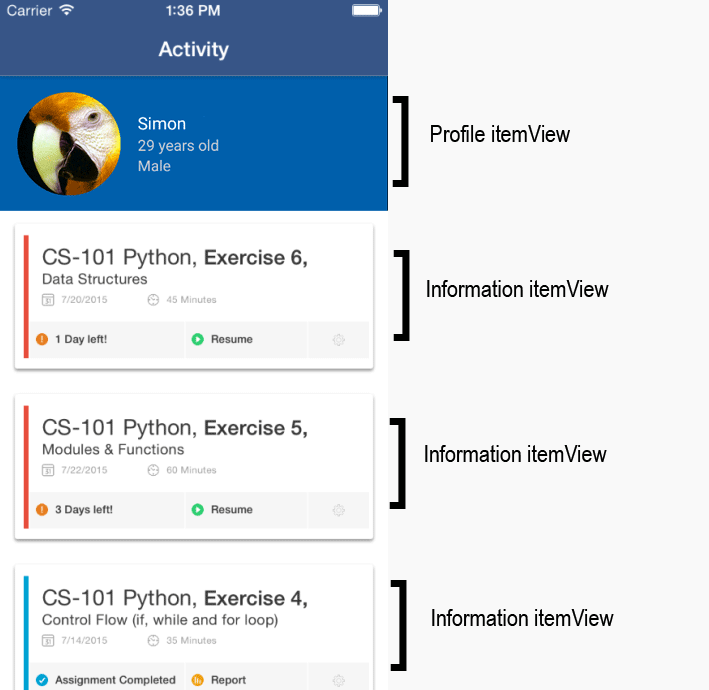
|  |  |
| --- | --- |
|  | **UNIVERSIDAD DON BOSCO**  **FACULTAD DE INGENIERÍA – ESCUELA DE COMPUTACIÓN** |
|  | TALLER ANDROID  **TEMA IV - Material Design**  RecyclerView - CardView |

**I Objetivos**

* Comprender el uso del RecyclerView para listas.
* Utilizar View.Adapter para crear diseños de celdas de listas.

**II Introducción**



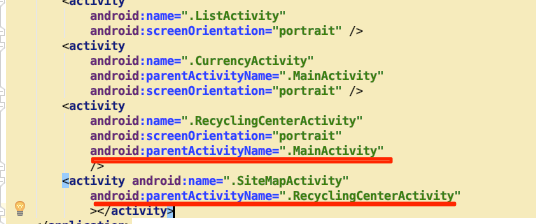
RecyclerView es oficialmente el elemento que reemplaza al ListView. Su implementación trata de aportar mejor rendimiento y manejo de grandes conjuntos de datos. Además, permite crear diseños llamativos. Incluye muchas característica, incluyendo gestos, respuesta a eventos, entre otros.

**III Desarrollo**

1. **Crear dos actividades:**

* RecyclingCenterActivity
* SiteMapActivity

Agregaremos a estas actividades la propiedad de parentActivityName para facilitar la navegación:



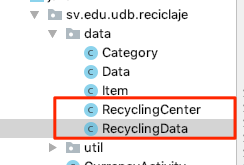
1. **Agregar al Gradle Module:App la dependencia RecyclerView:**

|  |
| --- |
| */\*Librería RecyclerView\*/* implementation **'com.android.support:recyclerview-v7:28.0.0'** |

1. **Para este ejercicio, es necesario que usted edite la actividad de inicio para lanzar RecyclingCenterActivity**



1. **Dentro del paquete data agregaremos un POJO RecyclingCenter.java y una clase denominada RecyclingData.java:**



1. **RecyclingCenter.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** java.io.Serializable;  **public class** RecyclingCenter **implements** Serializable {  **private** String **name**;  **private** String **address**;  **private** String **imageSiteUrl**;  **private** String **email**;  **private double latitude**;  **private double longitude**;   **public** String getName() {  **return name**;  }   **public void** setName(String name) {  **this**.**name** = name;  }   **public** String getAddress() {  **return address**;  }   **public void** setAddress(String address) {  **this**.**address** = address;  }   **public** String getImageSiteUrl() {  **return imageSiteUrl**;  }   **public void** setImageSiteUrl(String imageSiteUrl) {  **this**.**imageSiteUrl** = imageSiteUrl;  }   **public double** getLatitude() {  **return latitude**;  }   **public void** setLatitude(**double** latitude) {  **this**.**latitude** = latitude;  }   **public double** getLongitude() {  **return longitude**;  }   **public void** setLongitude(**double** longitude) {  **this**.**longitude** = longitude;  }   **public** String getEmail() {  **return email**;  }   **public void** setEmail(String email) {  **this**.**email** = email;  } } |

Ahora agregaremos la clase que generará la información correspondiente:

1. **RecyclingData.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje.data;  **import** java.util.ArrayList; **import** java.util.List;  **public class** RecyclingData {   **private** List<RecyclingCenter> **recyclingCenterList**;   **public** RecyclingData(){   **recyclingCenterList** = **new** ArrayList<>();   RecyclingCenter a = **new** RecyclingCenter();  a.setName(**"Salva Metal"**);  a.setAddress(**"BLVD. DEL EJERCITO #2,500 POLIG B, LOTE 9 URB. LOURDES, SAN SALVADOR, SAN SALVADOR."**);  a.setImageSiteUrl(**"http://thesystemroot.net/imagenes/centro-reciclado.jpg"**);  a.setLatitude(13.7436472);  a.setLongitude(-89.195672);   **recyclingCenterList**.add(a);   RecyclingCenter b = **new** RecyclingCenter();  b.setName(**"Recicladora Moderna"**);  b.setAddress(**"9na Calle Oriente, SAN SALVADOR, SAN SALVADOR."**);  b.setImageSiteUrl(**"http://www.santiagorecicla.cl/wp-content/uploads/2017/08/New-Recycle-660.jpg"**);  b.setLatitude(13.7447819);  b.setLongitude(-89.2208932);   **recyclingCenterList**.add(b);    RecyclingCenter c = **new** RecyclingCenter();  c.setName(**"Parque Industrial Verde"**);  c.setAddress(**"Parque Industrial Verde, Calle Agua Caliente, Km 5, Soyapango, San Salvador"**);  c.setImageSiteUrl(**"http://www.reciclaelsalvador.com/wp-content/uploads/2016/08/PIV\_Mockup-IMG-36.jpg"**);  c.setLatitude(13.7051254);  c.setLongitude(-89.1556134);   **recyclingCenterList**.add(c);     }    **public** List<RecyclingCenter> getRecyclingCenterList() {  **return recyclingCenterList**;  }   **public void** setRecyclingCenterList(List<RecyclingCenter> recyclingCenterList) {  **this**.**recyclingCenterList** = recyclingCenterList;  } } |

Lo que deseamos hacer es crear una lista que pueda desplazarse horizontalmente y muestre los sitios de reciclaje. Esto se logra gracias a un RecyclerView, el cual diseñaremos desde el layout contenedor de cada item para ser “inflado” y mostrado.



1. **Crearemos un layout nuevo sin asociar a ninguna actividad:**

item.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_height="wrap\_content"  android:layout\_width="match\_parent"  xmlns:app="http://schemas.android.com/apk/res-auto"  app:cardCornerRadius="10dp"  app:cardElevation="10dp"  app:contentPadding="5dp"  android:layout\_margin="10dp"** >   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="Nombre sitio"  android:id="@+id/centername"  android:textSize="30dp"  android:textAlignment="center"** />  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="210dp"  android:layout\_below="@+id/centername"  android:orientation="vertical"** >  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textAlignment="center"  android:text=""  android:id="@+id/centeraddress"** />  <**ImageView  android:layout\_width="match\_parent"  android:layout\_height="200dp"  android:id="@+id/centerimage"** />  </**LinearLayout**>  </**RelativeLayout**>  </**android.support.v7.widget.CardView**> |

1. **Ahora crearemos un paquete denominado util. Dentro de él, crearemos una clase adapter para “inflar” el contenido dentro del RecyclerView**

