



LABORATORIO 5



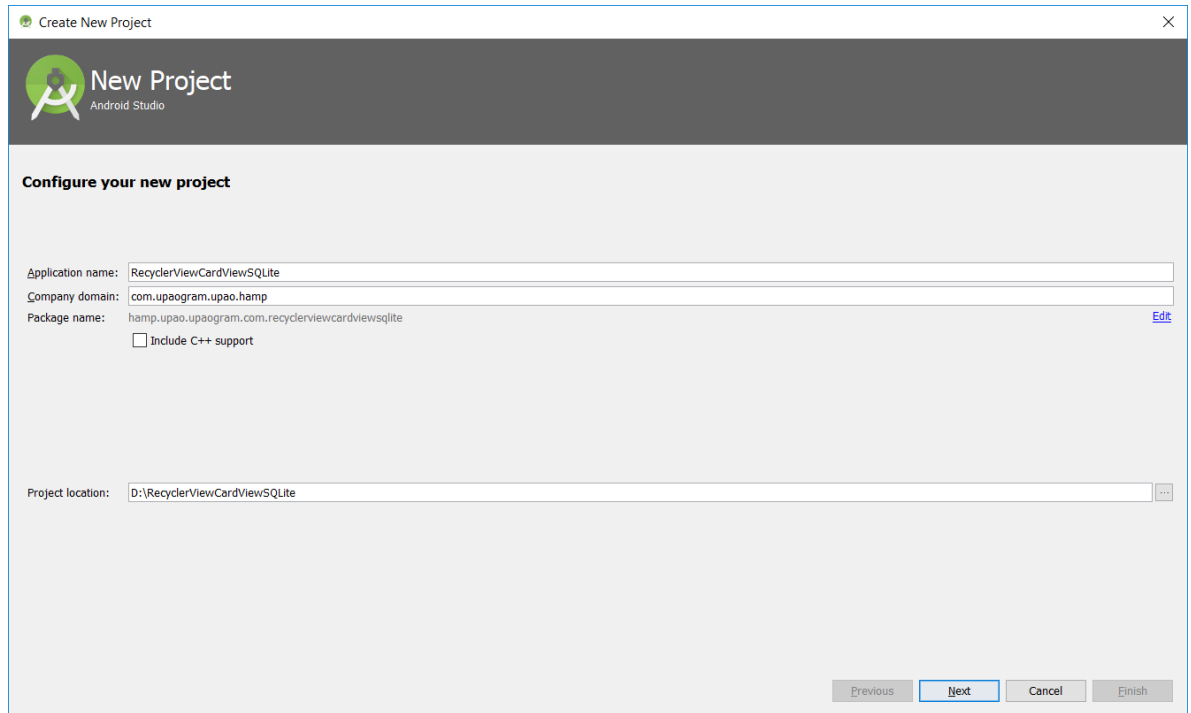
Henry A. Mendoza Puerta

Tabla de contenido

Creando Proyecto.....	2
Creando Paquetes	3
Creando Clase Product	4
Creando Clase Utilities	4
Creando Clase ConnectionSQLiteHelper	5
Editando Diseño de activity_main.....	6
Agregando Nuevo Activity InsertProductoActivity(Empty Activity)	7
Agregando Nuevo Activity ListProductoActivity(Empty Activity)	9
Agregando Codigo en MainActivity.....	11
Agregando Codigo en InsertProductActivity	12
Agregando Recurso Layout itemsproducts	13
Creando Clase Adapter.....	16
Implementando Activity Clase ListProductsActivity.....	17
Implementar metodo eliminar.....	18
Implementar eliminar deslizando registro a la derecha	19

