



Defensa Hito 4 - MY PLAYLIST APP

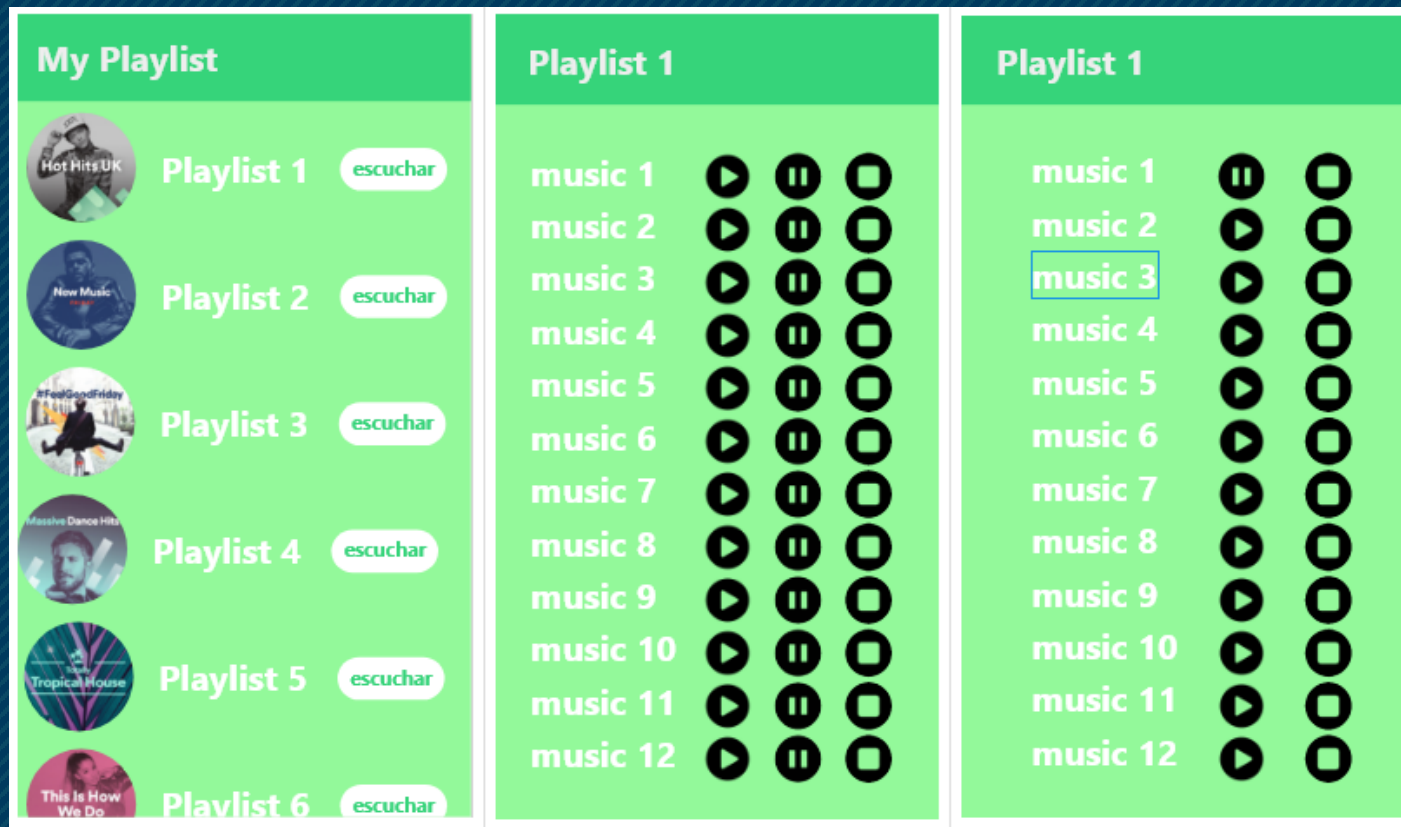
Estudiante: Jhelson Carrillo Ortega

MY PLAYLIST APP

El objetivo es crear un PLAYLIST en donde se listara una lista de álbumes, en donde cada ALBUM contiene una lista de canciones. Para poder generar este listado se deberá de utilizar un RecyclerView.

Cada ALBUM contiene cierta cantidad de canciones, cada canción tiene disponibles 3 OPCIONES. PLAY - PAUSE - STOP. De manera similar para generar este listado se debe utilizar el concepto de RecyclerView.

Para el desarrollo de esta APLICACION se tiene los siguientes SCREENSHOTS.



