs, put the value of the checkbox in the drinkValues ContentValues object.

Run the database code in a background thread.

Update the value of the FAVORITE column.

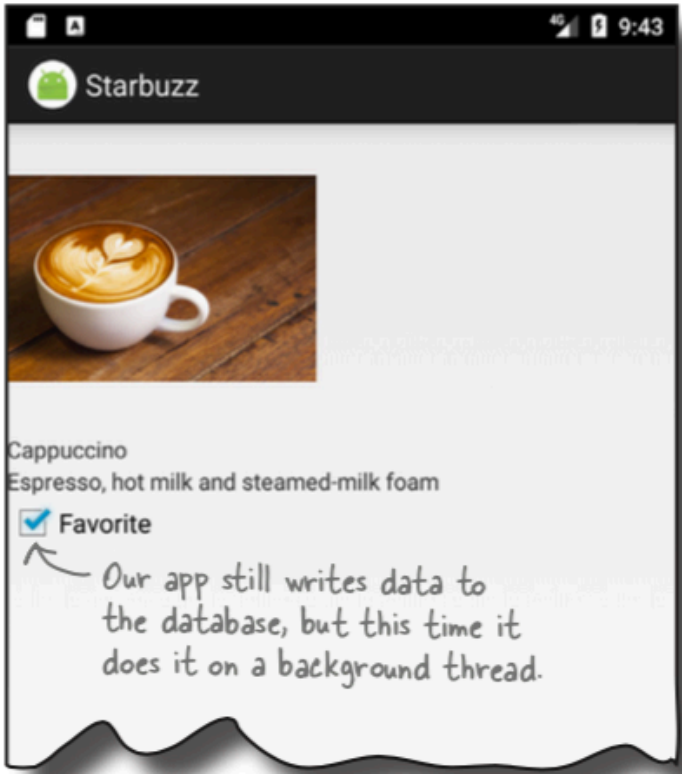
If the database code didn't run correctly, display a message to the user.



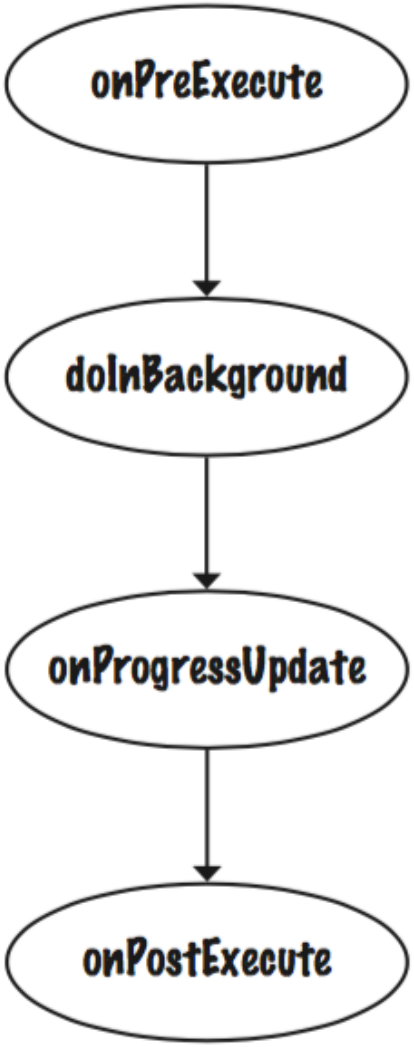
El código de DrinkActivity

Prueba

In an ideal world, all of your database code should run in the background. We're not going to change our other Starbuzz activities to do this, but why not make this change yourself?



Resumen de AsyncTask



1. **onPreExecute() is used to set up the task.**
It's called before the background task begins, and runs on the main event thread.

2. **doInBackground() runs in the background thread.**
It runs immediately after `onPreExecute()`. You can specify what type of parameters it has, and what its return type is.

3. **onProgressUpdate() is used to display progress.**
It runs in the main event thread when the `doInBackground()` method calls `publishProgress()`.

4. **onPostExecute() is used to display the task outcome to the user when doInBackground has finsihed.**
It runs in the main event thread. It takes the return value of `doInBackground()` as a parameter.

BULLET POINTS

- The `CursorAdapter changeCursor()` method replaces the cursor currently used by a cursor adapter with a new cursor that you provide. It then closes the old cursor.
- Run your database code in a background thread using `AsyncTask`.