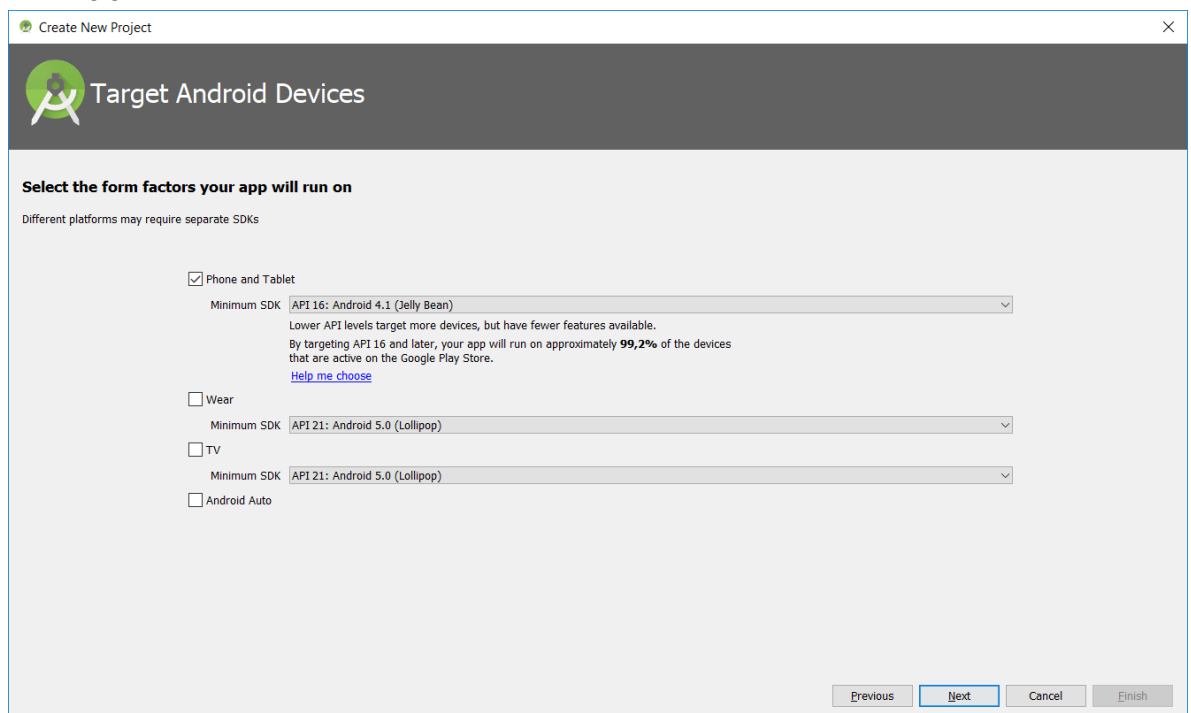
Creando Proyecto

1. Datos de Proyecto



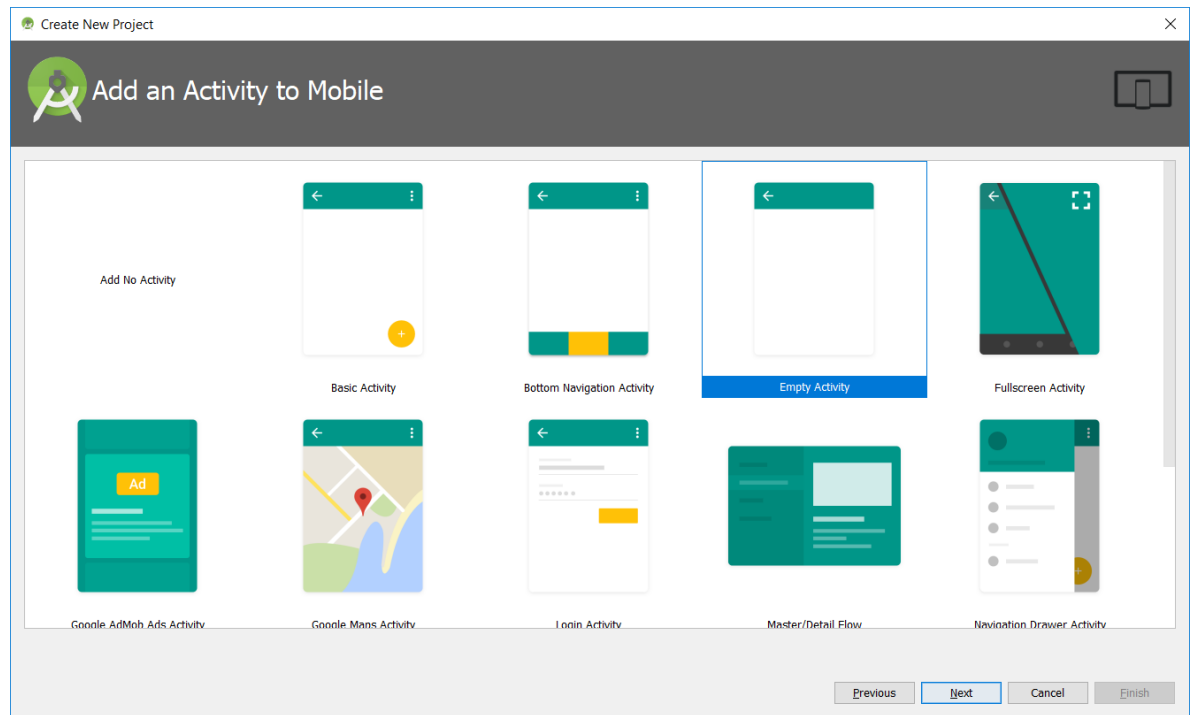
The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar says 'Create New Project'. The main header area has the Android Studio logo and the text 'New Project' and 'Android Studio'. Below this, the section 'Configure your new project' is visible. It contains several input fields: 'Application name' with the value 'RecyclerViewCardViewSQLite', 'Company domain' with 'com.upaogram.upao.hamp', and 'Package name' with 'hamp.upao.upaogram.com.recyclerviewcardviewsqllite'. There is an unchecked checkbox for 'Include C++ support'. At the bottom, the 'Project location' is set to 'D:\RecyclerViewCardViewSQLite'. Navigation buttons at the bottom right include 'Previous', 'Next' (highlighted), 'Cancel', and 'Finish'.

