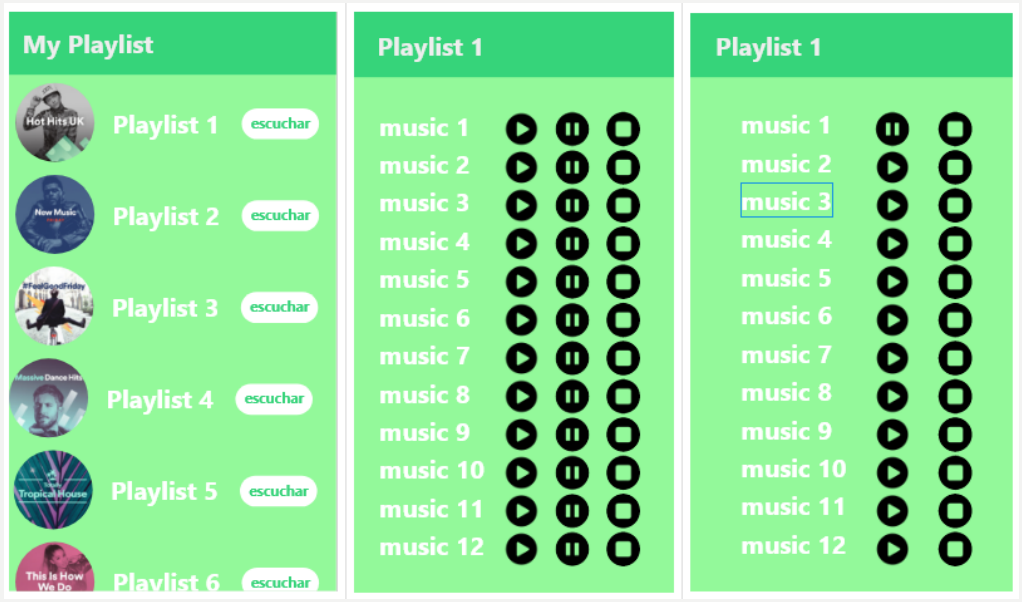
RESOLUCIÓN DEL EXAMEN HITO 4

NOMBRE: GUIDO PASTOR ESCOBAR IBAÑEZ

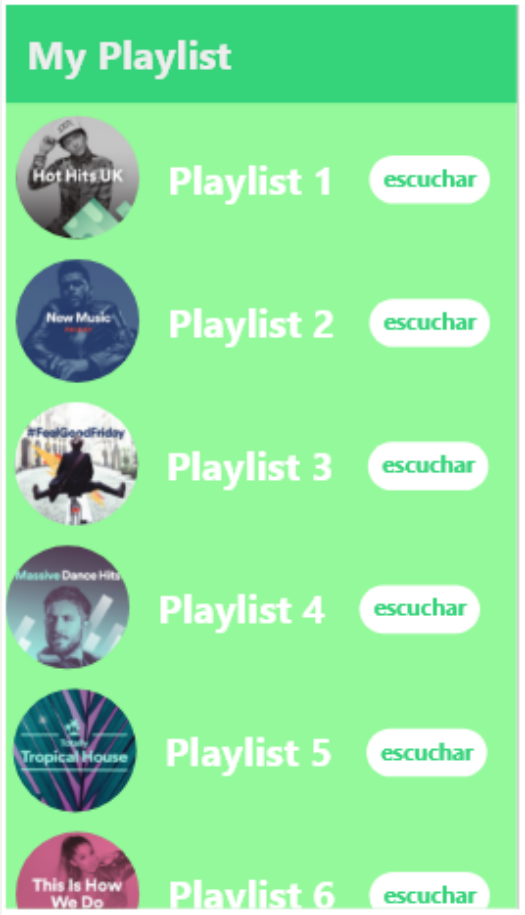
DEFENSA HITO 4

- El objetivo es crear un PLAYLIST en donde se listara una lista de albumes, en donde cada ALBUM contiene una lista de canciones. Para poder generar este listado se debera de utilizar un RecyclerView.
- Cada ALBUM contiene cierta cantidad de canciones, cada cancion tiene disponibles 3 OPCIONES. PLAY - PAUSE - STOP. De manera similar para generar este listado se debe utlizar el concepto de RecyclerView.
- Para el desarrollo de esta APLICACION se tiene los siguientes SCREENSHOTS.