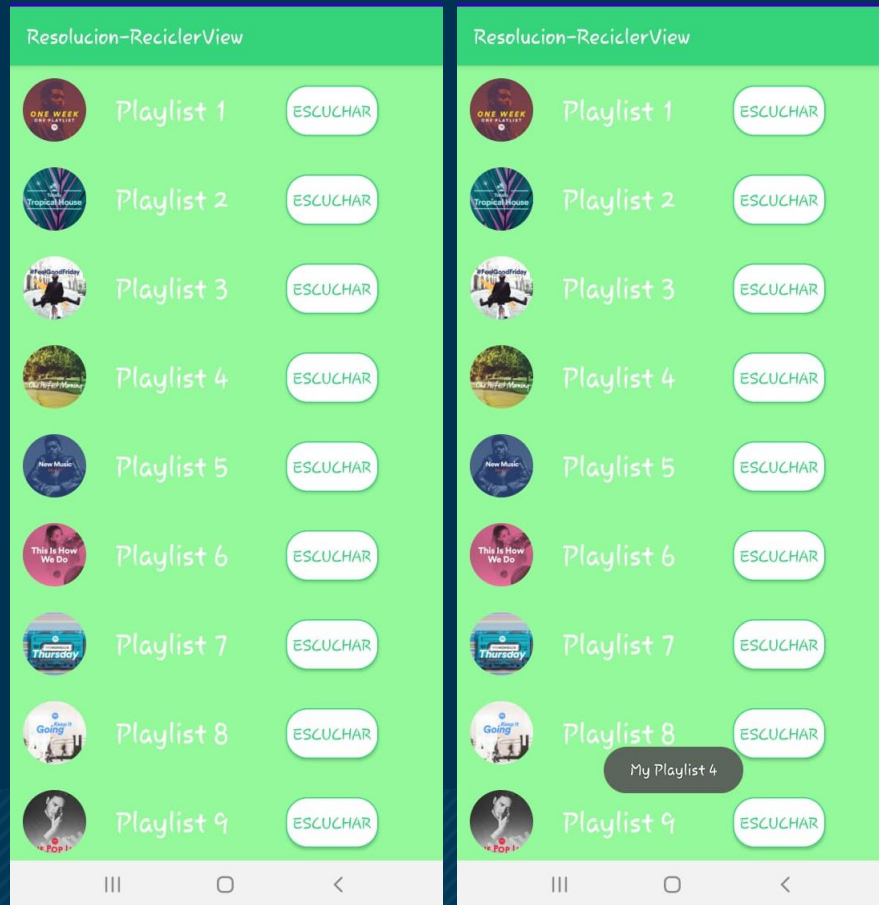
1ra Pregunta

2ra Pregunta

3ra Pregunta

Pregunta 1:

- Cuando se presiona en el TEXT(Ejem Playlist 5) debe de mostrar un mensaje(TOAST), una descripcion acerca de ese ALBUM(Ejem. "My Playlist 5").
- Cuando se presiona en el BUTTON(escuchar) debe de redirigir a un nuevo activity, en donde se mostrara nuevamente un RecyclerView con el listado de las canciones que tiene ese ALBUM.



Se logró desarrollar el código necesario para mostrar hasta este punto en la App.

Código .java del MainActivity:

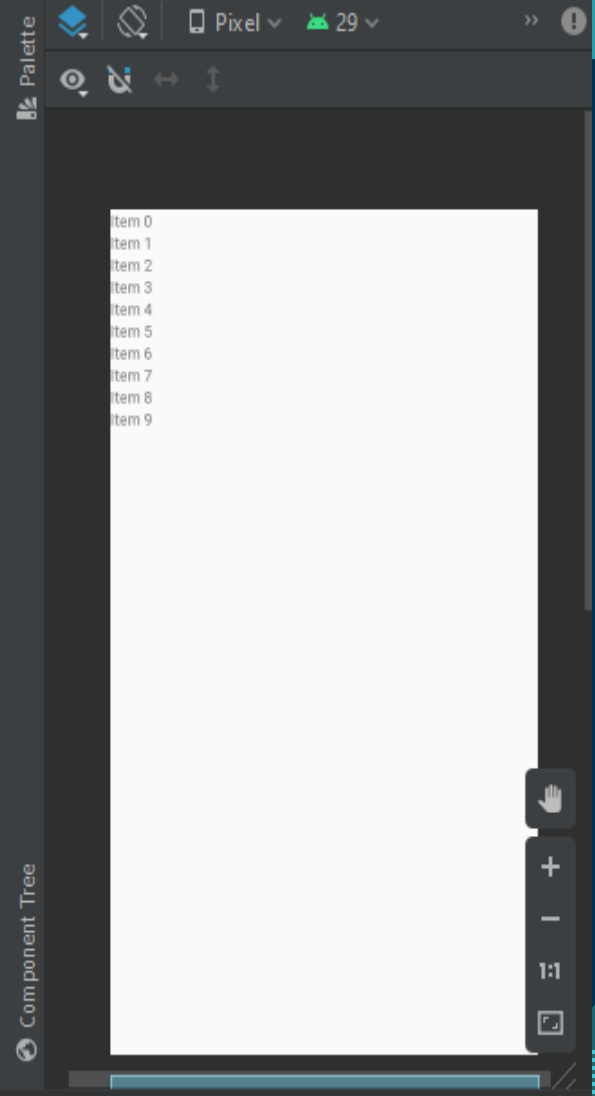
```
18 public class MainActivity extends AppCompatActivity
19 {
20     private RecyclerView recyclerView;
21     private RVAdapter rvAdapter;
22     private ArrayList<String> imagesName = new ArrayList<>();
23     private ArrayList<String> imagesURL = new ArrayList<>();
24     private final String imUR="V40gA40,yB3d2qH,9jMbaX2,rYndmdq,sypYnSP,HBeK1ot,YCqbt8r,eLk31XX," +
25         "4ZHp7F0,XzffKgy,DJabk5C";
26     private final String imName="Playlist 1,Playlist 2,Playlist 3,Playlist 4,Playlist 5,Playlist 6," +
27         "Playlist 7,Playlist 8,Playlist 9,Playlist 10,Playlist 11";
28     @Override
29     protected void onCreate(Bundle savedInstanceState)
30     {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_main);
33
34         initImageBitMaps();
35         initRecyclerView();
36     }
```