|  |
| --- |
|  |

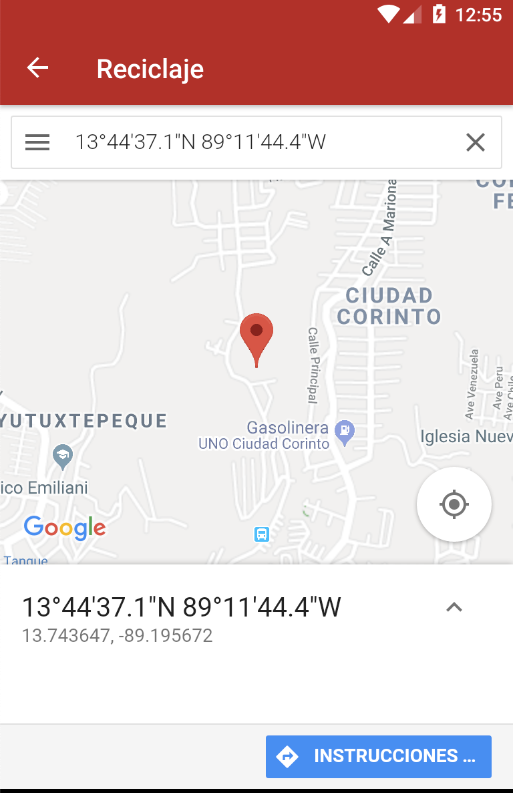
RecyclingCenterAdapter.java

|  |
| --- |
| **package** sv.edu.udb.reciclaje.util;  **import** android.content.Context; **import** android.content.Intent; **import** android.support.annotation.NonNull; **import** android.support.v7.widget.RecyclerView; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.ImageView; **import** android.widget.TextView;  **import** com.squareup.picasso.Picasso;  **import** java.util.List;  **import** sv.edu.udb.reciclaje.R; **import** sv.edu.udb.reciclaje.SiteMapActivity; **import** sv.edu.udb.reciclaje.data.RecyclingCenter;  **public class** RecyclingCenterAdapter **extends** RecyclerView.Adapter<RecyclingCenterAdapter.ViewHolder> {   List<RecyclingCenter> **sites**;   Context **context**;   **public** RecyclingCenterAdapter(List<RecyclingCenter> sites, Context context){  **this**.**context** = context;  **this**.**sites** = sites;  }   @NonNull  @Override  **public** ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, **int** i) {  Context context = viewGroup.getContext();  LayoutInflater inflater = LayoutInflater.*from*(context);   View siteView = inflater.inflate(R.layout.***item***,viewGroup,**false**);   ViewHolder viewHolder = **new** ViewHolder(siteView);     **return** viewHolder;  }   @Override  **public void** onBindViewHolder(@NonNull ViewHolder viewHolder, **final int** i) {  RecyclingCenter recyclingCenter = **sites**.get(i);  viewHolder.**name**.setText(recyclingCenter.getName());  viewHolder.**addess**.setText(recyclingCenter.getAddress());  Picasso.*get*().load(recyclingCenter.getImageSiteUrl()).into(viewHolder.**image**);  viewHolder.**itemView**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  Intent intent = **new** Intent(**context**, SiteMapActivity.**class**);  intent.putExtra(**"center"**, **sites**.get(i));  **context**.startActivity(intent);  }  });  }   @Override  **public int** getItemCount() {  **return sites**.size();  }   **public class** ViewHolder **extends** RecyclerView.ViewHolder {  TextView **name**;  TextView **addess**;  ImageView **image**;    **public** ViewHolder(@NonNull View itemView) {  **super**(itemView);  **name** = itemView.findViewById(R.id.***centername***);  **addess** = itemView.findViewById(R.id.***centeraddress***);  **image** = itemView.findViewById(R.id.***centerimage***);   }  }  } |

1. **Vamos a editar el RecyclingCenterActivity.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.support.v7.widget.LinearLayoutManager; **import** android.support.v7.widget.RecyclerView;  **import** sv.edu.udb.reciclaje.data.RecyclingCenter; **import** sv.edu.udb.reciclaje.data.RecyclingData; **import** sv.edu.udb.reciclaje.util.RecyclingCenterAdapter;  **public class** RecyclingCenterActivity **extends** AppCompatActivity{   RecyclerView **recyclerView**;   RecyclingData **recyclingData** = **new** RecyclingData();   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);    setContentView(R.layout.***activity\_recycling\_center***);   **recyclerView** = findViewById(R.id.***recyclerview***);   **recyclerView**.setLayoutManager(**new** LinearLayoutManager(**this**));   RecyclingCenterAdapter adapter = **new** RecyclingCenterAdapter(**recyclingData**.getRecyclingCenterList(), **this**);   **recyclerView**.setAdapter(adapter);   }  } |

Al seleccionar un item de la lista mostraremos el mapa de la ubicación del centro de reciclaje. Utilizaremos un elementos denominado WebView, con la intención de cargar el sitio de maps (Nota: esta no es la api de mapas de google).



1. **Editaremos el activity\_site\_map.xml**

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**android.support.constraint.ConstraintLayout 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=".SiteMapActivity"**>   <**WebView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/webview"** >   </**WebView**>  </**android.support.constraint.ConstraintLayout**> |

1. **Finalmente editaremos el SiteMapActivity.java**

|  |
| --- |
| **package** sv.edu.udb.reciclaje;  **import** android.support.v7.app.ActionBar; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.webkit.WebView;  **import** sv.edu.udb.reciclaje.data.RecyclingCenter;  **public class** SiteMapActivity **extends** AppCompatActivity {    String **URL\_MAP** = **"https://www.google.com/maps/search/?api=1&query="**;   **private** WebView **webView**;    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_site\_map***);   RecyclingCenter center = (RecyclingCenter) getIntent().getSerializableExtra(**"center"**);    **URL\_MAP** += center.getLatitude()+ **","** + center.getLongitude();   **webView** = (WebView) findViewById(R.id.***webview***);  **webView**.getSettings().setJavaScriptEnabled(**true**);    **webView**.loadUrl(**URL\_MAP**);  } } |

**IV Ejercicio complementario**

|  |
| --- |
| 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). |

Reemplace la lista de productos de ejercicio de la clase III, ahora debe implementar un RecyclerView. En la lista de elementos debe mostrar el id, el nombre y el precio del producto. Posteriormente al seleccionar uno de los items, debe desplegar la actividad que previamente había creado.