|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE INGENIERÍA – ESCUELA DE COMPUTACIÓN** |
|  | TALLER ANDROID  **TEMA III - Eventos de Android**  Listener e Intents |

**I Objetivos**

* Comprender el uso de eventos en android.
* Asociar los controles de las vistas con las actividades para acceder y asignar sus propiedades.

**II Introducción**

Android representa desde su UI diferentes controles que pueden ser accedidos desde las clases que extienden de Activity de java o kotlin, de esta manera puede tomar entradas de usuario y eventos que suceden sobre los controles. Es posible definir formularios y con “listeners” (respuestas de eventos) realizar operaciones sobre la información contenida.

**III Desarrollo**

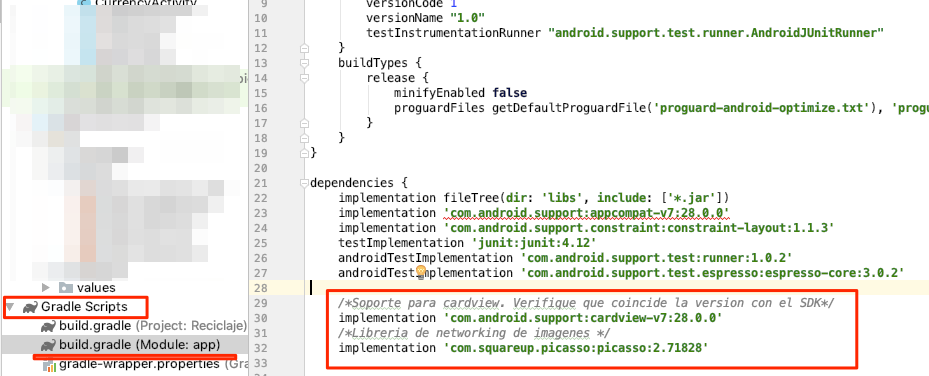
**1) Crear un proyecto con el nombre Reciclaje. El paquete a utilizar se llama sv.edu.udb.reciclaje.**

**2) Crear 4 Actividades con los siguientes nombres:**

* **MainActivity (Por defecto)**
* **CurrencyActivity**
* **DetailActivity**
* **ListActivity**

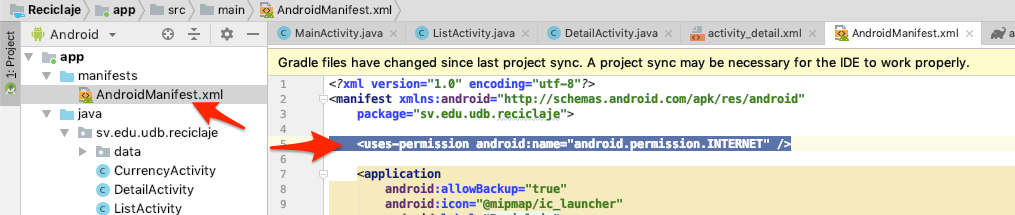
**3) Agregar en la sección de dependencias el siguiente código:**

|  |
| --- |
| */\*Soporte para cardview. Verifique que coincide la version con el SDK\*/* implementation **'com.android.support:cardview-v7:28.0.0'** */\*Libreria de networking de imagenes \*/* implementation **'com.squareup.picasso:picasso:2.71828'** |

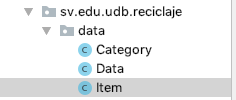


**4) Agregaremos permisos de internet al Manifest**

|  |
| --- |
| <**uses-permission android:name="android.permission.INTERNET"** /> |



**5) Crear un paquete denominado sv.edu.udb.data. Contendrá las clases Category.java, Item.java y Data.java**



**6) Crear la clase Category.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** java.io.Serializable; **import** java.util.List;  **public class** Category **implements** Serializable {   **private** String **id**;  **private** String **name**;  **private** List<Item> **itemList**;   **public** String getId() {  **return id**;  }   **public void** setId(String id) {  **this**.**id** = id;  }   **public** String getName() {  **return name**;  }   **public void** setName(String name) {  **this**.**name** = name;  }   **public** List<Item> getItemList() {  **return itemList**;  }   **public void** setItemList(List<Item> itemList) {  **this**.**itemList** = itemList;  } } |