Código .java del MainActivity:

```
37
38     public void initImageBitMaps()
39     {
40
41         imagesURL.addAll(Arrays.asList(imUR.split( regex: ",")));
42         imagesName.addAll(Arrays.asList(imName.split( regex: ",")));
43     }
44
45     public void initRecyclerView()
46     {
47         recyclerView = findViewById(R.id.rvContainer);
48         rvAdapter = new RVAdapter( context: this, imagesName, imagesURL);
49         recyclerView.setAdapter(rvAdapter);
50         recyclerView.setLayoutManager(new LinearLayoutManager( context: this));
51     }
52 }
```

Código .xml del activity_main:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9
10     <androidx.recyclerview.widget.RecyclerView
11         android:id="@+id/rvContainer"
12         android:layout_width="match_parent"
13         android:layout_height="match_parent">
14     </androidx.recyclerview.widget.RecyclerView>
15
16
17 </RelativeLayout>
```



Código .xml de los string necesarios:

```
1 <resources>
2   <string name="app_name">Resolucion-ReciclerView</string>
3   <string name="canada">Canada</string>
4   <string name="Escuchar">escuchar</string>
5 </resources>
6 |
```

Código .xml para el estilo de los botones:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android">
3     <solid android:color="@color/colorWhite"/>
4     <corners android:radius="20dp"/>
5     <stroke android:width="1dp" android:color="@color/colorLetra"/>
6 </shape>
```

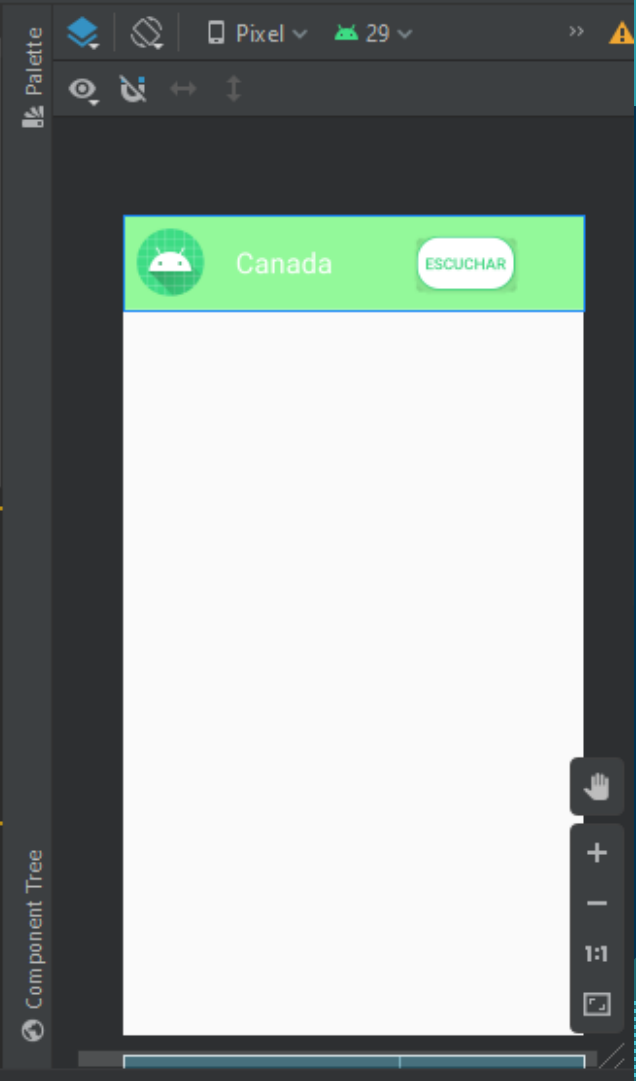


Código .xml para los colores utilizados:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="colorPrimary">#6200EE</color>
4      <color name="colorPrimaryDark">#3700B3</color>
5      <color name="colorAccent">#03DAC5</color>
6      <color name="colorWhite">#FFFFFF</color>
7      <color name="colorLetra">#36D47A</color>
8      <color name="color2">#93F99A</color>
9  </resources>
```

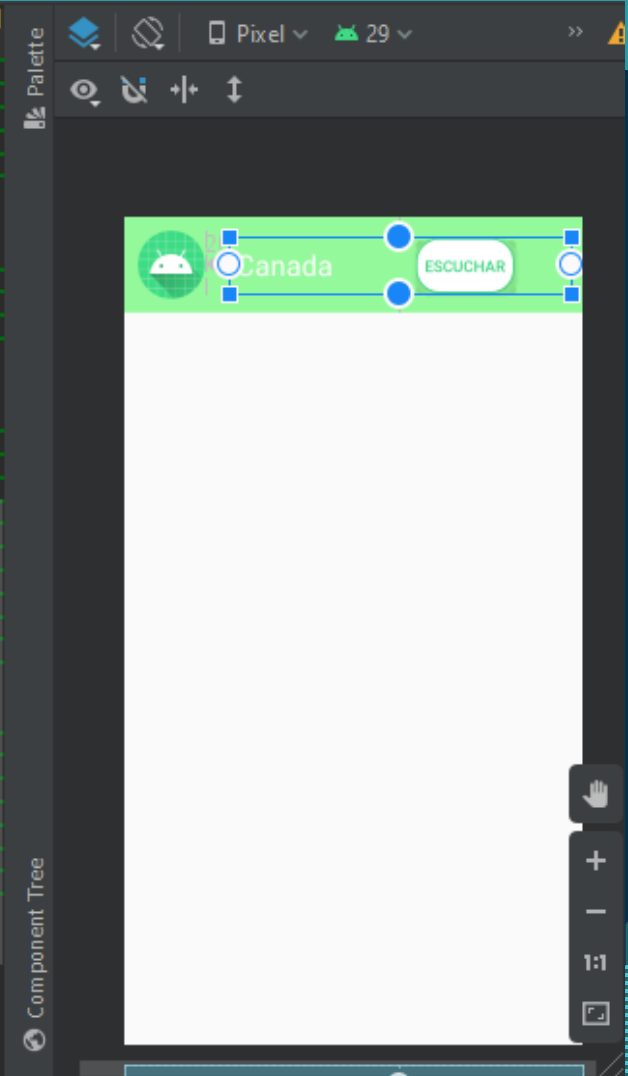
Código .xml del rl_list_item:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/rlListItem"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:background="@color/color2"
7     android:padding="12dp">
8
9     <de.hdodenhof.circleimageview.CircleImageView
10
11         android:id="@+id/civItem"
12         android:layout_width="60dp"
13         android:layout_height="60dp"
14         android:src="@mipmap/ic_launcher_round"
15     />
16
17     <TextView
18         android:id="@+id/tvImage"
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
```