2. Minimo SDK



The screenshot shows the 'Create New Project' dialog in Android Studio, specifically the 'Target Android Devices' step. The title bar says 'Create New Project'. The main header area has the Android Studio logo and the text 'Target Android Devices'. Below this, the section 'Select the form factors your app will run on' is visible, with a note: 'Different platforms may require separate SDKs'. There are four options with checkboxes: 'Phone and Tablet' (checked), 'Wear' (unchecked), 'TV' (unchecked), and 'Android Auto' (unchecked). Under 'Phone and Tablet', there is a 'Minimum SDK' dropdown menu set to 'API 16: Android 4.1 (Jelly Bean)'. Below this dropdown, there is explanatory text: 'Lower API levels target more devices, but have fewer features available. By targeting API 16 and later, your app will run on approximately 99,2% of the devices that are active on the Google Play Store.' and a link 'Help me choose'. The other three options ('Wear', 'TV', 'Android Auto') also have 'Minimum SDK' dropdown menus, all set to 'API 21: Android 5.0 (Lollipop)'. Navigation buttons at the bottom right include 'Previous', 'Next' (highlighted), 'Cancel', and 'Finish'.

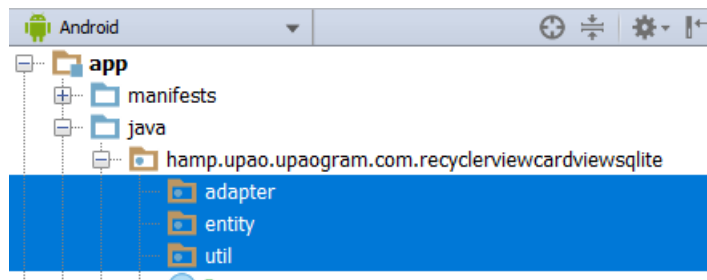
3. Activity tipo Empty Activity



Creando Paquetes

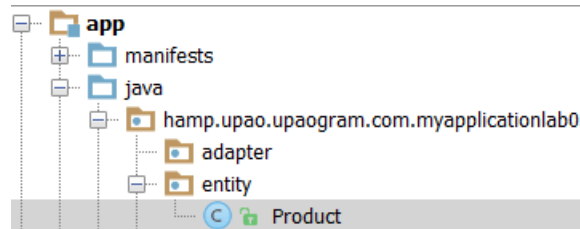
4. Debes crear 4 paquetes en el proyecto.

- adapter
- entity.
- util.



Creando Clase Product

5. Debes crear una clase Product en el paquete entity, esta clase debe contener la información a registrar y mostrar de un producto.



Debe generar los métodos get y set para sus atributos.

```
public class Product {
    private Integer id;
    private String name;
    private Integer quantity;
    private String image;

    public Product() {

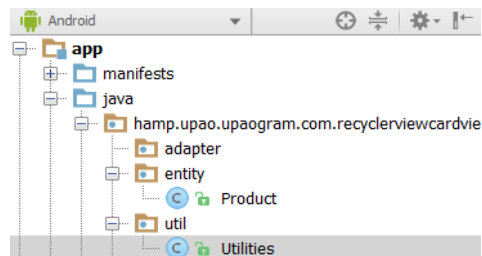
    }

    public Product(String name, Integer quantity,String image) {
        this.name = name;
        this.quantity = quantity;
        this.image=image;
    }

    public Product(Integer id, String name, Integer quantity, String image) {
        this.id = id;
        this.name = name;
        this.quantity = quantity;
        this.image=image;
    }
}
```

Creando Clase Utilities

6. Debes crear una clase Utilities en el paquete Util, esta clase se encuentra el código de la estructura de las tablas.



```

package hamp.upao.upaogram.com.recyclerviewcardviewsqllite.util;

/**
 * Created by hamp on 19/09/2017.
 */

public class Utilities {

    //Constantes campos tabla usuario
    public static final String TABLE_PRODUCT="mascota";
    public static final String COLUMN_ID_PRODUCT="id_product";
    public static final String COLUMN_NAME="name_product";
    public static final String COLUMN_QUANTITY="quantity_product";

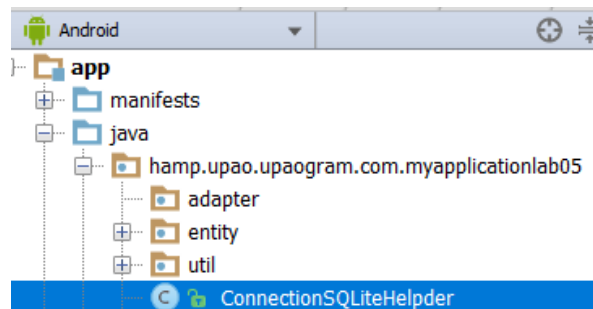
    public static final String CREATE_TABLE_PRODUCT="CREATE TABLE " +
        ""+TABLE_PRODUCT+" (" +COLUMN_ID_PRODUCT+" INTEGER PRIMARY KEY AUTOINCREMENT, "
        +COLUMN_NAME+" TEXT, "+COLUMN_QUANTITY+" INTEGER) ";

}

```

Creando Clase ConnectionSQLiteHelper

- Debes crear una clase ConnectionSQLiteHelper, no se debe crear dentro de ninguno de los paquetes creados.



```

public class ConnectionSQLiteHelper extends SQLiteOpenHelper {

    public ConnectionSQLiteHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(Utilities.CREATE_TABLE_PRODUCT);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int versionAntigua, int versionNueva) {
        db.execSQL("DROP TABLE IF EXISTS "+Utilities.TABLE_PRODUCT);
        onCreate(db);
    }

}

