

Defensa Hito4 SOME

Adriana Contreras Arancibia

Esta es la resolución del
examen en android studio
de "Manejo de
RecyclerView "

Empezar



Índice

Sistemas móviles y embebidos

Concepto de Android Studio

Resolución defensa

1ra consigna: Diseño de la playlist con las imágenes correspondientes y las descripciones mediante mensajes toast.


2da consigna: Mostrar el listado de las canciones que forman parte de un ALBUM. Cada canción tiene 3 opciones (PLAY, PAUSE y STOP)

3ra consigna: Unificar las opciones PLAY y PAUSE en un solo y cuando se hace PLAY debe de cambiar el imageView automáticamente a PAUSE.



01

Sistemas moviles y embebidos



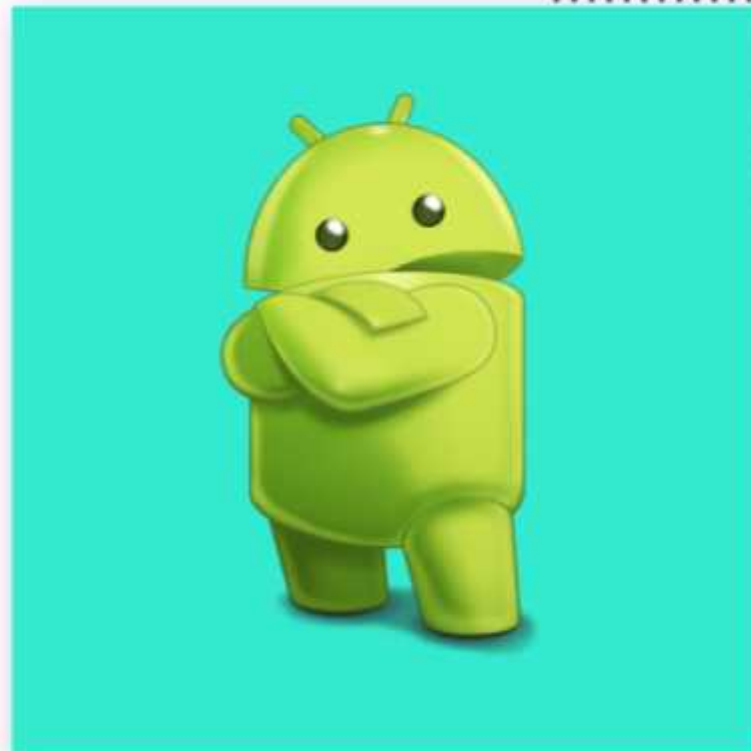
"Un sistema embebido o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real."

sistemas móviles y embebidos

Android Studio

Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android

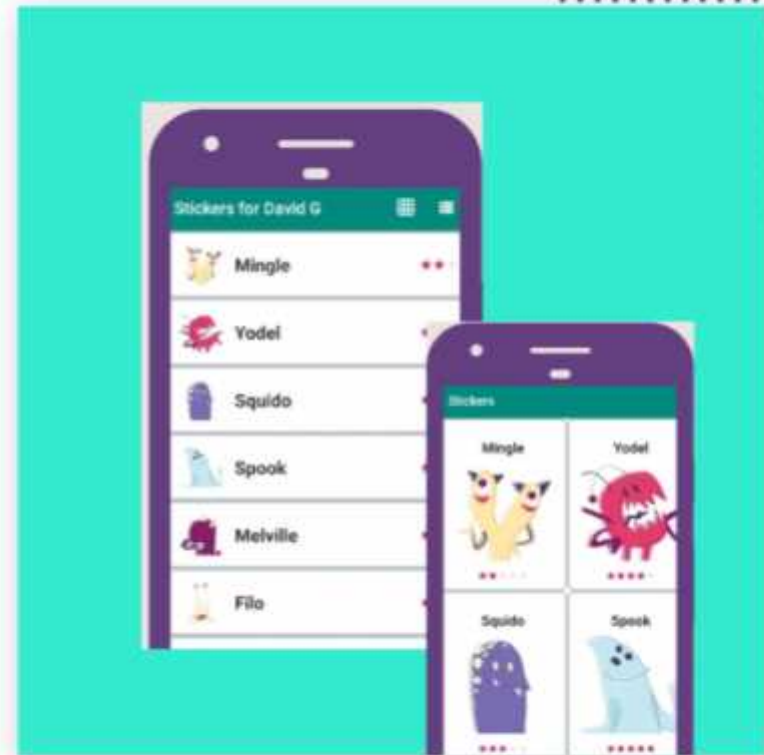
+info



sistemas móviles y embebidos

RecyclerView en Android

El RecyclerView es una versión más avanzada del tradicional ListView y lo que hace es mostrar datos cuyos elementos se van reciclando cuando ya no son visibles por el scroll de la lista, lo que mejora la performance en gran medida.