Código .xml del rl_list_item:

```
21 android:text="Canada"
22 android:layout_toRightOf="@id/civItem"
23 android:layout_centerVertical="true"
24 android:padding="8dp"
25 android:textColor="@color/colorWhite"
26 android:textSize="25sp"
27 android:layout_marginStart="20dp"
28 android:clickable="true">
29 </TextView>
30 <Button
31     android:id="@+id/skip1"
32     android:layout_width="wrap_content"
33     android:layout_marginTop="7dp"
34     android:layout_height="wrap_content"
35     android:layout_marginLeft="250dp"
36     android:background="@drawable/stylesheet"
37     android:text="escuchar"
38     android:textColor="@color/colorLetra">
39 </Button>
40 </RelativeLayout>
```



Código .java del ViewHolder:

```
18 public class ViewHolder extends RecyclerView.ViewHolder
19 {
20
21     private RelativeLayout rlContainer;
22     private CircleImageView circleImage;
23     private TextView tvImage;
24     private Button escu;
25     public ViewHolder( View itemView)
26     {
27         super(itemView);
28         initComponents(itemView);
29     }
30
31     @ private void initComponents(View itemView)
32     {
33         rlContainer= itemView.findViewById(R.id.rllistItem);
34         circleImage= itemView.findViewById(R.id.civItem);
35         tvImage= itemView.findViewById(R.id.tvImage);
36         escu=itemView.findViewById(R.id.skip1);
37     }
```

Código .java del ViewHolder:

```
38
39     public RelativeLayout getRlContainer()
40     {
41         return rlContainer;
42     }
43
44     public CircleImageView getCircleImage()
45     {
46         return circleImage;
47     }
48
49     public TextView getTvImage()
50     {
51         return tvImage;
52     }
53     public Button getEscu()
54     {
55         return escu;
56     }
57
58 }
```

Código .java del RVAdapter:

```
21 public class RVAdapter extends RecyclerView.Adapter<ViewHolder>
22 {
23     private static final String TAG = "RVAdapter";
24     private Context context;
25     private ArrayList<String> imageNames= new ArrayList<>();
26     private ArrayList<String> imagesURI= new ArrayList<>();
27
28     public RVAdapter(Context context, ArrayList<String> imageNames, ArrayList<String> imagesURI)
29     {
30         this.context=context;
31         this.imageNames=imageNames;
32         this.imagesURI=imagesURI;
33     }
34
35     @Override
36     public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType)
37     {
38         View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.rl_list_item,viewGroup, attachToRoot: false);
39         ViewHolder vHolder = new ViewHolder(view);
40         return vHolder;
41     }
```

Código .java del RVAdapter:

```
43      @Override
44      public void onBindViewHolder(ViewHolder holder, final int position)
45      {
46          String URI = "https://i.imgur.com/";
47          Glide.with(context).requestManager
48              .asBitmap().requestBuilder<Bitmap>
49              .load(URI + imagesURI.get(position) + ".png").requestBuilder<Bitmap>
50              .into(holder.getCircleImage());
51          holder.getTvImage().setText(imageNames.get(position));
52          holder.getTvImage().setOnClickListener(new View.OnClickListener()
53          {
54              @Override
55              public void onClick(View v)
56              {
57                  Toast.makeText(context, text: "My " + imageNames.get(position), Toast.LENGTH_SHORT).show();
58              }
59          });
```

Código .java del RVAdapter:

```
60     holder.getEscu().setOnClickListener(new View.OnClickListener()
61     {
62         @Override
63         public void onClick(View v)
64         {
65
66             Intent in = new Intent(context,Playlist_1.class);
67             context.startActivity(in);
68         }
69     });
70 }
71
72 @Override
73 public int getItemCount()
74 {
75     return imageNames.size();
76 }
77 }
```


“

Hasta este punto se tiene el diseño de la primera vista de la aplicación:

Resolucion-ReciclerView



Resolucion-ReciclerView



Pregunta 2:

- Mostrar el listado de todas las canciones que forman parte de ese ALBUM.
- Cada cancion tiene 3 opciones(PLAY, PAUSE y STOP)
- Cuando se presiona PLAY mostrar el TOAST("Play music").
- Cuando se presiona PAUSE mostrar el TOAST("Pause music").
- Cuando se presiona STOP mostrar el TOAST("Stop music").
- Importante, esta opciones **no son BUTTONS**. Debe de utilizar IMAGEVIEW.