```

Editando Diseño de activity_main

8. Ingrese al archivo XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"

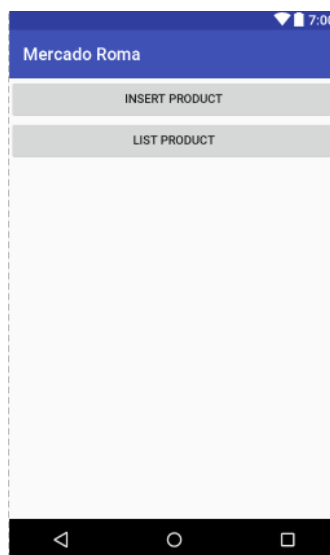
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context="hamp.upao.upaogram.com.recyclerviewcardviewsqllite.MainActi
vity">

    <Button
        android:id="@+id/btnCreateProduct"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert Product"
        android:onClick="onClick"/>

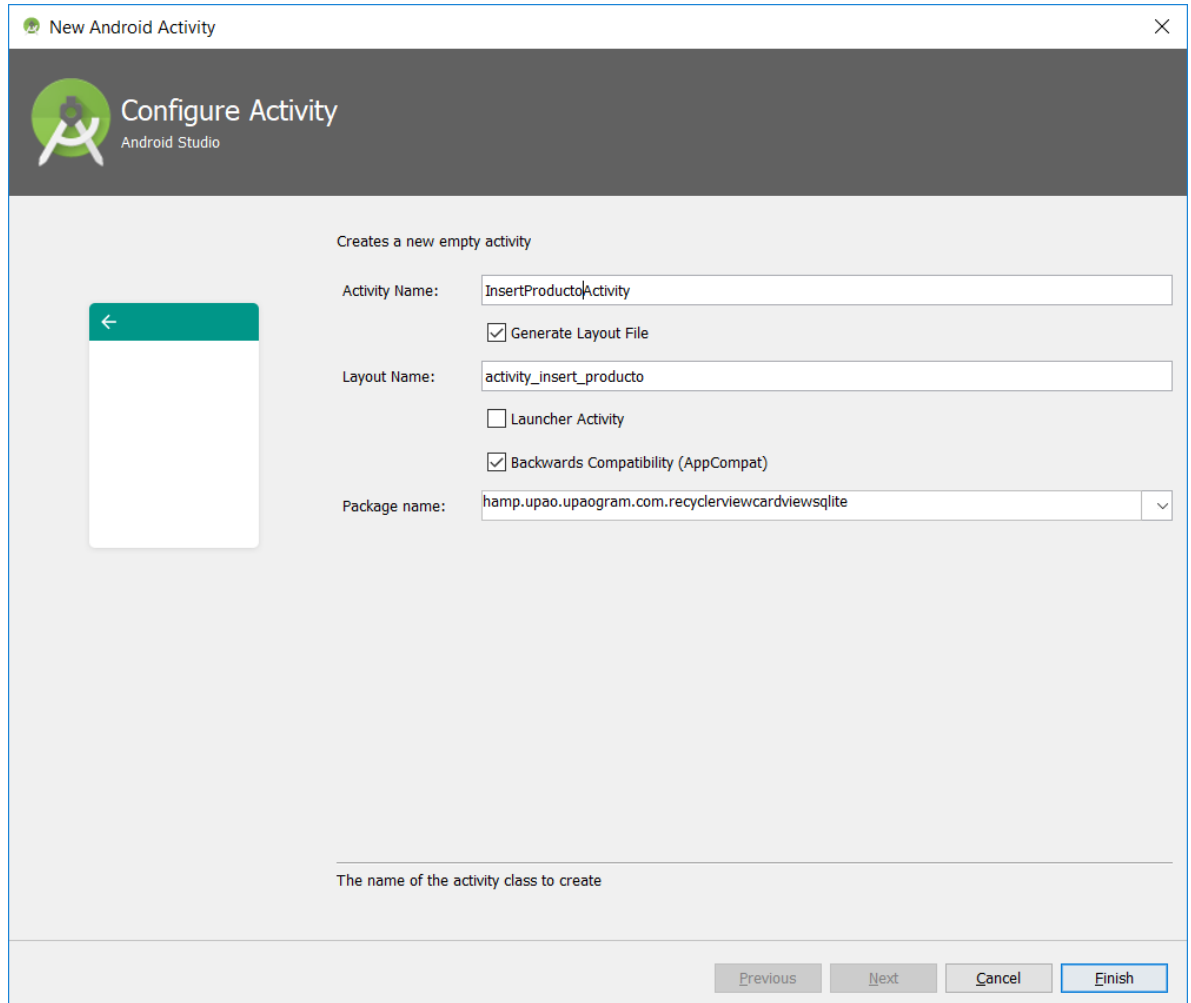
    <Button
        android:id="@+id/btnListProduct"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="List Product"
        android:onClick="onClick"/>

</LinearLayout>
```



Agregando Nuevo Activity InsertProductoActivity(Empty Activity)

9. Agregar activity tipo empty activity.



New Android Activity

Configure Activity
Android Studio

Creates a new empty activity

Activity Name: InsertProductoActivity

☒ Generate Layout File

Layout Name: activity_insert_producto

☐ Launcher Activity

☒ Backwards Compatibility (AppCompat)

Package name: hamp.upao.upaogram.com.recyclerviewcardviewsqllite

The name of the activity class to create

Previous Next Cancel Finish

Codigo XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"

    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"

    tools:context="hamp.upao.upaogram.com.recyclerviewcardviewsqlite.In
sertProductoActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="NEW PRODUCT"
        android:textSize="25sp"
        android:textStyle="bold"
        android:layout_marginBottom="30dp"/>

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="name"/>

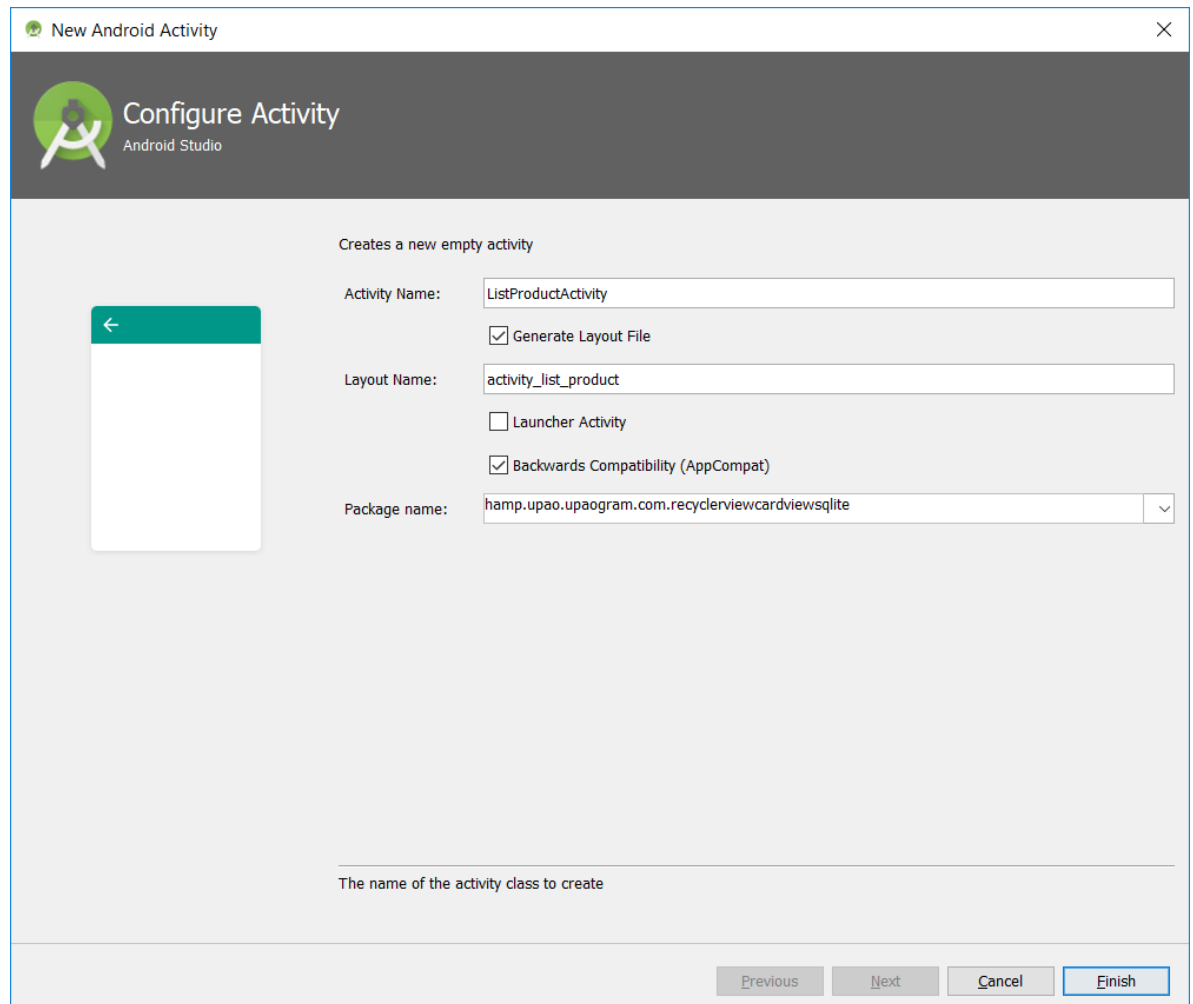
    <EditText
        android:id="@+id/quantity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:hint="quantity"/>

    <Button
        android:id="@+id/btnRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert"
        android:layout_marginTop="30dp"
        android:onClick="onClick"/>

</LinearLayout>
```

Agregando Nuevo Activity ListProductoActivity(Empty Activity)

10. Agregar activity tipo empty activity.



Antes de editar el XML, debe ir a Gradle Scripts y agregar en el archivo build.gradle (Module: app) en la sección de dependencias lo siguiente

```
compile 'com.android.support:cardview-v7:26+'  
compile 'com.android.support:recyclerview-v7:26+'
```

No olvide sincronizar para que los cambios se confirmen.