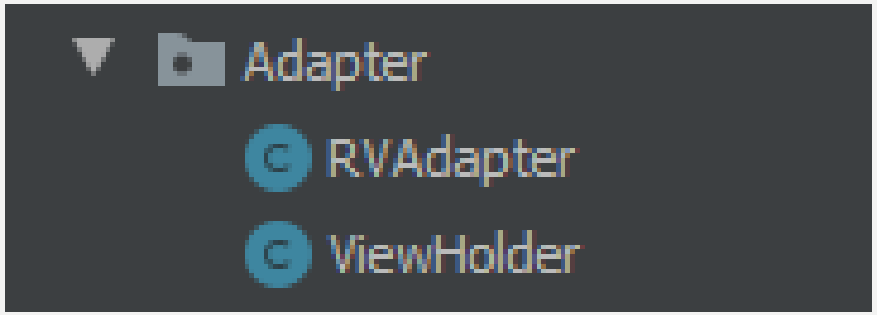
PREGUNTA 1

SCREEN 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.



RESOLUCIÓN DEL PROBLEMA PLANTEADO

- Bueno para hacer la resolución del primer ejerció tenemos que crear la primera vista que seria un reciclerView para eso tenemos que implementar las clases necesarias para adaptarlo al activi main y los layaots necesario para mostrar la vista necesaria. Tambien necesitamos un adapter y un viewHolder para adaptar la primera vista que es el ejercicio 1
- Implementamos las clases necesarias, el adapter es para la vista que se mostrara primero en vez del activitymain.
- El viewHolder es para declarar las variables necesarias para el layaut.



- Primero vamos a crear un archivo xml en la carpeta layaout este contendrá los elementos que se mostraran en en un circle view este contendrá las imágenes que se mostraran en un array
- Despues un textview y un button, también a este xml le vamos a dar un id para después llamarlo por código en el main activity

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.andro
    android:id="@+id/rlListItem"
    android:background="@color/verde2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="12dp">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/civItem"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:src="@mipmap/ic_launcher"/>
```

```
<TextView
    android:id="@+id/tvImage"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:text="canada"
    android:layout_marginStart="5dp"
    android:layout_toEndOf="@+id/civItem"
    android:layout_centerVertical="true"
    android:padding="15dp"
    android:textSize="20sp"
    android:textColor="@color/blanco"/>

<Button
    android:id="@+id/btplay"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:text="escuchar"
    android:layout_marginStart="10dp"
    android:layout_toEndOf="@id/tvImage"
    android:layout_centerVertical="true"
    android:background="@drawable/stilo"
```

Vista final

- Ahora una vez hecho la vista en el xml vamos a llamar al circle view, text view y al button en nuestra clase viewHolder este contendrá a nuestros id del circle view, text view y el button para después hacer sus propios geters
- Le damos a la clase un extends para que cree los métodos necesarios.

```
public class ViewHolder extends RecyclerView.ViewHolder
```

- Ahora declaramos las variables necesarias

```
public class ViewHolder extends RecyclerView.ViewHolder
{//clase k accede a nuestro componente    a la imagen y al texto

    private Button escuchar;

    private RelativeLayout rlContainer;
    private CircleImageView circleImage;
    private TextView tvImage;
```

- ahora creamos una función y la llamamos dentro de esta función llamaremos a las variables declaradas y llamamos a sus id en el xml

```
public ViewHolder(@NonNull View itemView)
{
    super(itemView);
    initComponents(itemView);
}
```

```
private void initComponents(View itemView)
{
    rlContainer = itemView.findViewById(R.id.rlListItem);
    circleImage = itemView.findViewById(R.id.civItem);
    tvImage = itemView.findViewById(R.id.tvImage);

    escuchar=itemView.findViewById(R.id.btplay);
}
```

- Ahora de las variables declaradas creamos sus geters

```
public Button getEscuchar() {
    return escuchar;
}
public CircleImageView getCircleImage() {
    return circleImage;
}

public TextView getTvImage() {
    return tvImage;
}
public RelativeLayout getRlContainer() {
    return rlContainer;
}
```

- Ahora en el RVAdapter llamamos al viewHolder para poder agarrar sus geters del viewholder para poder implemntarlos en los arrays
- Ahora a la clase RVAdapter utilizamos el extends para que nos cree 3 nuevos métodos necesarios para poder listarlos en la primera vista con sus geters.