- Se logró desarrollar todo el código y diseño de acuerdo a lo solicitado.
- A continuación en las siguientes diapositivas se mostrará todo el código para llegar a desarrollar estas la vista.



Código .java de Playlist_1:

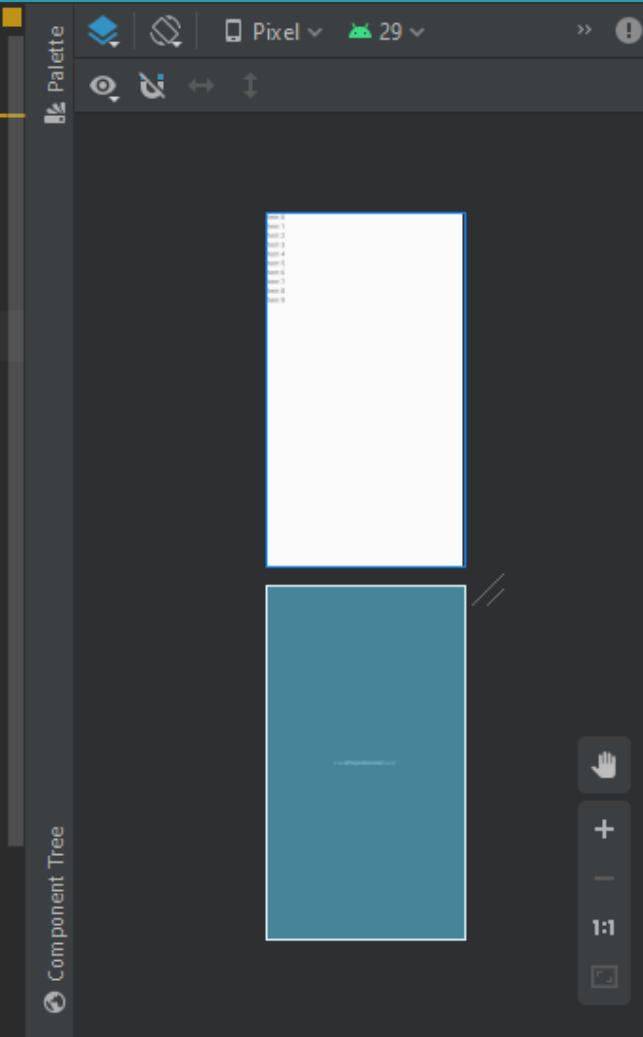
```
16 public class Playlist_1 extends AppCompatActivity
17 {
18     private RecyclerView recyclerView2;
19     private RVAdapter2 rvAdapter2;
20     private ArrayList<String> imagesName1 = new ArrayList<>();
21     private ArrayList<String> imagesURL1 = new ArrayList<>();
22     private ArrayList<String> imagesURL2 = new ArrayList<>();
23     private ArrayList<String> imagesURL3 = new ArrayList<>();
24     private final String imUR1="59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq";
25     private final String imUR2="2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ";
26     private final String imUR3="LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe,LjjdlQe";
27     private final String imName1="Music 1,Music 2,Music 3,Music 4,Music 5,Music 6,Music 7,Music 8,Music 9,Music 10,Music 11";
28     @Override
29     protected void onCreate(Bundle savedInstanceState)
30     {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_playlist_1);
33         initImageBitMaps();
34         initRecyclerView();
35     }
```

Código .java de Playlist_1:

```
36     public void initImageBitMaps()  
37     {  
38  
39         imagesURL1.addAll(Arrays.asList(imUR1.split( regex: ",")));  
40         imagesURL2.addAll(Arrays.asList(imUR2.split( regex: ",")));  
41         imagesURL3.addAll(Arrays.asList(imUR3.split( regex: ",")));  
42         imagesName1.addAll(Arrays.asList(imName1.split( regex: ",")));  
43     }  
44  
45     public void initRecyclerView()  
46     {  
47         recyclerView2 = findViewById(R.id.rvContainer2);  
48         rvAdapter2 = new RVAdapter2( context1: this, imagesName1, imagesURL1,imagesURL2,imagesURL3);  
49         recyclerView2.setAdapter(rvAdapter2);  
50         recyclerView2.setLayoutManager(new LinearLayoutManager( context: this));  
51     }  
52 }
```