**7) Crear la clase Item.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** java.io.Serializable;  **public class** Item **implements** Serializable {   **private** String **name**;  **private** Double **price**;  **private** String **description**;  **private** String **image**;   **public** String getName() {  **return name**;  }   **public void** setName(String name) {  **this**.**name** = name;  }   **public** Double getPrice() {  **return price**;  }   **public void** setPrice(Double price) {  **this**.**price** = price;  }   **public** String getDescription() {  **return description**;  }   **public void** setDescription(String description) {  **this**.**description** = description;  }   **public** String getImage() {  **return image**;  }   **public void** setImage(String image) {  **this**.**image** = image;  } } |

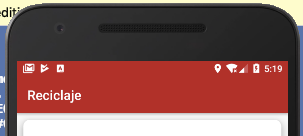
**8) Crear la clase Data.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** java.util.ArrayList; **import** java.util.List;  **public class** Data **implements** Serializable {   **private** List<Category> **categoryList**;    **public** Data(){  **this**.**categoryList** = **new** ArrayList<Category>();   Category aluminio = **new** Category();  aluminio.setName(**"Aluminio"**);  aluminio.setId(**"1"**);  aluminio.setItemList(**new** ArrayList<Item>());   Item latas = **new** Item();  latas.setName(**"Latas de Aluminio"**);  latas.setDescription(**"Latas de soda, productos enlatados, etc"**);  latas.setPrice(1.0);  latas.setImage(**"https://i.ytimg.com/vi/QTSDhriq3WA/maxresdefault.jpg"**);   Item desperdicioAlumnio = **new** Item();  desperdicioAlumnio.setName(**"Desperdicio aluminio"**);  desperdicioAlumnio.setDescription(**"Residuo de alumnio de procesos industriales"**);  desperdicioAlumnio.setPrice(1.50);  desperdicioAlumnio.setImage(**"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQdrSU88nlAmDsYqp8WqpGybs3Pog3670bZJ\_AE4u1gyIG1acfZEw"**);   aluminio.getItemList().add(latas);  aluminio.getItemList().add(desperdicioAlumnio);   **categoryList**.add(aluminio);   Category derivadosPapel = **new** Category();  derivadosPapel.setName(**"Derivados del papel"**);  derivadosPapel.setId(**"2"**);  derivadosPapel.setItemList(**new** ArrayList<Item>());   Item papel = **new** Item();  papel.setName(**"Papel blanco"**);  papel.setPrice(0.40);  papel.setDescription(**"Papel blanco, papel bond, cuadernos, etc"**);  papel.setImage(**"http://www.inforeciclaje.com/imagenes/papel-reciclado.jpg"**);   Item papelPeriodico = **new** Item();  papelPeriodico.setName(**"Papel de periódico"**);  papelPeriodico.setDescription(**"Periodico de toda gama"**);  papelPeriodico.setPrice(0.25);  papelPeriodico.setImage(**"https://comohacerpara.com/imgn/07744-reciclar-papel-periodico.jpg"**);   derivadosPapel.getItemList().add(papel);  derivadosPapel.getItemList().add(papelPeriodico);   **categoryList**.add(derivadosPapel);   Category madera = **new** Category();  madera.setName(**"Madera procesada o tratada, muebles, construcción, etc"**);  madera.setId(**"3"**);  madera.setItemList(**new** ArrayList<Item>());   Item maderaMuebles = **new** Item();  maderaMuebles.setName(**"Madera de muebles"**);  maderaMuebles.setDescription(**"Madera proveniente de reciclaje de muebles, sofás, camas, etc"**);  maderaMuebles.setPrice(0.30);  maderaMuebles.setImage(**"http://infomaderas.com/wp-content/uploads/2013/04/madera-reciclada.jpg"**);   madera.getItemList().add(maderaMuebles);   **categoryList**.add(madera);   }    **public** List<Category> getCategoryList() {  **return categoryList**;  }   **public void** setCategoryList(List<Category> categoryList) {  **this**.**categoryList** = categoryList;  } } |