```
extends RecyclerView.Adapter<ViewHolder>
```

- Ahora vamos a declarar las variables necesarias

```
private static final String TAG = "RVAdapter";
private Context context;
private  ArrayList<String> imageNames = new ArrayList<>();
private  ArrayList<String> imagesURI = new ArrayList<>();
```

- Ahora aplicamos algo de código para que sea funcional

```
public RVAdapter(Context context, ArrayList<String> imageNames, ArrayList<String> imagesURI)
{
    this.context = context;
    this.imageNames = imageNames;
    this.imagesURI = imagesURI;

}
```

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType)
{
    View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.rl_list_item, viewGroup, attachToRoot: false);
    ViewHolder vHolder = new ViewHolder(view);
    return vHolder;
}
```

```java
@Override
public void onBindViewHolder(@NonNull ViewHolder holder, final int position)
{
    String URI = "https://i.imgur.com/";
    Glide.with(context)
            .asBitmap()
            .load(URI + imagesURI.get(position) + ".png")
            .into(holder.getCircleImage());

    holder.getTvImage().setText(imageNames.get(position));
    holder.getEscuchar().setOnClickListener((v) -> {
            Intent intent = new Intent(context,playMusic.class   );
            context.startActivity(intent);
    });
    holder.getTvImage().setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(context,imageNames.get(position),Toast.LENGTH_SHORT).show();
    });
}
```

```java
@Override
public int getItemCount() {
    return imageNames.size();
}
```

- Ahora vamos al mainactivity.xml para declarar el recicler view donde se listaran todos los objetos que necesitamos
-

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.c
    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">



    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvContainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>

</RelativeLayout>
```

- Ahora en el activity main .java le damos la funcionalidad para que se listen los obetos que están en los arrays sus imágenes nombres y el button
- Ahora declaramos las variables necesarias para que se puedan listar el circleView el Textview y el button

```
private RecyclerView recyclerView;
private RVAdapter rvAdapter;
private ArrayList<String> imagesName = new ArrayList<>();
private ArrayList<String> imagesURL = new ArrayList<>();
private final String imgURL = "sypYnSP,9jMbaX2,rYndmdq,yB3d2qH,XzffKgy,XzffKgy,HBeK1ot,YCqbt8r,e
private final String imgNam = "Playist 1,Playist 2,Playist 3,Playist 4,Playist 5,Playist 6,Playi
```

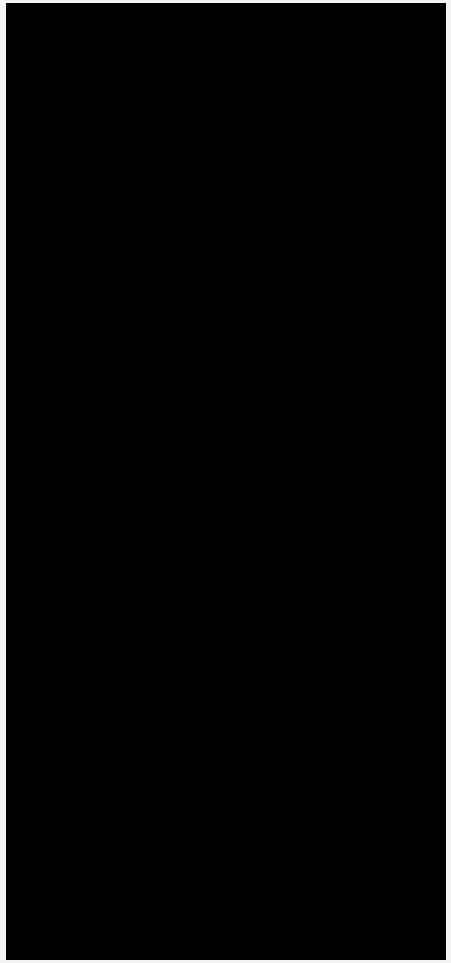
- Ahora declaramos algunas funciones necesarias para mostrar los contenidos mencionados

```
public void initImageBitMaps()
{
    imagesURL.addAll(Arrays.asList(imgURL.split( regex: ",")));
    imagesName.addAll(Arrays.asList(imgNam.split( regex: ",")));

}
```

```
public void initRecyclerView()
{
    recyclerView = findViewById(R.id.rvContainer);
    rvAdapter = new RVAdapter( context: this, imagesName, imagesURL);
    recyclerView.setAdapter(rvAdapter);
    recyclerView.setLayoutManager(new LinearLayoutManager( context: this));