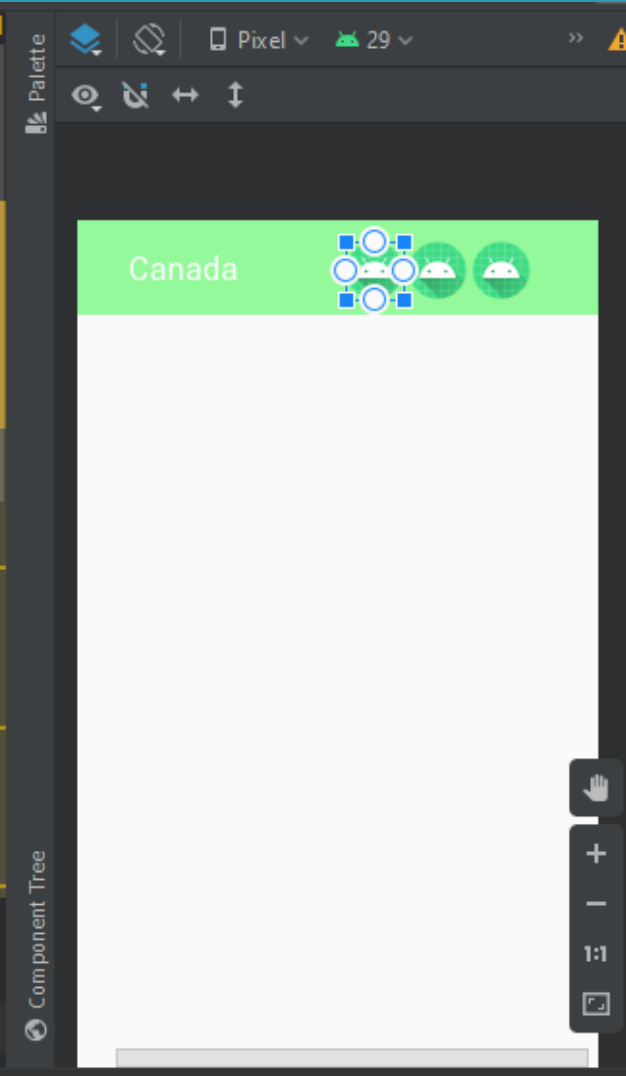
Código .xml del Playlist_1:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".App.Playlist_1">
8
9      <androidx.recyclerview.widget.RecyclerView
10         android:id="@+id/rvContainer2"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent">
13     </androidx.recyclerview.widget.RecyclerView>
14     <androidx.viewpager.widget.ViewPager
15         android:id="@+id/viewPagerContainer2"
16         android:layout_width="match_parent"
17         android:layout_height="match_parent">
18     </androidx.viewpager.widget.ViewPager>
19
20 </RelativeLayout>
```



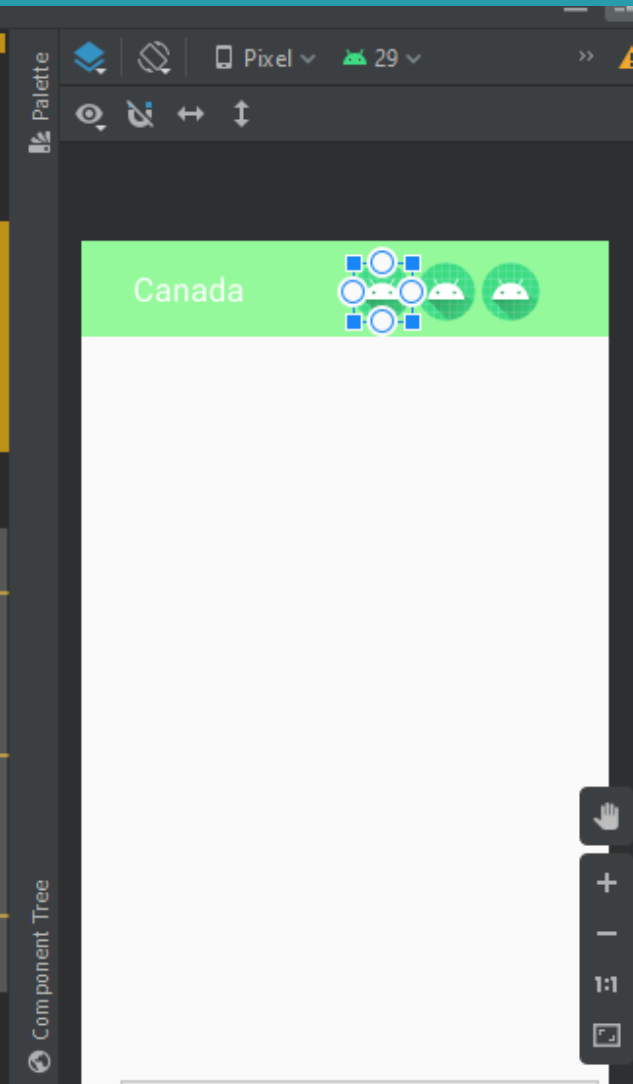
Código .xml del rl_list_item2:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/rlListItem2"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:background="@color/color2"
7     android:padding="12dp">
8     <TextView
9         android:id="@+id/tvImage"
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content"
12        android:text="Canada"
13        android:layout_centerVertical="true"
14        android:padding="8dp"
15        android:textColor="@color/colorWhite"
16        android:textSize="25sp"
17        android:layout_marginStart="20dp"
18        android:clickable="true"></TextView>
19     <de.hdodenhof.circleimageview.CircleImageView
20         android:id="@+id/civItem1"
```



Código .xml del rl_list_item2:

```
21     android:layout_width="45dp"
22     android:layout_height="45dp"
23     android:src="@mipmap/ic_launcher_round"
24     android:layout_marginLeft="200dp"
25     android:layout_marginTop="5dp"/>
26     <de.hdodenhof.circleimageview.CircleImageView
27     android:id="@+id/civItem2"
28     android:layout_width="45dp"
29     android:layout_height="45dp"
30     android:src="@mipmap/ic_launcher_round"
31     android:layout_marginLeft="250dp"
32     android:layout_marginTop="5dp"/>
33     <de.hdodenhof.circleimageview.CircleImageView
34     android:id="@+id/civItem3"
35     android:layout_width="45dp"
36     android:layout_height="45dp"
37     android:src="@mipmap/ic_launcher_round"
38     android:layout_marginLeft="300dp"
39     android:layout_marginTop="5dp" />
40 </RelativeLayout>
```