02

Defensa hito4 Consignas



1

Diseño de "My playlist"

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.

+info

2

Mostrar el listado de canciones y sus opciones PLAY, PAUSE, STOP

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 utilizar el concepto de RecyclerView.

+info

3

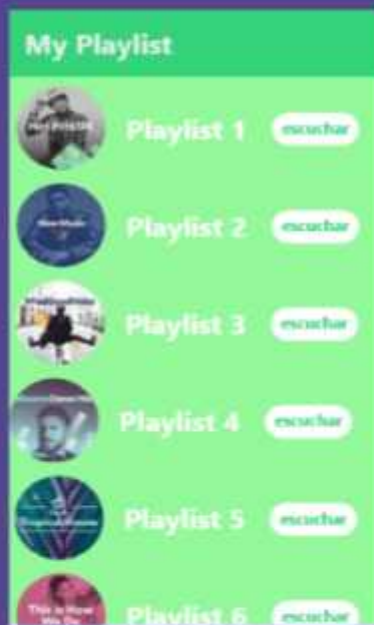
Unificar las opciones PLAY y PAUSE

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 a PLAY.

+info

1 Diseño de My Playlist

Se debe recrear el siguiente diseño



Realizar el diseño planteado respetando las imagenes debe de mostrar un mensaje(TOAST), una descripcion acerca de ese ALBUM(Ejem. "My Playlist 5") y al presionar en el BUTTON(escuchar) debe de redirigir a un nuevo activity,

1 Resolución



Para Generar el RecyclerView

Agregar la siguientes dependencias en el gradle.

Dependencia para el recyclerView

```
implementation 'androidx.recyclerview:recyclerview:1.1.0'
```

Dependencia para generar el circle imageView

```
implementation 'de.hdodenhof:circleimageview:3.1.0'
```

Dependencia para generar el glide for images

```
implementation 'com.github.bumptech.glide:glide:4.11.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'
```

Dentro del android Manifest agrgamos el permiso de internet

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

1 Resolucion



Codigo que resuleve el problema .xml

pieza de codigo que va dentro del Activity_Main.xml

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/rvContainer"  
    android:layout_width="odp"  
    android:layout_height="odp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
/>
```

pieza de codigo que va dentro del rl_list_item.xml para generar el texto de imageCircle

```
<TextView  
    android:id="@+id/tvImage"  
    android:layout_width="wrap_content"  
    android:layout_height="45dp"  
    android:layout_marginLeft="1dp"  
    android:layout_marginTop="15dp"  
    android:layout_toRightOf="@id/civItem"  
    android:fontFamily="sans-serif-medium"  
    android:padding="8dp"  
    android:text="@string/canada"  
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
    android:textColor="@color/white"  
    android:textSize="20sp">
```

pieza de codigo que va dentro del rl_list_item.xml para generar el imageCircle

```
<de.hdodenhof.circleimageview.CircleImageView  
    android:id="@+id/civItem"  
    android:layout_width="85dp"  
    android:layout_height="85dp"  
    android:src="@mipmap/ic_launcher_round"  
/>
```



1 Resolución



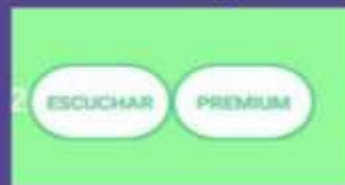
CODIGO QUE RESUELVE EL PROBLEMA .XML

pieza de código que va dentro del
rl_list_item.xml para generar los botones

```
<Button
    android:id="@+id/btnEscuchar1"
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textSize="11dp"
    android:layout_centerVertical="true"
    android:background="@drawable/circle"
    android:textColor="@color/verdeBar"
    android:text="@string/escuchar" />
<Button
    android:id="@+id/btnEscucharPremium2"
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textSize="11dp"
    android:layout_centerVertical="true"
    android:background="@drawable/circle"
    android:textColor="@color/verdeBar"
    android:text="@string/escucharPremium" />
```

Código que da estilos a los botones

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="40dp" />
    <stroke
        android:width="1.5dp"
        android:color="@color/verdeBar"
    />
    <solid android:color="@color/white"></solid>
</shape>
```



1 Resolución



CODIGO DEL MAIN ACTIVITY .JAVA

Declarar las variables para el recyclerView,
adapter, arrays y string

```
private RecyclerView recyclerView;  
private RVAdapter rvAdapter;  
private ArrayList<String> imagesName = new ArrayList<>();  
private ArrayList<String> imagesURL = new ArrayList<>();  
private final String imgURL = "V4OgA4OyB3dzqH.gjMbaX2.rYndmdq.sypYnSP.HBeK1ot.YCqbt8r.eLk31XX.4ZH7p7FO.XzffKgy.DJabk5C";  
private final String imgNam = "Playlist 1,Playlist 2 ,Playlist 3,Playlist 4,Playlist 5,Playlist 6, Playlist 7, Playlist 8";
```

Codigo de la funcion onCreate

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

Codigo de la separar el string mediante
"split"

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

Codigo para inicializar las variables
del recyclerView

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

1 Resolucion



Codigo del RVAdapter .Java

Decalarar las variables de los distintos arrays

```
private static final String TAG = "RVAdapter";  
private Context context;  
private ArrayList<String> imageNames= new ArrayList<>();  
private ArrayList<String> imageNames2= new ArrayList<>();  
private ArrayList<String> imagesURI= new ArrayList<>();  
private ArrayList<String> imagesURI2= new ArrayList<>();
```

Constructor del RVAdapter

```
public RVAdapter(Context context, ArrayList<String> imageNames,  
ArrayList<String> imagesURI) {  
    this.context=context;  
    this.imageNames=imageNames;  
    this.imagesURI=imagesURI;  
    this.imageNames2=imageNames2;  
    this.imagesURI2 = imagesURI2;  
}
```

Metodo del OnCreateViewHolder

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

funcion para retornar el conteo de las imagenes

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


1 Resolucion



Codigo del RVAdapter .Java

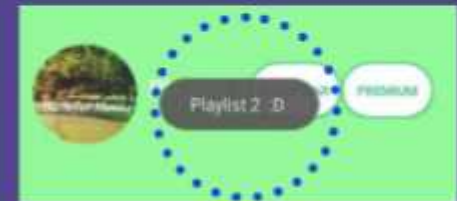
Funcion que nos permite obtener las imagenes de internet mediante su link y generar un mensaje cada vez que se presione sobre el texto "My playlist"

```
@Override
public void onBindViewHolder(ViewHolder holder, final int position) {
    holder.setOnClickListener();
    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.getRlContainer().setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Log.d(TAG, "Clicked on: " + imageNames.get(position));
            Toast.makeText(context, imageNames.get(position) + ":D", Toast.LENGTH_LONG).show();
        }
    });
}
```

Url inicial de cada imagen la cual se concatenara para obtener la imagen establecida

Establecer un evento click cada ves que se precione sobre la imagen y el textView

mostrar un mensaje(TOAST), una descripcion acerca de ese ALBUM(Ejem. "My Playlist 5")



1 Resolución



Codigo del ViewHolder .Java

Declarar las variables para el rlContainer, los botones las imagenes y los textView que se usaron en el diseño

Context context;
private RelativeLayout rlContainer;
private RelativeLayout rlContainer2;
private CircleImageView circleImage;
private TextView tvImage;
private ImageView imgPlay, imgPause, imgStop;
private Button btnescuchar, btnescucharPremium;

Inicializar los botones, las imagenes con sus respectivos id's

```
private void initComponents(View itemView) {  
    context = itemView.getContext();  
    rlContainer = itemView.findViewById(R.id.rListItem);  
    circleImage = itemView.findViewById(R.id.civItem);  
    tvImage = itemView.findViewById(R.id.tvImage);  
    btnescucharPremium = itemView.findViewById(R.id.btnEscucharPremium2);  
    btnescuchar = itemView.findViewById(R.id.btnEscuchar1);  
    rlContainer2 = itemView.findViewById(R.id.rListItem);  
    imgPlay = itemView.findViewById(R.id.imgPlay1);  
    imgPause = itemView.findViewById(R.id.imgPause1);  
    imgStop = itemView.findViewById(R.id.imgStop1);  
}
```

1 Resolución



Codigo del ViewHolder .Java

Mediante el Evento onClick y usando un switch
llamamos al siguiente activity llamado "escuchados"

```
void setOnClickListener()  
{  
    btnescuchar.setOnClickListener(this);  
    btnescucharPremium.setOnClickListener(this);  
}  
  
@Override  
public void onClick(View v) {  
    switch (v.getId())  
    {  
        case R.id.btnEscuchar:  
            Intent intent = new Intent(context, escuchardos.class);  
            context.startActivity(intent);  
            break;  
        case R.id.btnEscucharPremium:  
            Intent intent2 = new Intent(context, escuchardos.class);  
            context.startActivity(intent2);  
            break;  
    }  
}
```

Generamos los constructores getters

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


1 Resolución



RESULTADO FINAL DEL DISEÑO

Obtenemos las imagenes mediante el link de direccion todo esto para que la aplicación sea más liviana

Descripcion de la play list mediante un Toast.make que describe que numero de playlist correspondiente



aplicando estilos generamos estos tipos de Botones

TextView del nombre de la playlist

Diseño de "My playlist"

2 Mostrar el listado de canciones y sus opciones PLAY, PAUSE, STOP

SE DEBE RECREAR EL SIGUIENTE DISEÑO

Playlist 1			
music 1	▶	⏸	⏹
music 2	▶	⏸	⏹
music 3	▶	⏸	⏹
music 4	▶	⏸	⏹
music 5	▶	⏸	⏹
music 6	▶	⏸	⏹
music 7	▶	⏸	⏹
music 8	▶	⏸	⏹
music 9	▶	⏸	⏹
music 10	▶	⏸	⏹
music 11	▶	⏸	⏹
music 12	▶	⏸	⏹

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.



2 Resolución



CODIGO QUE RESUELVE EL PROBLEMA .XML

pieza de código que se agrega en el Activity_escuchados.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:background="@color/verdeBackground"
    tools:context="Adapterescuchados">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvContainerR"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>
</RelativeLayout>
```

Dentro del rl_listem_dos.xml se diseña y posiciona los imageView de PLAY, PAUSE, STOP

```
<ImageView
    android:id="@+id/imgStop1"
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textSize="11dp"
    android:layout_marginLeft="250dp"
    android:layout_centerVertical="true"
    android:background="@drawable/circle"
    android:textColor="@color/verdeBar" />
```

```
<ImageView
    android:id="@+id/imgPlay1"
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textSize="11dp"
    android:layout_centerVertical="true"
    android:layout_marginLeft="170dp"
    android:background="@drawable/circle"
    android:textColor="@color/verdeBar" />
```

```
<ImageView android:id="@+id/imgPause1"
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textSize="11dp"
    android:layout_marginLeft="170dp"
    android:layout_centerVertical="true"
    android:background="@drawable/circle"
    android:textColor="@color/verdeBar" />
```


2 Resolución



CODIGO QUE RESUELVE EL PROBLEMA .JAVA

Dentro de la clase "escuchados" se agrega el siguiente código para declarar los arrays para la url y nombre de cada imagen de play, pause, stop

```
private RecyclerView recyclerView;  
private RVAdapterdos rvAdapterdos;  
private ArrayList<String> imagesStopURL = new ArrayList<>();  
private ArrayList<String> imagesPauseURL = new ArrayList<>();  
private ArrayList<String> imagesPlayURL = new ArrayList<>();  
private ArrayList<String> imagesName2 = new ArrayList<>();
```

En una variable de tipo string declaramos los nombres, los url correspondientes a cada imagen

```
private final String imgplay = "59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq,59AoTWq";  
private final String imgpause = "2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO,2NatoRO";  
private final String imgstop = "LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe,LjdlQe";  
private final String imgNam2 = "music 1,music 2,music 3,music 4,music 5,music 6,music 7,music 8,music 9,music 10,music 11,music 12";
```

music 1



Imagen de referencia

2 Resolución



Metodo onCreate y initRecyclerView

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

public void initRecyclerView() {
    recyclerView = findViewById(R.id.rvContainerR);
    rvAdapterdos = new RVAdapterdos(this,
    imagesName2, imagesPlayURL, imagesPauseURL, imagesStopU
    RL);
    recyclerView.setAdapter(rvAdapterdos);
    recyclerView.setLayoutManager(new
    LinearLayoutManager(this));
}
```

Inicializacion de la variables para
posteriormente separar los caracteres
string mediante un "split"

```
public void initImageBitMaps() {
    imagesPlayURL.addAll(Arrays.asList(imgplay.split(",")));
    imagesPauseURL.addAll(Arrays.asList(imgpause.split(",")));
    imagesStopURL.addAll(Arrays.asList(imgstop.split(",")));
    imagesName2.addAll(Arrays.asList(imgNam2.split(",")));
}
```

2 Resolución



CODIGO QUE RESUELVE EL RVADAPTER_DOS .JAVA

Variables que se declaran

```
private static final String TAG = "RVAdapterdos";  
private Context context;  
private ArrayList<String> imagesStopURL = new ArrayList<>();  
private ArrayList<String> imagesPauseURL = new ArrayList<>();  
private ArrayList<String> imagesPlayURL = new ArrayList<>();  
private ArrayList<String> imagesName2 = new ArrayList<>();
```

Se genera el constructor

```
public RVAdapterdos(Context context, ArrayList<String>  
imageNames2, ArrayList<String>  
imagesPlayURL, ArrayList<String>  
imagesPauseURL, ArrayList<String> imagesStopURL ) {  
    this.context = context;  
    this.imagesName2 = imageNames2;  
    this.imagesPlayURL = imagesPlayURL;  
    this.imagesPauseURL = imagesPauseURL;  
    this.imagesStopURL = imagesStopURL;  
}
```

Se crea el metodo de onCreateViewHolder

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


2 Resolución



Mediante el metodo "onBindViewHolder" obtenemos las imagenes de play, pause y stop desde la url de internet

```
String URI = "https://i.imgur.com/";  
Glide.with(context)  
    .asBitmap()  
    .load(URI + imagesPlayURL.get(position) + ".png")  
    .into(holder.getImgPlay());  
holder.getTvImage1().setText(imagesName2.get(position));
```



En esta pieza de código obtiene la imagen de "stop" desde la url de internet

```
Glide.with(context)  
    .asBitmap()  
    .load(URI + imagesStopURL.get(position) + ".png")  
    .into(holder.getImgStop());
```



En esta pieza de código obtiene la imagen de "pause" desde la url de internet

```
Glide.with(context)  
    .asBitmap()  
    .load(URI + imagesPauseURL.get(position) + ".png")  
    .into(holder.getImgPause());
```



2 Resolución



pieza de código que cuando se presiona
PLAY mostrar el TOAST("Play music").

```
holder.getImgPlay().setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(context,"play  
music",Toast.LENGTH_SHORT).show();  
    }  
});
```

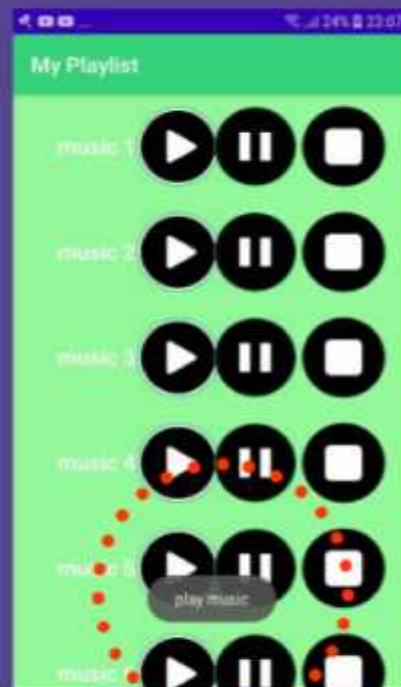


imagen de referencia

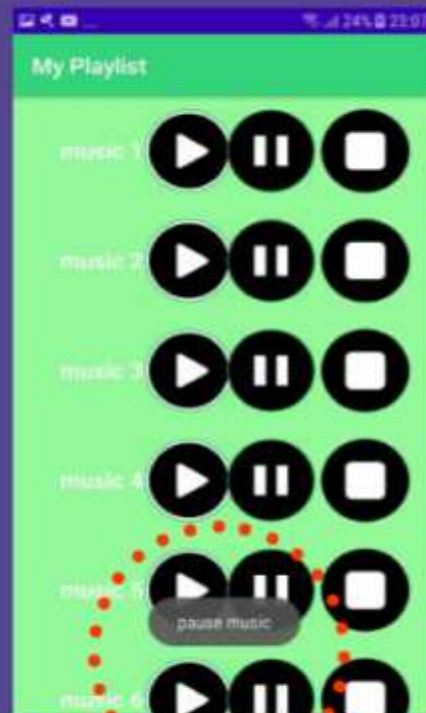
2 Resolución



pieza de código que cuando se presiona PAUSE mostrar el TOAST("Pause music").

```
holder.getImgPause().setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(context, "pause  
music", Toast.LENGTH_SHORT).show();  
    }  
});
```

imagen de referencia



2 Resolución



pieza de código que cuando se presiona STOP mostrar el TOAST("Stop music").

```
holder.getImgStop().setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(context, "stop  
music", Toast.LENGTH_SHORT).show();  
    }  
});
```

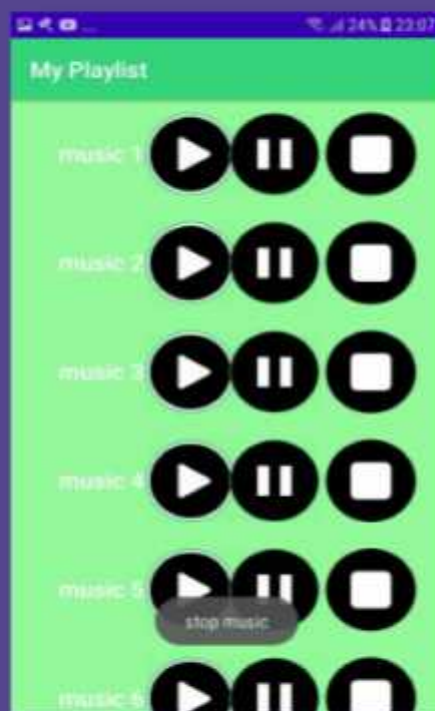


imagen de referencia



2 Resolucion



CODIGO QUE RESUELVE EL VIEWHOLDER_DOS .JAVA

declaracion de variables

```
Context context;  
private RelativeLayout rlContainer;  
private ImageView imgPlay, imgPause, imgStop;  
private TextView tvImage1;
```

metodo ViewHolderDos

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

metodo initComponents

```
private void initComponents(View  
itemView) {  
    context = itemView.getContext();  
    rlContainer =  
itemView.findViewById(R.id.rListItem1);  
    tvImage1 =  
itemView.findViewById(R.id.tvImage2);  
    imgPlay =  
itemView.findViewById(R.id.imgPlay1);  
    imgPause =  
itemView.findViewById(R.id.imgPause1);  
    imgStop =  
itemView.findViewById(R.id.imgStop1);  
}
```

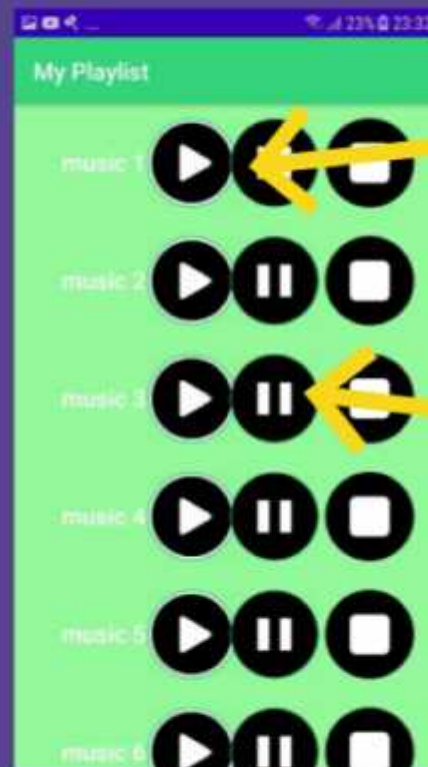
pieza de código que genera los getters de los imageView y el rlContainer

```
public Context getContext() {  
    return context;  
}  
public RelativeLayout getRlContainer() {  
    return rlContainer;  
}  
public ImageView getImgPlay() {  
    return imgPlay;  
}  
public ImageView getImgPause() {  
    return imgPause;  
}  
public ImageView getImgStop() {  
    return imgStop;  
}  
public TextView getTvImage1() {  
    return tvImage1;  
}
```

2 Resolución



RESULTADO FINAL DEL DISEÑO



Diseño del album de canciones

`Toast.makeText(context,"play music",Toast.LENGTH_SHORT).show();`

`Toast.makeText(context,"pause music",Toast.LENGTH_SHORT).show();`

`Toast.makeText(context,"stop music",Toast.LENGTH_SHORT).show();`

3 Diseño y funcionalidad de la calculadora custom

SE DEBE RECREAR EL SIGUIENTE DISEÑO

Playlist 1

music 1	⏸	⏮
music 2	▶	⏮
music 3	▶	⏮
music 4	▶	⏮
music 5	▶	⏮
music 6	▶	⏮
music 7	▶	⏮
music 8	▶	⏮
music 9	▶	⏮
music 10	▶	⏮
music 11	▶	⏮
music 12	▶	⏮

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 a PLAY.



3 Resolución



PARA REOLVER ESTA PREGUNTA VOLVEREMOS AL RL_LISTEN_DOS.XML Y MODIFICAREMOS UNA LINEA DE CODIGO QUE ESLA PROPIEDAD "ANDROID:LAYOUT_MARGINLEFT"

Modificamos el margin left de la primera imagen(play) y le cambiamos a "170dp"

```
<ImageView  
    android:id="@+id/imgPlay1"  
    android:layout_width="75dp"  
    android:layout_height="wrap_content"  
    android:textSize="11dp"  
    android:layout_centerVertical="true"  
    android:layout_marginLeft="170dp"  
    android:background="@drawable/circle"  
    android:textColor="@color/verdeBar" />
```

La idea es que quede sobrepuesto, otra opción tambien es usar las propiedades visible e invisible de android

Modificamos el margin left de la segunda imagen(pause) y le cambiamos a "170dp"

```
<ImageView  
    android:id="@+id/imgPause1"  
    android:layout_width="75dp"  
    android:layout_height="wrap_content"  
    android:textSize="11dp"  
    android:layout_marginLeft="170dp"  
    android:layout_centerVertical="true"  
    android:background="@drawable/circle"  
    android:textColor="@color/verdeBar" />
```

3 Resolución



A LA VEZ QUE MODIFICAMOS RL_LISTEN_DOS.XML LA PROPIEDAD "ANDROID:LAYOUT_MARGINLEFT",
VAMOS A MODIFICAR EN EL "RVADPATERDOS. JAVA" AGREGANDO DOS LINEAS DE CÓDIGO



Agregamos las siguientes dos líneas en el metodo onCreate del boton play

```
holder.getImgPlay().setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        holder.getImgPlay().setVisibility(View.INVISIBLE);
        holder.getImgPause().setVisibility(View.VISIBLE);
        Toast.makeText(context,"play
music",Toast.LENGTH_SHORT).show();
    }
});
```



Agregamos las siguientes dos líneas en el metodo onCreate del boton pause

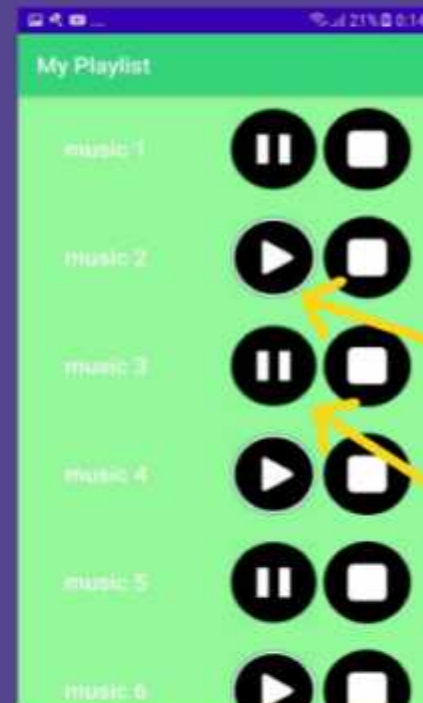
```
holder.getImgPause().setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        holder.getImgPause().setVisibility(View.INVISIBLE);
        holder.getImgPlay().setVisibility(View.VISIBLE);
        Toast.makeText(context,"pause
music",Toast.LENGTH_SHORT).show();
    }
});
```


3 Resolución



RESULTADO FINAL DEL DISEÑO

Botones
unificados
en uno



```
holder.getImgPlay().setVisibility(View.INVISIBLE);
```

```
holder.getImgPause().setVisibility(View.VISIBLE);
```

```
holder.getImgPause().setVisibility(View.INVISIBLE);
```

```
holder.getImgPlay().setVisibility(View.VISIBLE);
```

Diseño Unificando las opciones PLAY y PAUSE en un solo.

¡Gracias!



UNIFRANZ
Internacionalizate