}
```

AHORA VEAMOS EL FUNCIONAMIENTO DE LA APLICACIÓN

Reproducir video



PREGUNTA 2

SCREEN 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.

- Ahora para la pregunta dos necesitamos implementar lo mismo tenemos que crear un archivo xml en los layaouts para que tenga el diseño de como se va a visualizar
- Tambien tenemos que crear un nevos adpters y view holder y también un nuevo emply activity
- Todo esto es necesario para la vista dos que seria el nievo recicler view
- Ahora aplicaremos al rl list ítem 2. la vista necesaria donde se mostrara lo que necesitamos que se vea según la imagen anterior
- Ahora a este archivo xml le damos un id y el textView y tres circleView

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/ap
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@color/verde2"
android:padding="12dp"
    android:id="@+id/rlListItem2" >
```

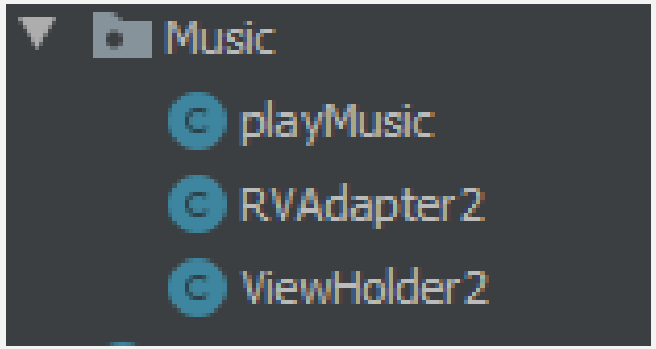
```
<TextView
    android:id="@+id/tv1"
    android:layout_width="240dp"
    android:layout_height="wrap_content"
    android:text="canada"
    android:layout_marginStart="5dp"
    android:layout_centerVertical="true"
    android:padding="15dp"
    android:textColor="@color/blanco"
    android:textSize="20sp"/>

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/cvR"
    android:layout_marginLeft="110dp"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:src="@mipmap/ic_launcher"/>
```

```
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/cvP"
    android:layout_marginLeft="210dp"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:src="@mipmap/ic_launcher"
    />

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/cvA"
    android:layout_marginLeft="290dp"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:src="@mipmap/ic_launcher"
    />
```

- Ahora colocamos el reciclre view en el activity play music

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/ap
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=".Music.playMusic">

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvContainerR"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</androidx.recyclerview.widget.RecyclerView>




</RelativeLayout>
```

- Ahora debemos crear una nueva carpeta que contendrá nuestro emply activity nuestro adpter y nuestro viewHolder.



- Ahora vamos a view holder2 para declarar los id y variables necesarias del rl list ítem 2.xml

```
public class ViewHolder2 extends RecyclerView.ViewHolder{

    private RelativeLayout rlcontainer;
    private CircleImageView cv1;
    private CircleImageView cv2;
    private CircleImageView cv3;
    private TextView tvnombre;
```

```
private void initComponents(View itemView)
{
    rlcontainer = itemView.findViewById(R.id.rlListItem2);
    cv1 = itemView.findViewById(R.id.cvR);
    cv2 = itemView.findViewById(R.id.cvP);
    cv3 = itemView.findViewById(R.id.cvA);
    tvnombre= itemView.findViewById(R.id.tv1);
}

public ViewHolder2(@NonNull View itemView) {
    super(itemView);
    initComponents(itemView);
}
public RelativeLayout getRlcontainer() {
    return rlcontainer;
}
```

```
public CircleImageView getCv1() {
    return cv1;
}

public CircleImageView getCv2() {
    return cv2;
}

public CircleImageView getCv3() {
    return cv3;
}

public TextView getTvnombre() {
    return tvnombre;
}
```

- Ahora implementamos los métodos necesarios para el rvadpter2

```
public class RVAdapter2 extends RecyclerView.Adapter<ViewHolder2> {
    private static final String TAG = "RVAdapter";
    private Context context;
    private ArrayList<String> imageNames = new ArrayList<>();
    private  ArrayList<String> im_rep = new ArrayList<>();
    private  ArrayList<String> im_pausa = new ArrayList<>();
    private  ArrayList<String> im_stopp = new ArrayList<>();
```

```
public RVAdapter2(Context context, ArrayList<String> imageNames, ArrayList<String> im_rep,
                  ArrayList<String> im_pausa,ArrayList<String> im_stop)
{
    this.context = context;
    this.imageNames = imageNames;
    this.im_rep = im_rep;
    this.im_pausa = im_pausa;
    this.im_stopp = im_stop;////////correccion
}
```

```java
@Override
public ViewHolder2 onCreateViewHolder( ViewGroup  viewGroup, int viewType) {
    View view = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.rl_list_item_2,
            viewGroup, attachToRoot: false);
    ViewHolder2 vHolder = new ViewHolder2(view);
    return vHolder;
}
```

```java
@Override
public void onBindViewHolder(final ViewHolder2 holder, final int position) {
    String URI = "https://i.imgur.com/";
    Glide.with(context)
            .asBitmap()
            .load(URI + im_rep.get(position) + ".png")
            .into(holder.getCv1());
    holder.getTvnombre().setText(imageNames.get(position));

    String URI2 = "https://i.imgur.com/";
    Glide.with(context)
            .asBitmap()
            .load(URI2 + im_pausa.get(position) + ".png")
            .into(holder.getCv2());

    String URI3 = "https://i.imgur.com/";
    Glide.with(context)
            .asBitmap()
            .load(URI3 + im_stopp.get(position) + ".png")
            .into(holder.getCv3());
```

```
holder.getCv1().setOnClickListener((v) → {
        //holder.getCv1().setVisibility(View.INVISIBLE);
        //holder.getCv2().setVisibility(View.VISIBLE);
         Toast.makeText(context, text: "PLAY MUSIC"+imageNames.get(position),Toast.LENGTH_SHORT).show();
});
holder.getCv2().setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //holder.getCv2().setVisibility(View.INVISIBLE);
        //holder.getCv1().setVisibility(View.VISIBLE);
         Toast.makeText(context, text: "PAUSE MUSIC",Toast.LENGTH_SHORT).show();
    }
});
holder.getCv3().setOnClickListener((v) → {
        Toast.makeText(context, text: "STOP MUSIC",Toast.LENGTH_SHORT).show();
});
```

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

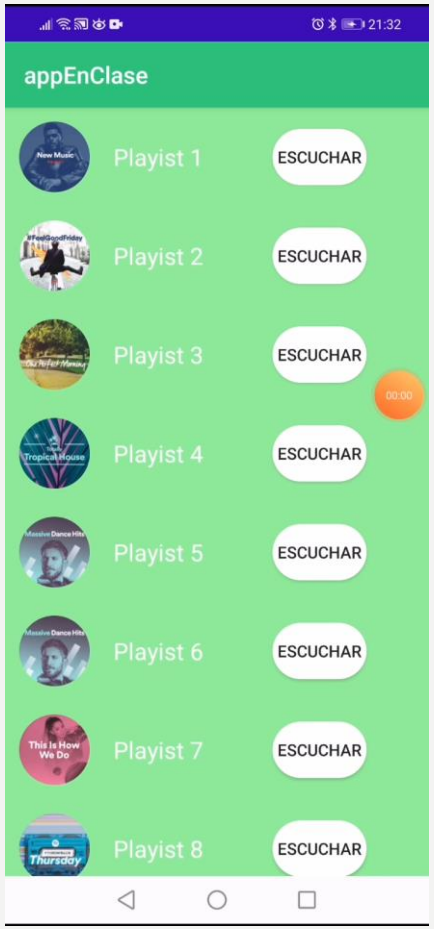
- Una vez implementado esto se vamos a implementar código al playMusic para que se pueda ver y listar la vista.

```
public class playMusic extends AppCompatActivity {



    private RecyclerView recyclerView1;
    private RVAdapter2 rvAdapter2;
    private ArrayList<String> imagesName = new ArrayList<>();
    private ArrayList<String> urlrep = new ArrayList<>();
    private ArrayList<String> urlpausa = new ArrayList<>();
    private ArrayList<String> urlstop = new ArrayList<>();

     private final String im_urlrep = "59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoT
    private final String im_urlpausa ="2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0RQ,2Nat0
    private final String im_urlstop ="LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQe,LJjdlQ
    private final String imagesNameee = "Playist 1,Playist 2,Playist 3,Playist 4,Playist 5,Playist 6,Playist 7,Playist 8, Pl
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_play_music);
    initImageBitMaps();
    initRecyclerView();
}