Ahora si editamos el archivo XML.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"

android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

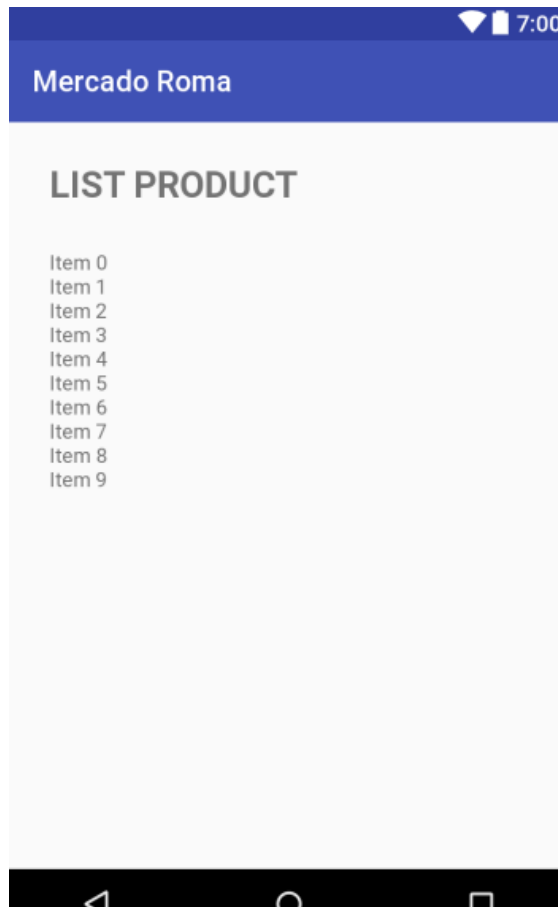
tools:context="hamp.upao.upaogram.com.recyclerviewcardviewsqllite.ListProd
uctActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="12dp">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="LIST PRODUCT"
            android:textSize="25sp"
            android:textStyle="bold"
            android:layout_marginBottom="30dp"/>

        <android.support.v7.widget.RecyclerView
            android:id="@+id/recyclerProducts"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </LinearLayout>
</RelativeLayout>
```



Agregando Codigo en MainActivity

11. Implementar el método onClick

```
public void onClick(View view) {  
    Intent miIntent=null;  
    switch (view.getId()) {  
        case R.id.btnCreateProduct:  
            miIntent = new Intent(MainActivity.this, InsertProductoActivity.class);  
            break;  
  
        case R.id.btnListProduct:  
            miIntent = new Intent(MainActivity.this, ListProductActivity.class);  
            break;  
    }  
  
    if (miIntent!=null){  
        startActivity(miIntent);  
    }  
}
```

12. Ejecuta la aplicación y verifica que al dar click en el botón insert product muestre el formulario insert product asi mismo del list product.

Agregando Código en InsertProductActivity

13. Implementa el método insertProduct

```
private void insertProduct() {  
    ConnectionSQLiteHelper conn=new ConnectionSQLiteHelper(this, "dbmercado", null, 1);  
    SQLiteDatabase db=conn.getWritableDatabase();  
    ContentValues values=new ContentValues();  
  
    values.put(Utilities.COLUMN_NAME, Name.getText().toString());  
    values.put(Utilities.COLUMN_QUANTITY, Quantity.getText().toString());  
  
    Long idResult=db.insert(Utilities.TABLE_PRODUCT, Utilities.COLUMN_ID_PRODUCT, values);  
  
    Toast.makeText(getApplicationContext(), "Id Product: "+idResult, Toast.LENGTH_SHORT).show();  
    db.close();  
}
```

14. Implementa el método Clear

```
public void Clear() {  
    Name.setText("");  
    Quantity.setText("");  
}
```

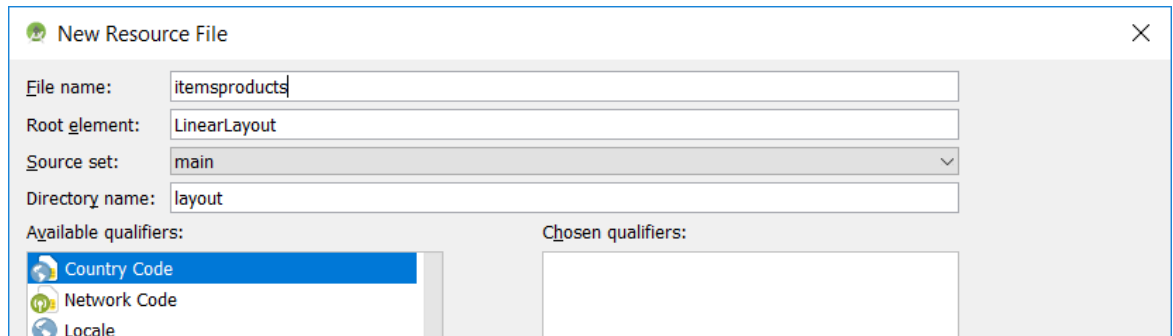
15. Invocar en el método onClick

```
public void onClick(View view) {  
    insertProduct();  
    Clear();  
}
```

16. Ejecuta la aplicación y realiza un registro, si todo está bien te debe mostrar un mensaje con el id del registro generado.

Agregando Recurso Layout itemsproducts

17. Selecciona la carpeta layout y agregue un Layotu resource file.



```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="1dp"
    card_view:cardElevation="2dp"
    card_view:cardUseCompatPadding="true">
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="12dp">
```

```
<ImageView
    android:id="@+id/product_image"
    android:layout_width="144dp"
    android:layout_height="164dp"
    android:layout_gravity="center"
    android:contentDescription="Description"
    android:src="@drawable/food"
    android:scaleType="centerCrop"/>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical">
```

```
<TextView
    android:id="@+id/product_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```

        android:layout_weight="8"
        android:paddingLeft="16dp"
        android:text="Id"
        android:textSize="12dp"
        android:textStyle="bold" />

<TextView
    android:id="@+id/product_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="8"
    android:paddingLeft="16dp"
    android:text="Name"
    android:textSize="12dp"
    android:textStyle="bold" />

<TextView
    android:id="@+id/product_quantity"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="8"
    android:paddingLeft="16dp"
    android:text="quantity"
    android:textSize="12dp"
    android:textStyle="bold" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="8"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/edit_product"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_weight="1"
        android:contentDescription="Description"
        android:src="@drawable/pedit" />

    <ImageView
        android:id="@+id/delete_product"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:contentDescription="Description"
        android:src="@drawable/pdelete" />

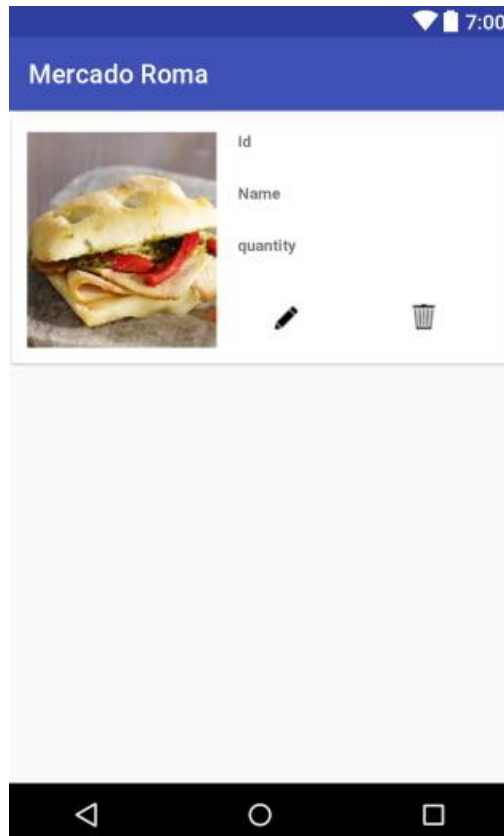
</LinearLayout>

</LinearLayout>

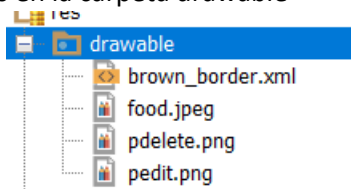
</LinearLayout>

</android.support.v7.widget.CardView>

```

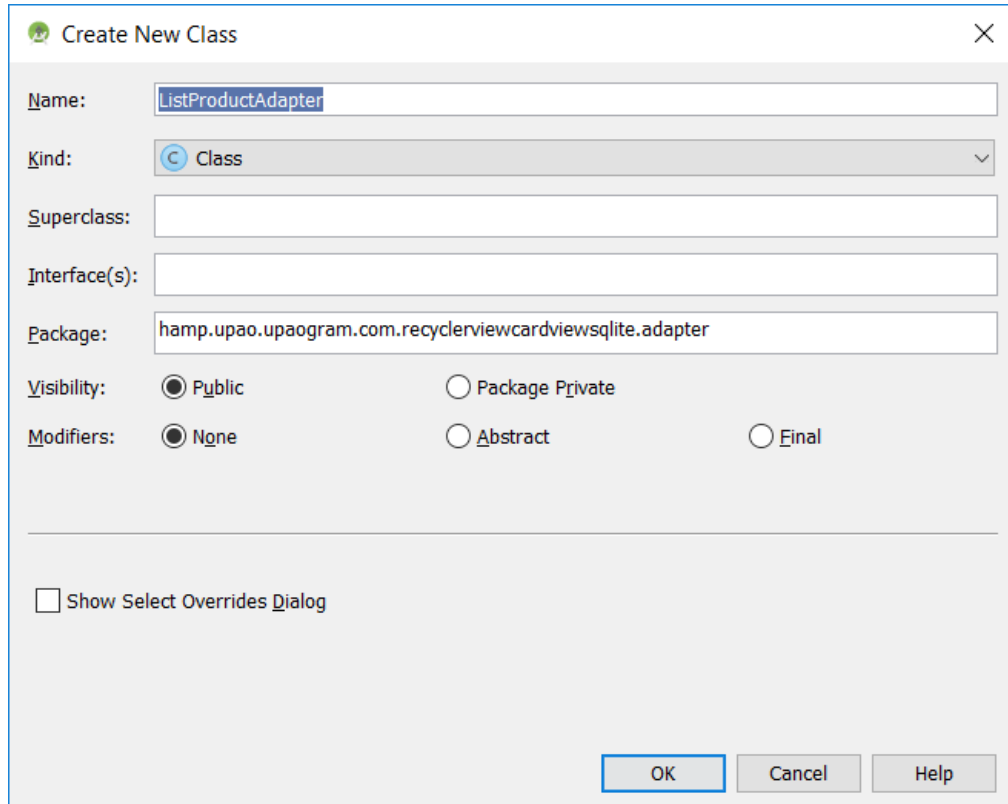


No olvide colocar las imagenes en la carpeta drawable



Creando Clase Adapter

18. Debe crear una clase ListProductAdapter en el paquete adapter.



The screenshot shows the 'Create New Class' dialog box. The fields are filled as follows:

- Name:** ListProductAdapter
- Kind:** Class
- Superclass:** (empty)
- Interface(s):** (empty)
- Package:** hamp.upao.upaogram.com.recyclerviewcardviewsqllite.adapter
- Visibility:** Public (selected)
- Modifiers:** None (selected)

At the bottom, there is a checkbox for 'Show Select Overrides Dialog' which is unchecked. The 'OK', 'Cancel', and 'Help' buttons are at the bottom right.

Implementar el siguiente código

```
public class ListProductAdapter extends RecyclerView.Adapter<ListProductAdapter.ProductsViewHolder> {  
  
    //Context context;  
    ArrayList<Product> lstProducts;  
  
    public ListProductAdapter(ArrayList<Product> lstProducts) {  
        //this.context=context;  
        this.lstProducts = lstProducts;  
    }  
  
    @Override  
    public ProductsViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.itemsproducts,parent,false);  
        return new ProductsViewHolder(view);  
    }  
}
```

```

@Override
public void onBindViewHolder(ProductsViewHolder holder, int position) {
    holder.id.setText(lstProducts.get(position).getId().toString());
    holder.name.setText(lstProducts.get(position).getName());
    holder.quantity.setText(lstProducts.get(position).getQuantity().toString());
    holder.image.setImageResource(R.drawable.food);
}

@Override
public int getItemCount() {
    return lstProducts.size();
}

public class ProductsViewHolder extends RecyclerView.ViewHolder {
    TextView name, quantity, id;
    ImageView image;
    ImageView deleteProduct;
    ImageView editProduct;

    public ProductsViewHolder(View itemView) {
        super(itemView);
        id = (TextView) itemView.findViewById(R.id.product_id);
        name = (TextView) itemView.findViewById(R.id.product_name);
        quantity = (TextView) itemView.findViewById(R.id.product_quantity);
        image = (ImageView) itemView.findViewById(R.id.product_image);

        deleteProduct = (ImageView) itemView.findViewById(R.id.delete_product);
        editProduct = (ImageView) itemView.findViewById(R.id.edit_product);
    }
}
}

```

Implementando Activity Clase ListProductsActivity

19. Implementar metodo consultListProducts

```

private void consultListProducts() {
    SQLiteDatabase db=conn.getReadableDatabase();

    Product product=null;

    Cursor cursor=db.rawQuery("SELECT * FROM "+ Utilities.TABLE_PRODUCT,null);

    while (cursor.moveToNext()) {
        product=new Product();
        product.setId(cursor.getInt(0));
        product.setName(cursor.getString(1));
        product.setQuantity(Integer.parseInt(cursor.getString(2)));

        lstProducts.add(product);
    }
}

```

20. Implementar metodo onCreate y declaracion de variables.

```
public class ListProductActivity extends AppCompatActivity {

    ArrayList<Product> listProducts;
    RecyclerView recyclerViewProducts;
    ConnectionSQLiteHelpder conn;
    ListProductAdapter adapter;

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

        conn=new ConnectionSQLiteHelpder(this,"dbmercado",null,1);

        listProducts=new ArrayList<>();

        recyclerViewProducts= (RecyclerView) findViewById(R.id.recyclerProducts);
        recyclerViewProducts.setLayoutManager(new LinearLayoutManager(this));

        consultListProducts();

        adapter=new ListProductAdapter(listProducts);
        recyclerViewProducts.setAdapter(adapter);
    }
}
```

21. Ejecuta la aplicación da click en el botón List Products y verifica que se muestra los productos registrados.

Implementar metodo eliminar

22. Método deleteProduct

```
public void deleteProduct(int id){
    String[] whereArgs = new String[] {String.valueOf(id)};
    SQLiteDatabase db=conn.getWritableDatabase();
    ContentValues values=new ContentValues();

    db.delete(Utilities.TABLE_PRODUCT,"id_product=?",whereArgs);

    db.close();
}
```

Implementar eliminar deslizando registro a la derecha

23. Debe ir dentro del método onCreate

```
ItemTouchHelper.SimpleCallback simpleCallback = new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.RIGHT) {  
  
    @Override  
    public boolean onMove(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder,  
                          RecyclerView.ViewHolder target) {  
        return false;  
    }  
  
    @Override  
    public void onSwiped(RecyclerView.ViewHolder viewHolder, int direction) {  
        int position = viewHolder.getAdapterPosition();  
        int id = listProducts.get(position).getId();  
        deleteProduct(id);  
        finish();  
        startActivity(getIntent());  
    }  
};  
  
ItemTouchHelper itemTouchHelper = new ItemTouchHelper(simpleCallback);  
itemTouchHelper.attachToRecyclerView(recyclerViewProducts);
```