**9) Vamos a deshabilitar el giro de pantalla para cada actividad. Además, haremos que la actividad llamada CurrencyActivity tenga opciones de navegación y regrese a la MainActivity, la propiedad se llama parentActivityName**



**10) Vamos a editar el estilo del AppTheme ubicado en el xml de estilos.**



|  |
| --- |
| <**resources**>   *<!-- Base application theme. -->* <**style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"**>  *<!-- Customize your theme here. -->* <**item name="colorPrimary"**>#C20E0E</**item**>  <**item name="colorPrimaryDark"**>#C20E0E</**item**>  <**item name="colorAccent"**>@color/colorAccent</**item**>  </**style**>  </**resources**> |

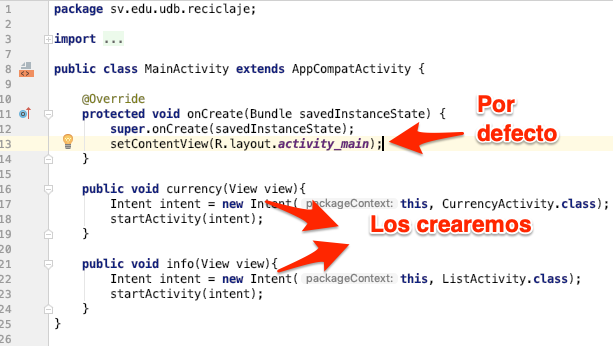
**11) Vamos a crear dos botones en MainActivity, para que nos permita tener una navegación simple:**



activity\_main.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity"  app:cardCornerRadius="10dp"  app:cardElevation="10dp"  android:layout\_margin="10dp"  app:contentPadding="10dp"** >  <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"** >   <**Button  android:id="@+id/button"  android:layout\_width="match\_parent"  android:layout\_height="181dp"  android:background="#9B6D69"  android:onClick="currency"  android:text="Actividad Tasa de cambio"  android:layout\_margin="30dp"  android:textColor="#fff"  android:textSize="20sp"** />   <**Button  android:id="@+id/button2"  android:layout\_width="match\_parent"  android:layout\_height="181dp"  android:layout\_below="@+id/button"  android:background="#A27009"  android:onClick="info"  android:text="Actividad Reciclaje"  android:layout\_margin="30dp"  android:textColor="#fff"  android:textSize="20sp"** />  </**RelativeLayout**>  </**android.support.v7.widget.CardView**> |

**12) Ahora vamos a editar nuestro primer actividad java:**



MainActivity.java

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.content.Intent; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View;  **public class** MainActivity **extends** AppCompatActivity {   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  }   **public void** currency(View view){  Intent intent = **new** Intent(**this**, CurrencyActivity.**class**);  startActivity(intent);  }   **public void** info(View view){  Intent intent = **new** Intent(**this**, ListActivity.**class**);  startActivity(intent);  } } |

**13) CurrencyActivity es una pequeña aplicación para tasas de cambio. Vamos a editar el activity\_currency.xml para que tenga la apariencia a continuación:**

|  |  |
| --- | --- |
|  | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".CurrencyActivity"  app:cardCornerRadius="10dp"  app:cardElevation="10dp"  android:layout\_margin="5dp"  android:padding="5dp"  app:contentPadding="5dp"** >   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>    <**Spinner  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/currencyfrom"** >  </**Spinner**>   <**Spinner  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/currencyto"  android:layout\_below="@+id/currencyfrom"** >  </**Spinner**>   <**EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/quantity"  android:hint="Cantidad"  android:layout\_below="@+id/currencyto"  android:inputType="numberDecimal"** />   <**Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/evaluate"  android:onClick="convert"  android:text="Calcular"  android:layout\_below="@+id/quantity"  android:textColor="#fff"  android:backgroundTint="#94200C"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/result"  android:layout\_below="@+id/evaluate"  android:textSize="30sp"  android:background="#072669"  android:text="0"  android:textColor="#fff"  android:textAlignment="center"** />   </**RelativeLayout**>  </**android.support.v7.widget.CardView**> |