private void initRecyclerView() {
    recyclerView1 = findViewById(R.id.rvContainerR);
    rvAdapter2 = new RVAdapter2( context: this, imagesName, urlrep,urlpausa,urlstop);
    recyclerView1.setAdapter(rvAdapter2);
    recyclerView1.setLayoutManager(new LinearLayoutManager( context: this));
}

private void initImageBitMaps() {

    urlrep.addAll(Arrays.asList(im_urlrep.split( regex: ",")));
    urlpausa.addAll(Arrays.asList(im_urlpausa.split( regex: ",")));
    urlstop.addAll(Arrays.asList(im_urlstop.split( regex: ",")));
    imagesName.addAll(Arrays.asList(imagesNameee.split( regex: ",")));
}
```

AHORA VEAMOS EL FUNCIONAMIENTO

Reproducir video

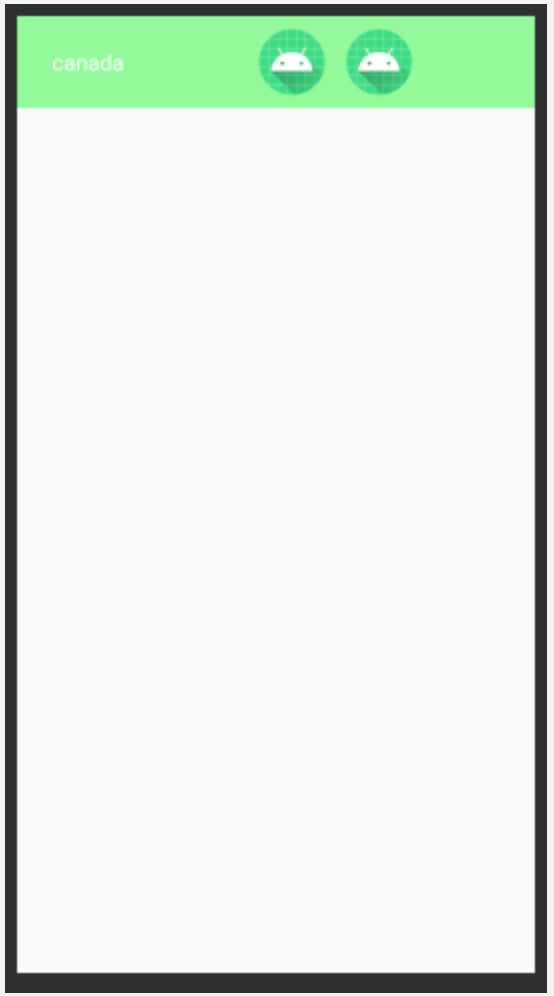


PREGUNTA 3

SCREEN 3	• Unificar las opciones PLAY y PAUSE en un solo. • Es decir cuando se hace PLAY debe de cambiar el imageView automaticamente a PAUSE. • Ocurre lo mismo cuando se presiona PAUSE este cambia automaticamente a PLAY.

- Para realizar la pregunta 3 solo se tiene que sobre poner el circle view de pause y play en el rl list ítem2.xml

```
<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/cvR"
    android:layout_marginLeft="210dp"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:src="@mipmap/ic_launcher"/>

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/cvP"
    android:layout_marginLeft="210dp"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:src="@mipmap/ic_launcher"
    />
```



- Ahora para que el circleview se oculte y se muestre solo es necesario aplicar un pequeño código esto se coloca en los get del cirlce view

```
holder.getCv1().setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        holder.getCv1().setVisibility(View.INVISIBLE);
        holder.getCv2().setVisibility(View.VISIBLE);
        Toast.makeText(context, text: "PLAY MUSIC"+" "+imageNames.get(position),Toast.LENGTH_SHORT).show();
    }
});
holder.getCv2().setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        holder.getCv2().setVisibility(View.INVISIBLE);
        holder.getCv1().setVisibility(View.VISIBLE);
        Toast.makeText(context, text: "PAUSE MUSIC"+" "+imageNames.get(position),Toast.LENGTH_SHORT).show();
    }
});
holder.getCv3().setOnClickListener((v) -> {
        Toast.makeText(context, text: "STOP MUSIC"+" "+imageNames.get(position),Toast.LENGTH_SHORT).show();
});
```

AHORA VEAMOS EL FUNCIONAMIENTO

Reproducir video