Código .java del ViewHolder2:

```
18 public class ViewHolder2 extends RecyclerView.ViewHolder {
19
20     private RelativeLayout rlContainer;
21     private CircleImageView circleImage,circleImage2,circleImage3,circleImage4;
22     private TextView tvImage;
23
24     public ViewHolder2( View itemView)
25     {
26         super(itemView);
27         initComponents(itemView);
28     }
29
30     @private void initComponents(View itemView)
31     {
32         rlContainer= itemView.findViewById(R.id.rllistItem2);
33         circleImage= itemView.findViewById(R.id.civItem1);
34         circleImage2= itemView.findViewById(R.id.civItem2);
35         circleImage3= itemView.findViewById(R.id.civItem3);
36         tvImage= itemView.findViewById(R.id.tvImage);
37     }
```


Código .java del ViewHolder2:

```
39     public RelativeLayout getRlContainer()
40     {
41         return rlContainer;
42     }
43     public CircleImageView getCircleImage()
44     {
45         return circleImage;
46     }
47     public CircleImageView getCircleImage2()
48     {
49         return circleImage2;
50     }
51     public CircleImageView getCircleImage3()
52     {
53         return circleImage3;
54     }
55     public TextView getTvImage()
56     {
57         return tvImage;
58     }
59 }
```

Código .java del RVAdapter2:

```
17 public class RVAdapter2 extends RecyclerView.Adapter<ViewHolder2>
18 {
19     private static final String TAG = "RVAdapter2";
20     private Context context1;
21     private ArrayList<String> imageNames1= new ArrayList<>();
22     private ArrayList<String> imagesURI1= new ArrayList<>();
23     private ArrayList<String> imagesURI2= new ArrayList<>();
24     private ArrayList<String> imagesURI3= new ArrayList<>();
25     public RVAdapter2(Context context1, ArrayList<String> imageNames1, ArrayList<String> imagesURI1 ,
26         ArrayList<String> imagesURI2,ArrayList<String> imagesURI3 )
27     {
28         this.context1=context1;
29         this.imageNames1=imageNames1;
30         this.imagesURI1=imagesURI1;
31         this.imagesURI2=imagesURI2;
32         this.imagesURI3=imagesURI3;
33     }
```

Código .java del RVAdapter2:

```
35  @Override
36  @ public ViewHolder2 onCreateViewHolder(ViewGroup viewGroup, int viewType){
37      View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.rl_list_item2,viewGroup, attachToRoot: false);
38      ViewHolder2 vHolder = new ViewHolder2(view);
39      return vHolder;
40  }
41
42  @Override
43  @ public void onBindViewHolder(final ViewHolder2 holder, final int position)
44  {
45      String URI = "https://i.imgur.com/";
46      Glide.with(context1) RequestManager
47          .asBitmap() RequestBuilder<Bitmap>
48          .load(URI + imagesURI1.get(position)+ ".png") RequestBuilder<Bitmap>
49          .into(holder.getCircleImage());
50
51      Glide.with(context1) RequestManager
52          .asBitmap() RequestBuilder<Bitmap>
53          .load(URI + imagesURI2.get(position)+ ".png") RequestBuilder<Bitmap>
54          .into(holder.getCircleImage2());
```

Código .java del RVAdapter2:

```
55 Glide.with(context1) RequestManager
56     .asBitmap() RequestBuilder<Bitmap>
57     .load(URI + imagesURI3.get(position)+ ".png") RequestBuilder<Bitmap>
58     .into(holder.getCircleImage3());
59 holder.getTvImage().setText(imageNames1.get(position));
60 holder.getCircleImage().setOnClickListener(new View.OnClickListener()
61 {
62     @Override
63     public void onClick(View v)
64     {
65         Toast.makeText(context1, text: "Play Music " , Toast.LENGTH_SHORT).show();
66         //holder.getCircleImage().setVisibility(View.INVISIBLE);
67         //holder.getCircleImage2().setVisibility(View.VISIBLE);
68     }
69 });
70 holder.getCircleImage2().setOnClickListener(new View.OnClickListener()
71 {
72     @Override
73     public void onClick(View v)
74     {
75         Toast.makeText(context1, text: "Pause Music " , Toast.LENGTH_SHORT).show();
```

Código .java del RVAdapter2:

```
75         Toast.makeText(context1, text: "Pause Music " , Toast.LENGTH_SHORT).show();|
76         //holder.getCircleImage().setVisibility(View.VISIBLE);
77         //holder.getCircleImage2().setVisibility(View.INVISIBLE);
78     }
79 });
80 holder.getCircleImage3().setOnClickListener(new View.OnClickListener()
81 {
82     @Override
83     public void onClick(View v) {
84         Toast.makeText(context1, text: "Stop Music " , Toast.LENGTH_SHORT).show();
85     }
86 });
87 }
88
89 @Override
90 public int getItemCount()
91 {
92     return imageNames1.size();
93 }
```



Hasta este punto se tiene el diseño de la segunda vista de la app, el usuario presiona botones de play, pause, stop y se muestra un mensaje:



Pregunta 3:

- Unificar las opciones PLAY y PAUSE en un solo.
- Es decir cuando se hace PLAY debe de cambiar el imageView automaticamente a PAUSE.
- Ocorre lo mismo cuando se presiona PAUSE este cambia automaticamente a PLAY.