**14) Ahora editaremos nuestra CurrencyActivity.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.View; **import** android.widget.ArrayAdapter; **import** android.widget.EditText; **import** android.widget.Spinner; **import** android.widget.TextView; **import** android.widget.Toast;  **import** java.text.DecimalFormat;  **public class** CurrencyActivity **extends** AppCompatActivity {   Spinner **currencyFrom**;  Spinner **currencyTo**;  EditText **quantity**;  TextView **result**;   String **currencies**[] = {**"Dólar"**,**"Euro"**,**"Peso Mexicano"**,**"Peso Colombiano"**};   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_currency***);     **currencyFrom** = findViewById(R.id.***currencyfrom***);  **currencyTo** = findViewById(R.id.***currencyto***);  **quantity** = findViewById(R.id.***quantity***);  **result** = findViewById(R.id.***result***);    ArrayAdapter<String> listAdapter = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_spinner\_dropdown\_item***,**currencies**);   **currencyFrom**.setAdapter(listAdapter);   **currencyTo**.setAdapter(listAdapter);   }    **public void** convert(View view) {  String currencyFromString = **currencyFrom**.getSelectedItem().toString();  String currencyToString = **currencyTo**.getSelectedItem().toString();   Double value = conversion(currencyFromString,currencyToString);  Toast.*makeText*(**this**, value.toString(),Toast.***LENGTH\_LONG***).show();   DecimalFormat df = **new** DecimalFormat(**".##"**);   **result**.setText(**""**+df.format(value));  }    **private** Double conversion(String currencyFrom, String currencyTo){  Double result;  Double factor = 0.0;  **if**(currencyFrom.equals(**"Dólar"**)){  **if**(currencyTo.equals(**"Dólar"**)) factor = 1.0;  **if**(currencyTo.equals(**"Peso Mexicano"**)) factor = 19.40;  **if**(currencyTo.equals(**"Euro"**)) factor = 0.89;  **if**(currencyTo.equals(**"Peso Colombiano"**)) factor = 3185.50;  }   **if**(currencyFrom.equals(**"Euro"**)){  **if**(currencyTo.equals(**"Dólar"**)) factor = 1.12;  **if**(currencyTo.equals(**"Peso Mexicano"**)) factor = 21.77;  **if**(currencyTo.equals(**"Euro"**)) factor = 1.0;  **if**(currencyTo.equals(**"Peso Colombiano"**)) factor = 3573.48;  }   **if**(currencyFrom.equals(**"Peso Mexicano"**)){  **if**(currencyTo.equals(**"Dólar"**)) factor = 0.052;  **if**(currencyTo.equals(**"Peso Mexicano"**)) factor = 1.0;  **if**(currencyTo.equals(**"Euro"**)) factor = 0.046;  **if**(currencyTo.equals(**"Peso Colombiano"**)) factor = 164.17;  }    **if**(currencyFrom.equals(**"Peso Colombiano"**)){  **if**(currencyTo.equals(**"Dólar"**)) factor = 0.00031;  **if**(currencyTo.equals(**"Peso Mexicano"**)) factor = 0.0061;  **if**(currencyTo.equals(**"Euro"**)) factor = 0.00028;  **if**(currencyTo.equals(**"Peso Colombiano"**)) factor = 1.0;  }   String stringValue = **quantity**.getText().toString();   **if**(stringValue.isEmpty() || stringValue.equals(**""**) ) **return** 0.0;   Double quantityValue = Double.*parseDouble*(stringValue);   result = quantityValue\*factor;   **return** result;  } } |

**15) ListActivity nos mostrará un listado de categorías e items por categoría. Editaremos activity\_list.xml**

|  |  |
| --- | --- |
|  | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".ListActivity"  app:cardCornerRadius="10dp"  app:cardElevation="10dp"  android:layout\_margin="5dp"  app:contentPadding="5dp"** >   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**Spinner  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/categoryspinner"** >   </**Spinner**>   <**ListView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/itemlistview"  android:layout\_below="@+id/categoryspinner"** >   </**ListView**>   </**RelativeLayout**>  </**android.support.v7.widget.CardView**> |

**16) ListActivity.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.content.Intent; **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log; **import** android.view.MenuItem; **import** android.view.View; **import** android.widget.AdapterView; **import** android.widget.ArrayAdapter; **import** android.widget.ListView; **import** android.widget.Spinner;  **import** java.util.ArrayList; **import** java.util.List;  **import** sv.edu.udb.reciclaje.data.Category; **import** sv.edu.udb.reciclaje.data.Data; **import** sv.edu.udb.reciclaje.data.Item;  **public class** ListActivity **extends** AppCompatActivity {   List<String> **categoryStringList** = **new** ArrayList<>();  List<String> **itemStringList** = **new** ArrayList<>();  Data **data** = **new** Data();   Spinner **categorySpinner**;  ListView **itemListView**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_list***);    ActionBar actionBar = getSupportActionBar();  actionBar.setDisplayHomeAsUpEnabled(**true**);   **categorySpinner** = findViewById(R.id.***categoryspinner***);  **itemListView** = findViewById(R.id.***itemlistview***);     **for**(Category category:**data**.getCategoryList()){  **categoryStringList**.add(category.getName());  }   **categorySpinner**.setAdapter(**new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_spinner\_dropdown\_item***,**categoryStringList**));   **categorySpinner**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> adapterView, View view, **int** i, **long** l) {  loadListView();  }   @Override  **public void** onNothingSelected(AdapterView<?> adapterView) {   }  });   }   @Override  **protected void** onResume() {  **super**.onResume();  loadListView();  }   **void** loadListView(){  **final int** index = **categorySpinner**.getSelectedItemPosition();   **itemStringList**.clear();   **for**(Item selected:**data**.getCategoryList().get(index).getItemList()){  **itemStringList**.add(selected.getName());  }   ArrayAdapter<String> listViewAdapter = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_list\_item\_1***,**itemStringList**);   **itemListView**.setAdapter(listViewAdapter);   **itemListView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  @Override  **public void** onItemClick(AdapterView<?> adapterView, View view, **int** i, **long** l) {  Item item = **data**.getCategoryList().get(index).getItemList().get(i);  Log.*d*(**"UDBLog"**, item.getDescription() );  Log.*d*(**"UDBLog"**, item.getName() );  Log.*d*(**"UDBLog"**, item.getPrice() + **""** );   Intent intent = **new** Intent(ListActivity.**this**,DetailActivity.**class**);  intent.putExtra(**"item"**,item);  startActivity(intent);   }  });   **itemListView**.deferNotifyDataSetChanged();  }   @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** android.R.id.***home***:  finish();  **return true**;  }   **return super**.onOptionsItemSelected(item);  } } |

**17) Editaremos el activity\_detail.xml**

|  |  |
| --- | --- |
|  | *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".DetailActivity"  app:cardElevation="10dp"  app:cardCornerRadius="10dp"  android:layout\_margin="5dp"  app:contentPadding="5dp"** >   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/itemname"  android:textSize="40sp"  android:background="#94200C"  android:textColor="#fff"** />  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/itemdescription"  android:layout\_below="@+id/itemname"  android:textSize="20sp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Valor: $"  android:id="@+id/labelvalue"  android:textSize="20sp"  android:layout\_below="@+id/itemdescription"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/itemvalue"  android:layout\_below="@+id/itemdescription"  android:layout\_toRightOf="@+id/labelvalue"  android:textSize="20sp"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text=" por libra"  android:id="@+id/labelpound"  android:textSize="20sp"  android:layout\_below="@+id/itemdescription"  android:layout\_toRightOf="@+id/itemvalue"** />  <**ImageView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/itemimage"  android:layout\_below="@+id/itemvalue"  android:layout\_margin="10dp"** />  </**RelativeLayout**>  </**android.support.v7.widget.CardView**> |

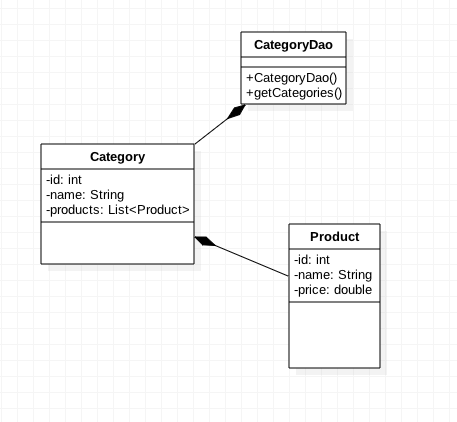
**18) DetailActivity.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.view.MenuItem; **import** android.widget.ImageView; **import** android.widget.TextView;  **import** com.squareup.picasso.Picasso;  **import** sv.edu.udb.reciclaje.data.Item;  **public class** DetailActivity **extends** AppCompatActivity {   Item item;   TextView itemName;  TextView itemDescription;  TextView itemValue;  ImageView itemImage;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.activity\_detail);   ActionBar actionBar = getSupportActionBar();  actionBar.setDisplayHomeAsUpEnabled(**true**);   itemName = findViewById(R.id.itemname);  itemDescription = findViewById(R.id.itemdescription);  itemValue = findViewById(R.id.itemvalue);  itemImage = findViewById(R.id.itemimage);   item = (Item) getIntent().getSerializableExtra(**"item"**);   itemName.setText(item.getName());  itemValue.setText(item.getPrice() + **""**);  itemDescription.setText(item.getDescription());   Picasso.get().load(item.getImage()).into(itemImage);   }   @Override  **public boolean** onOptionsItemSelected(MenuItem item) {  **switch** (item.getItemId()) {  **case** android.R.id.home:  finish();  **return true**;  }   **return super**.onOptionsItemSelected(item);  } } |

**IV Ejercicios**

|  |
| --- |
| Proyecto: **Sistema de cotizaciones móvil.** |
| Localmente vía GreenDAO, almacenará su base de datos de clientes. Debe guardar nombre, dirección, teléfono, email.  Listará categorías de productos, al seleccionarlo desplegará los productos con sus precios. Esta información será accedida vía webservice.  El usuario vía GreenDAO podrá guardar la información de la cotización. Deberá guardar el nombre de usuario de aplicación, el correo(destino), los IDs de los item en la cotización individualmente, cantidad, precio sugerido (webservice) precio ofertado (manual) y el total.  Finalmente, la información será enviada en formato json vía petición POST hacia el webservice, el cual enviará un correo detallado hacia el cliente.  La aplicación pedirá usuario y password (Esto estará habilitado al final del taller). |

* 1. Crear las siguientes clases para su proyecto:



* 1. Crear una actividad para cotización de productos. Debe mostrar la lista de categorías y al seleccionar un item de esta lista, en una actividad nueva debe mostrar los productos. Al seleccionar uno de ellos, debe mostrar el nombre, el precio y agregar dos EditText para ingresar la cantidad y precio de venta. Agregar un botón, que permita tomar el precio del producto (desde la clase Product) y desde el campo de cantidad y precio de venta. A continuación calcule el total de descuento, el porcentaje de descuento y el total a pagar por ese producto. Si uno de los campos están vacíos el teléfono debe vibrar y emitir un toast indicando el campo de error.
  2. Agregue una lista de clientes (los clientes pueden estar hardcoded por ahora), muéstrela con un ListView. Agregue un botón (puede investigar sobre el uso del floating button si lo desea) y muestre su formulario de clientes. De su formulario, valide los campos de entrada para que no estén vacíos.
  3. En su menú habilite las opciones: Listado de clientes y Listado de productos.