- Se logró modificar el código desarrollado para combinar el play/pause.
- A continuación en las siguientes diapositivas se mostrará las modificaciones realizadas en código.



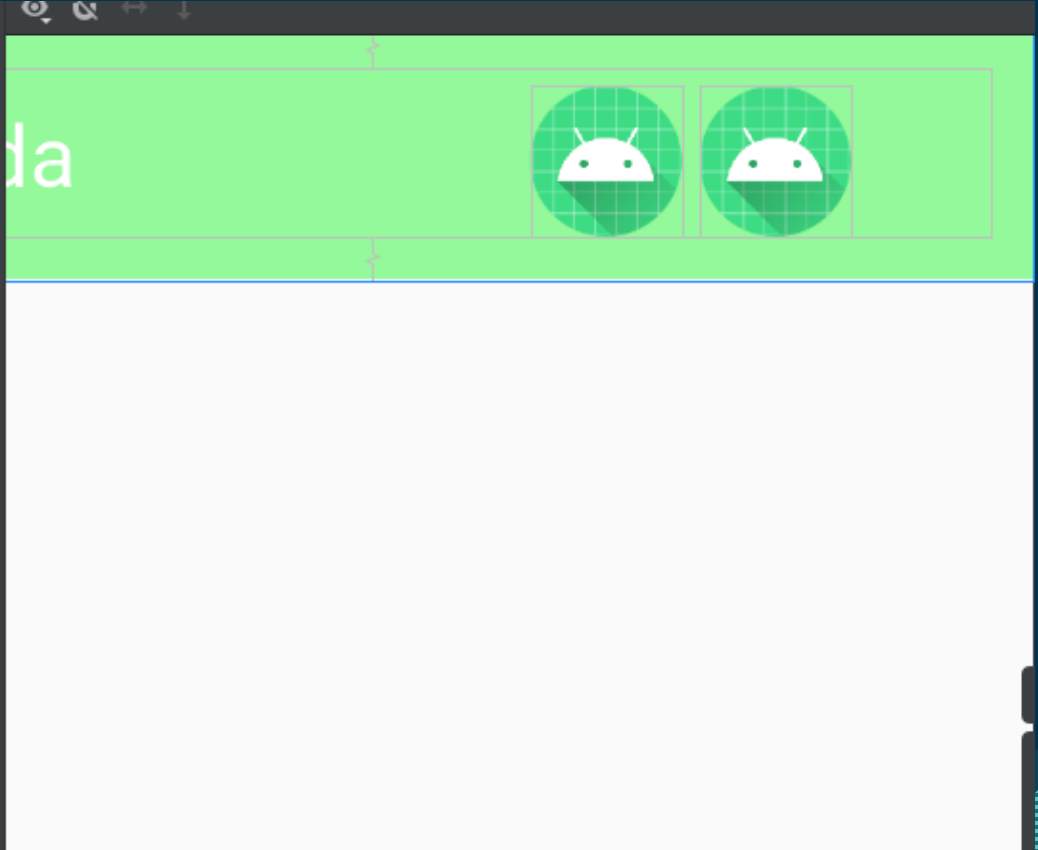
Código .java modificado de RVAdapter2:

```
60 holder.getCircleImage().setOnClickListener(new View.OnClickListener()
61 {
62     @Override
63     public void onClick(View v)
64     {
65         Toast.makeText(context1, text: "Play Music " , Toast.LENGTH_SHORT).show();
66         holder.getCircleImage().setVisibility(View.INVISIBLE);
67         holder.getCircleImage2().setVisibility(View.VISIBLE);
68     }
69 });
70 holder.getCircleImage2().setOnClickListener(new View.OnClickListener()
71 {
72     @Override
73     public void onClick(View v)
74     {
75         Toast.makeText(context1, text: "Pause Music " , Toast.LENGTH_SHORT).show();
76         holder.getCircleImage().setVisibility(View.VISIBLE);
77         holder.getCircleImage2().setVisibility(View.INVISIBLE);
78     }
79 });
```


Código .xml modificado de rl_list_item2:

Modificamos los marginLeft de play y pause, poniendo uno sobre otro y también en pause ponemos visibility="invisible"

```
18 android:clickable="true"></TextView>
19 <de.hdodenhof.circleimageview.CircleImageView
20     android:id="@+id/civItem1"
21     android:layout_width="45dp"
22     android:layout_height="45dp"
23     android:src="@mipmap/ic_launcher_round"
24     android:layout_marginLeft="250dp"
25     android:layout_marginTop="5dp"/>
26 <de.hdodenhof.circleimageview.CircleImageView
27     android:id="@+id/civItem2"
28     android:layout_width="45dp"
29     android:layout_height="45dp"
30     android:src="@mipmap/ic_launcher_round"
31     android:layout_marginLeft="250dp"
32     android:layout_marginTop="5dp"
33     android:visibility="invisible"/>
```





Hasta este punto se
obtiene como
resultado la
modificación de los
botones (imagenes)
play/pause:





A continuación se muestra un video de la funcionalidad de la aplicación, con sus respectivos requerimientos:

