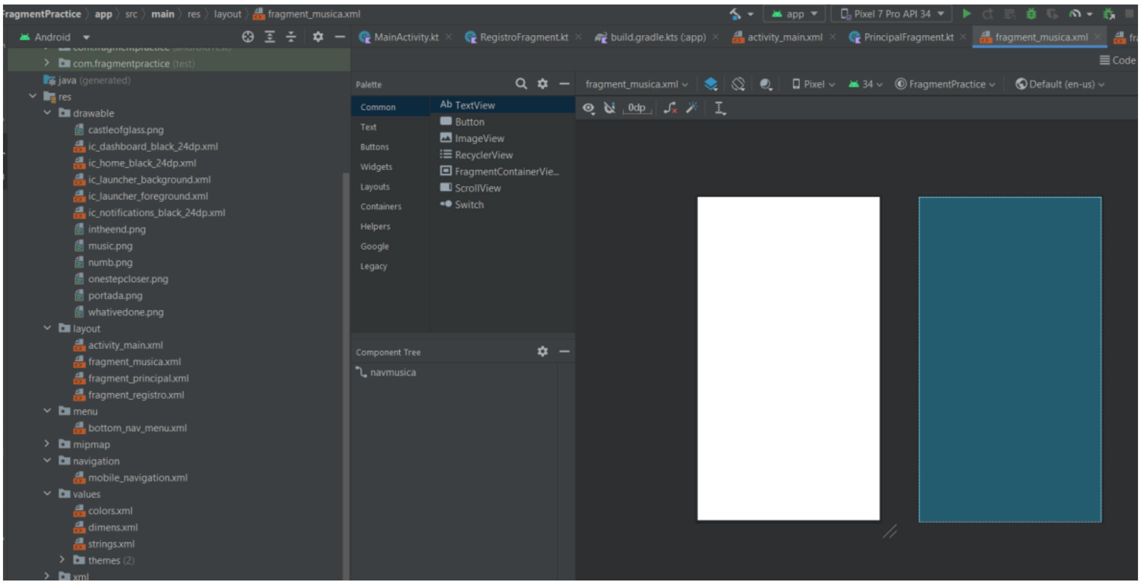
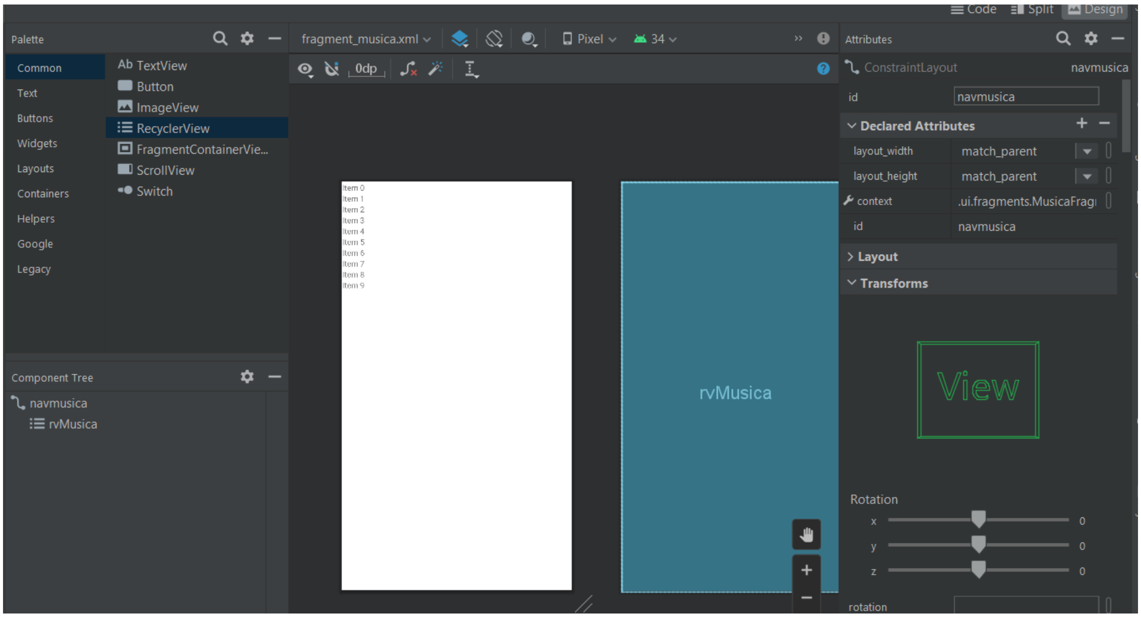
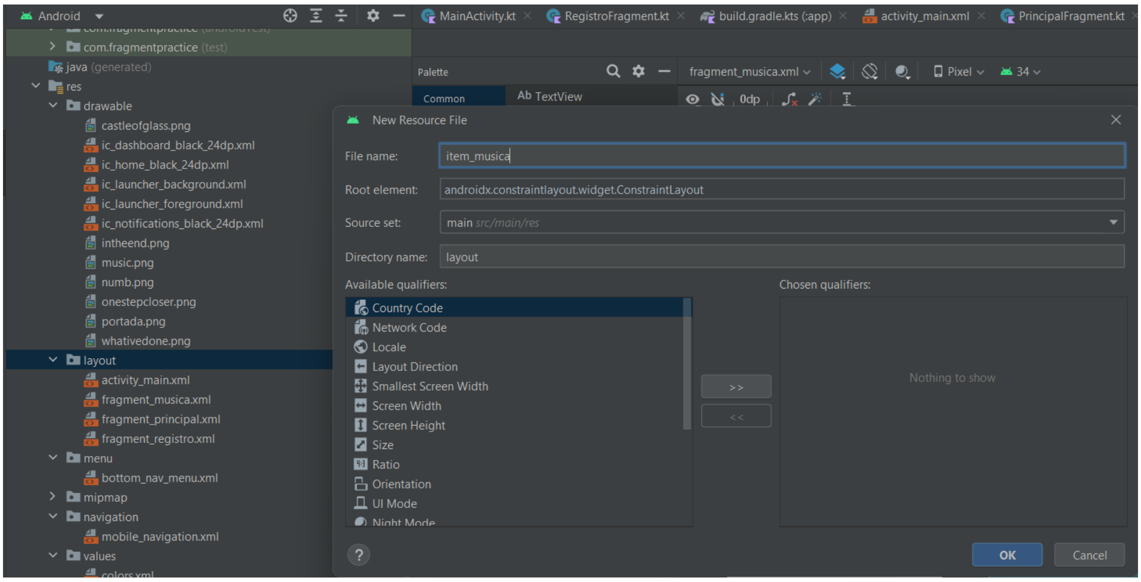
1



2. Agregar contenedor Recycleview:



3. Agregar en la carpeta Layouts : New layout resourcefile



New Resource File

File name: item_musica

Root element: androidx.constraintlayout.widget.ConstraintLayout

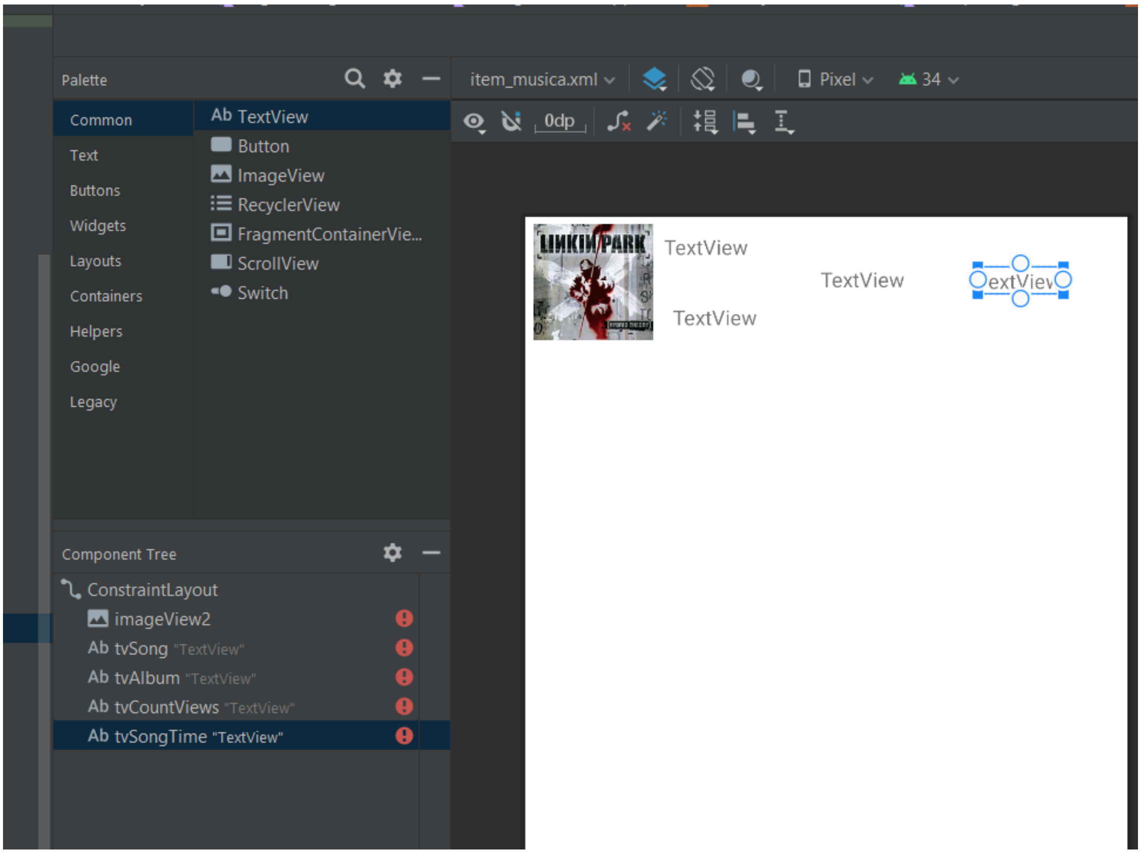
Source set: main src/main/res

Directory name: layout

Available qualifiers:

Chosen qualifiers: Nothing to show

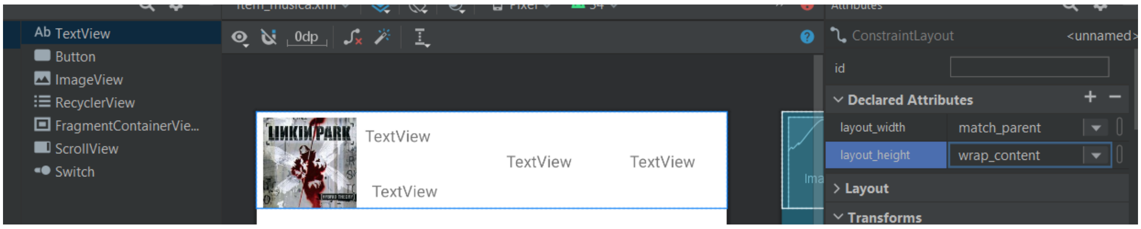
4. Agregar y cambiar id de los componentes:



Component Tree

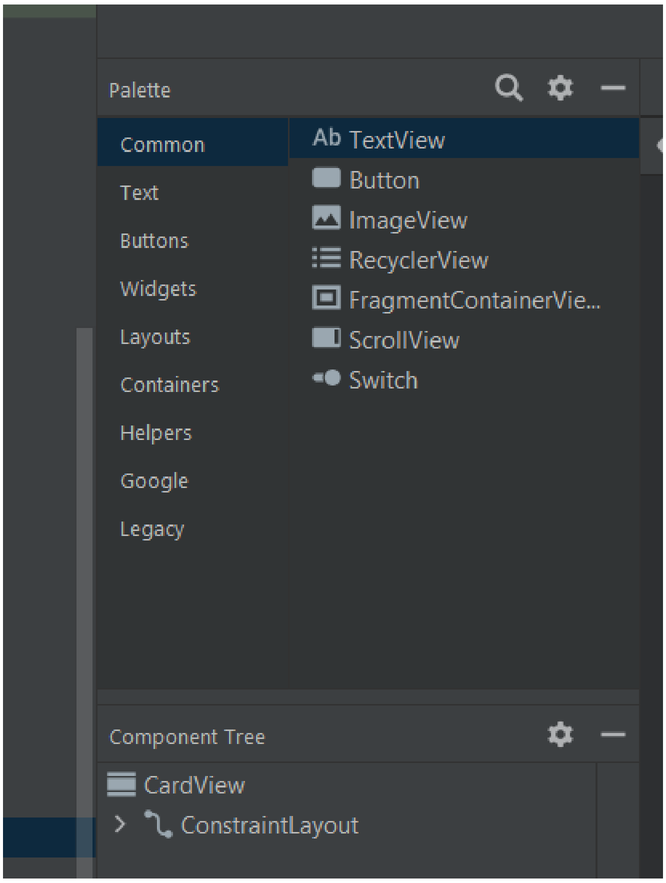
ConstraintLayout
imageView2
Ab tvSong "TextView"
Ab tvAlbum "TextView"
Ab tvCountViews "TextView"
Ab tvSongTime "TextView"

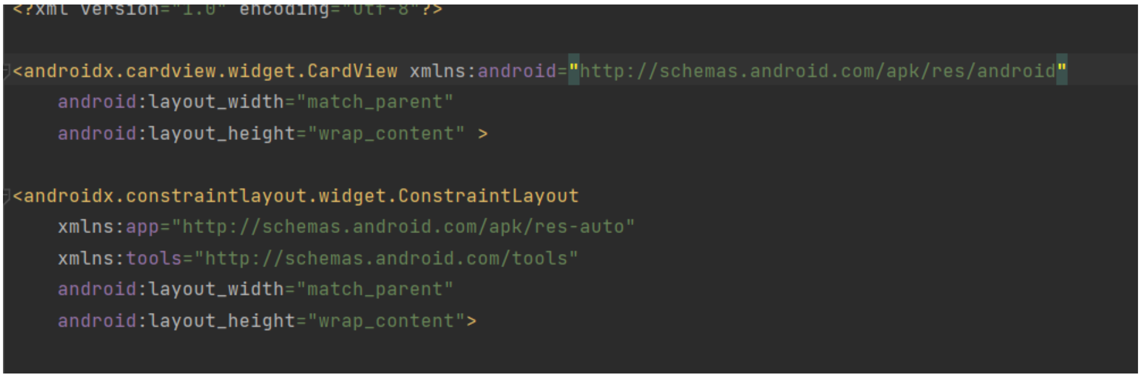
5. wrap_content en height



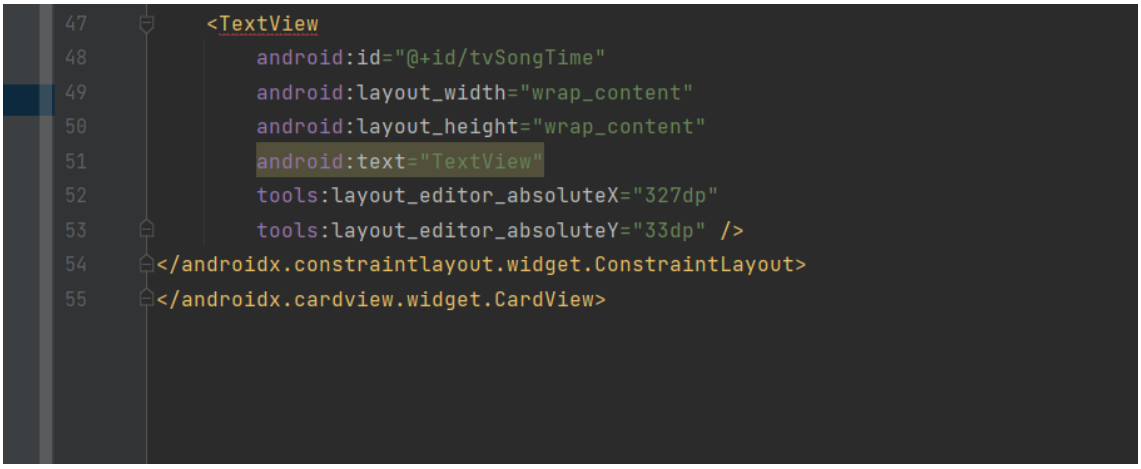
layout_width match_parent
layout_height wrap_content

6. Agregar un cardview:



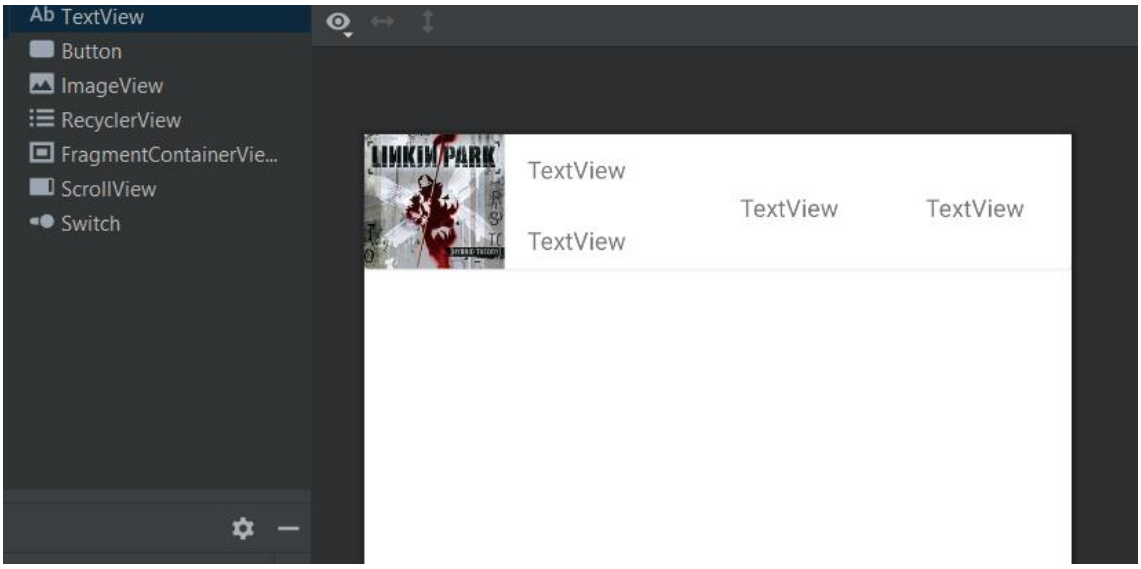


```
<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

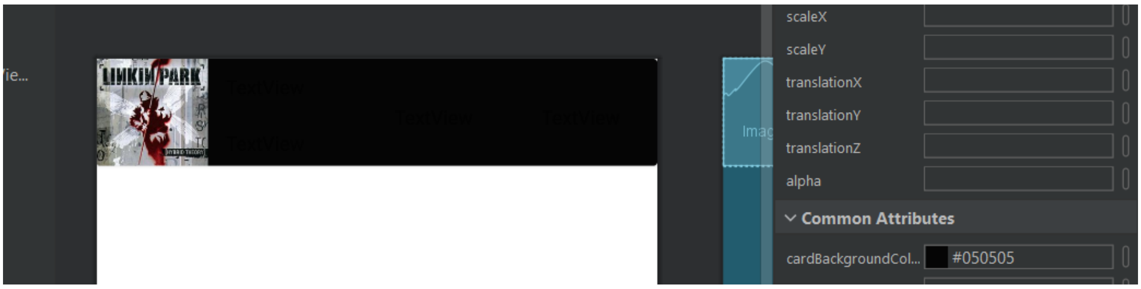


```
    <TextView
        android:id="@+id/tvSongTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        tools:layout_editor_absoluteX="327dp"
        tools:layout_editor_absoluteY="33dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>
```

7. Acomodar los layouts:

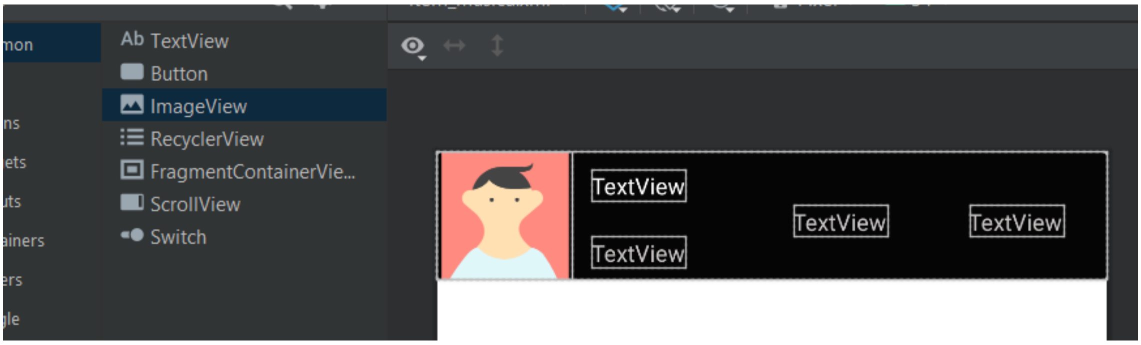


8. Cardviewbackgroundcolor

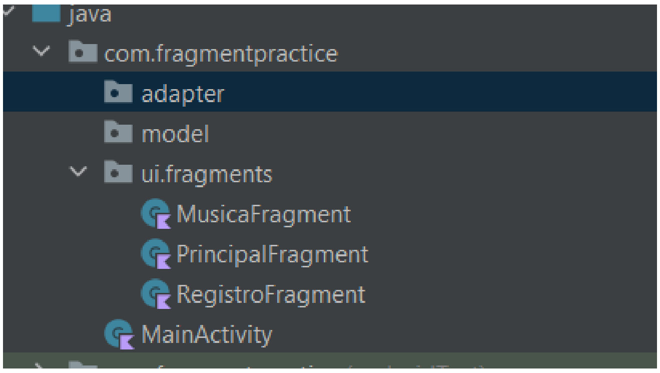


cardBackgroundCol... #050505

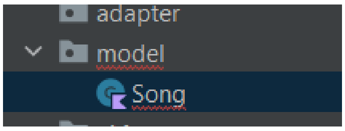
9. Usar un imageview en lugar de la imagen insertada anteriormente

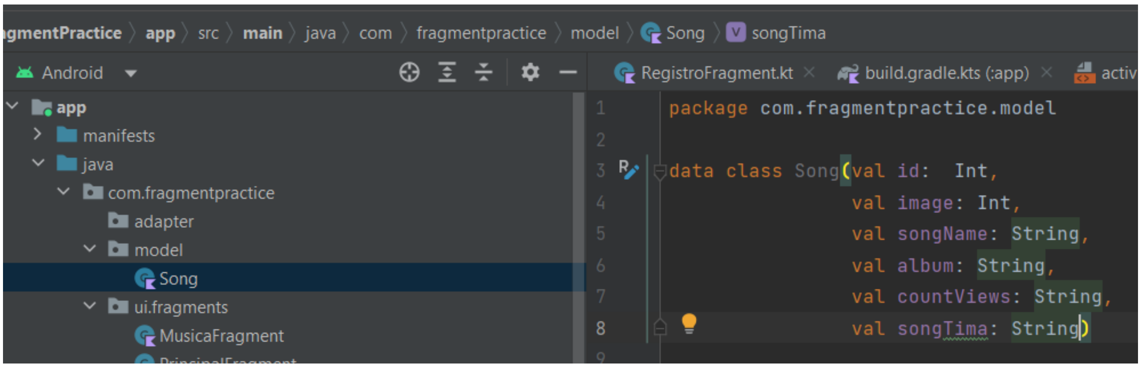


10. Agregar carpeta model y adapter



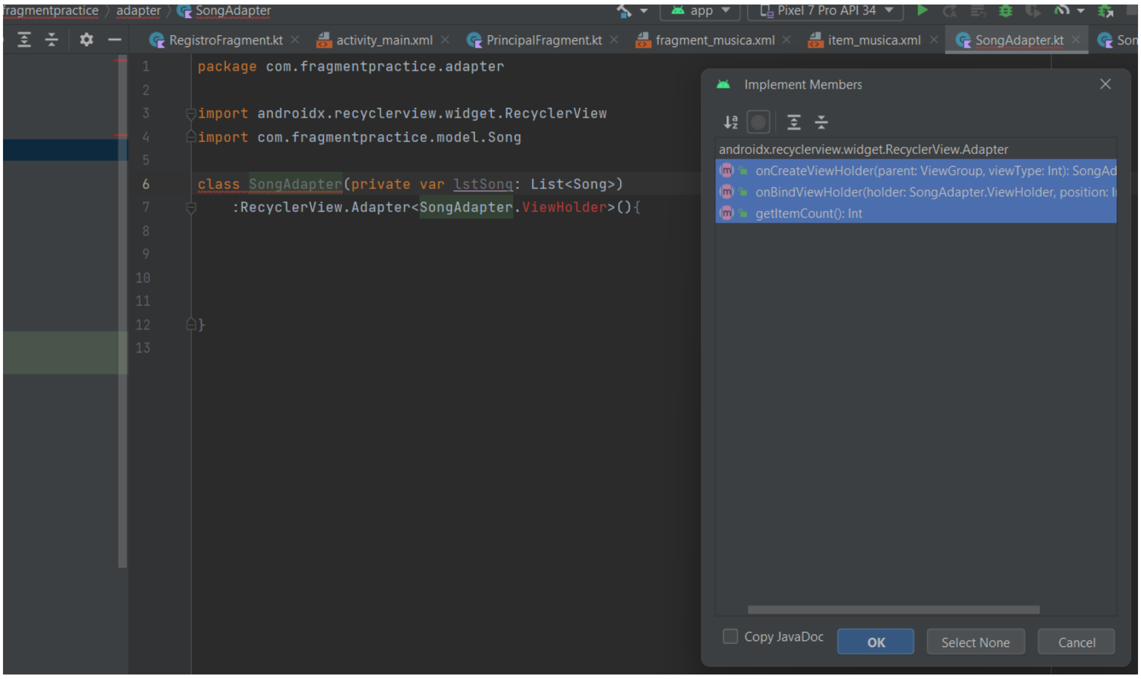
11. Agregar clase song en moderls (new – koltinclassfile - dataclass)





```
package com.fragmentpractice.model

data class Song(val id:  Int,
                val image: Int,
                val songName: String,
                val album: String,
                val countViews: String,
                val songTima: String)
```

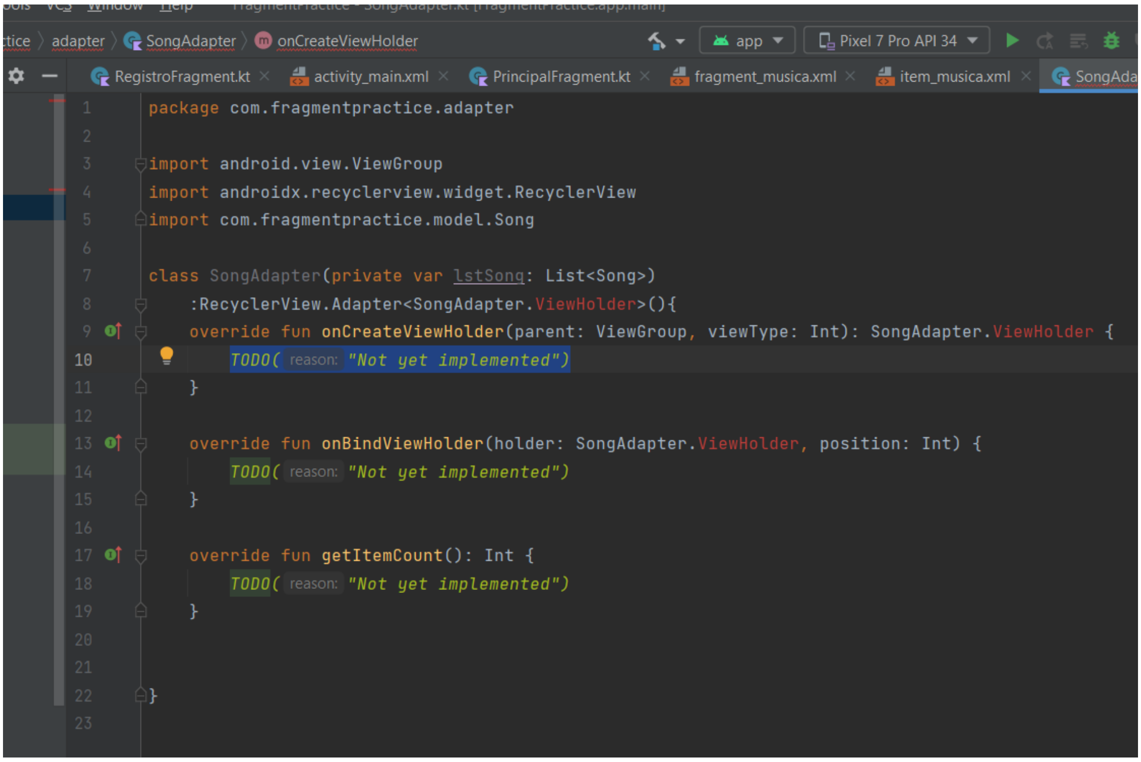
12. Agregar clase SongAdapter en adapter (new – koltinclassfile - dataclass)



```
package com.fragmentpractice.adapter

import androidx.recyclerview.widget.RecyclerView
import com.fragmentpractice.model.Song

class SongAdapter(private var lstSong: List<Song>)
    :RecyclerView.Adapter<SongAdapter.ViewHolder>(){



}
```

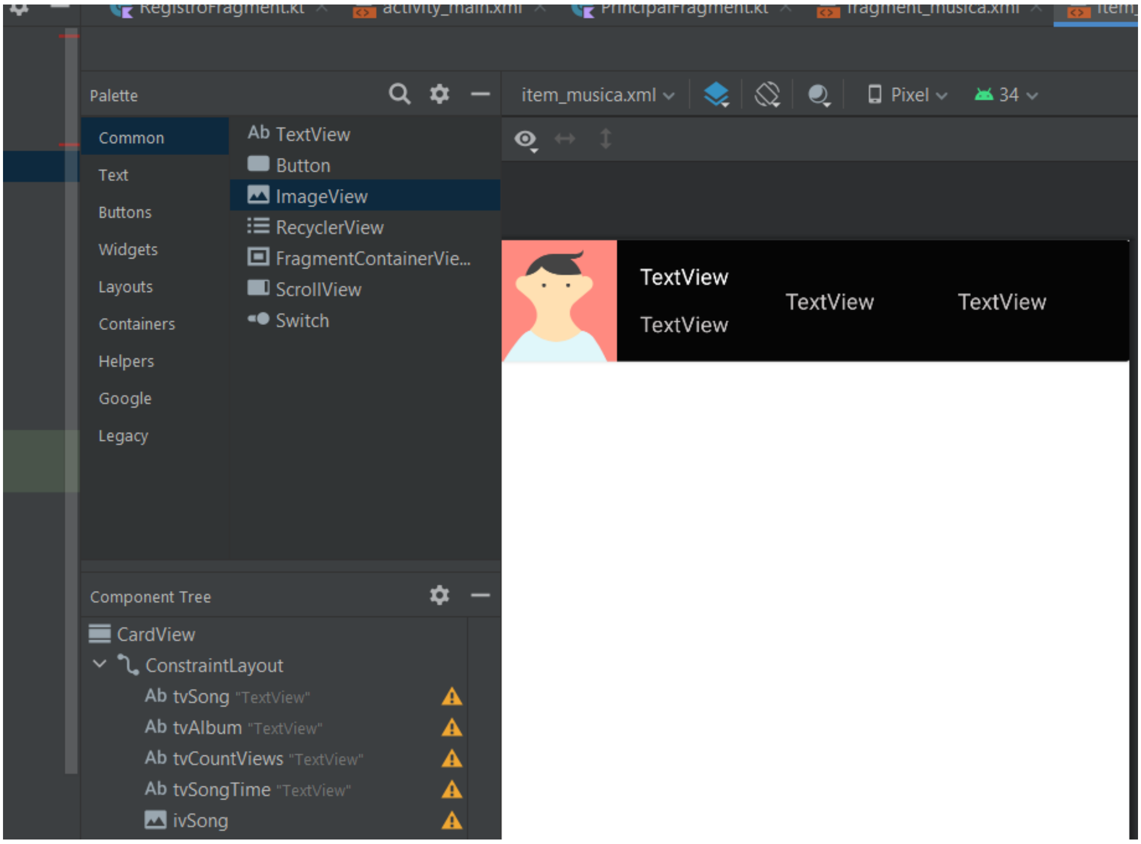
Implement Members

androidx.recyclerview.widget.RecyclerView.Adapter
- onCreateViewHolder(parent: ViewGroup, viewType: Int): SongAd
- onBindViewHolder(holder: SongAdapter.ViewHolder, position: I
- getItemCount(): Int

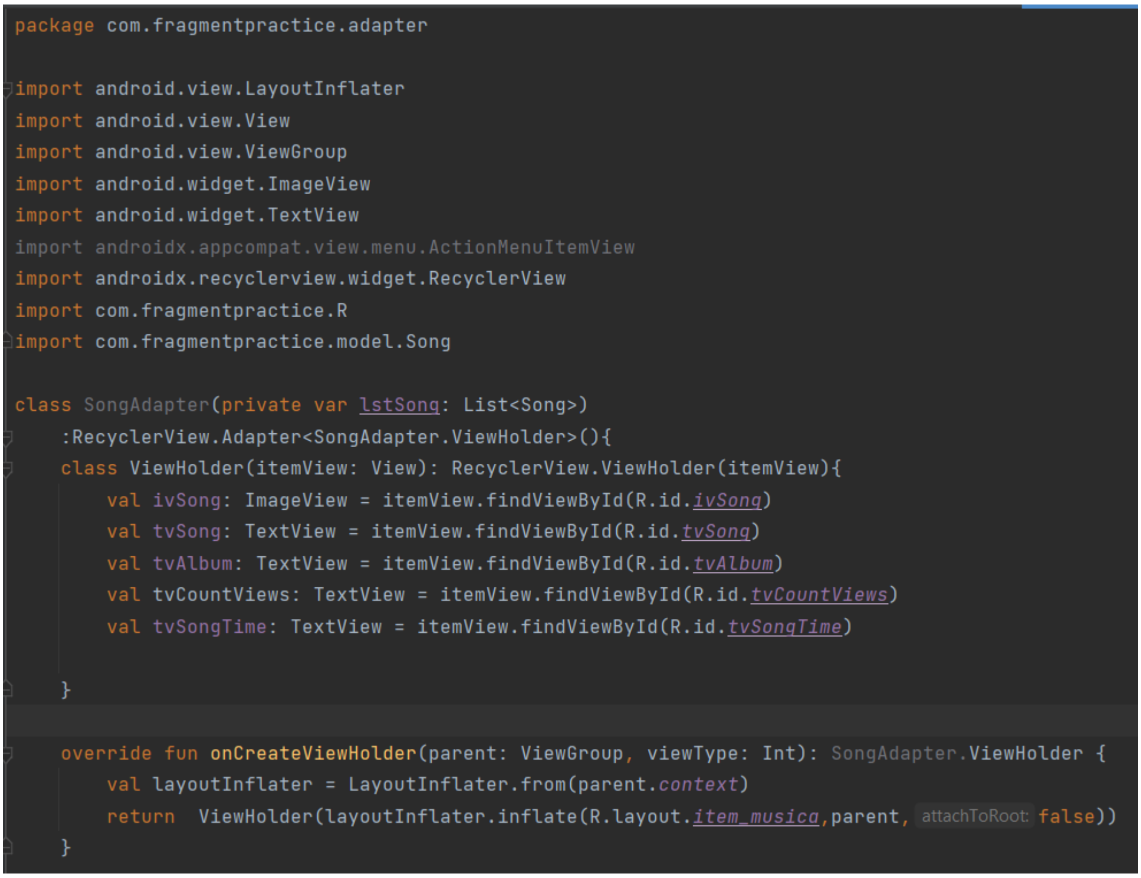


```
package com.fragmentpractice.adapter

import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.fragmentpractice.model.Song

class SongAdapter(private var lstSong: List<Song>)
    :RecyclerView.Adapter<SongAdapter.ViewHolder>(){
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): SongAdapter.ViewHolder {
        TODO( reason: "Not yet implemented")
    }

    override fun onBindViewHolder(holder: SongAdapter.ViewHolder, position: Int) {
        TODO( reason: "Not yet implemented")
    }

    override fun getItemCount(): Int {
        TODO( reason: "Not yet implemented")
    }


}
```

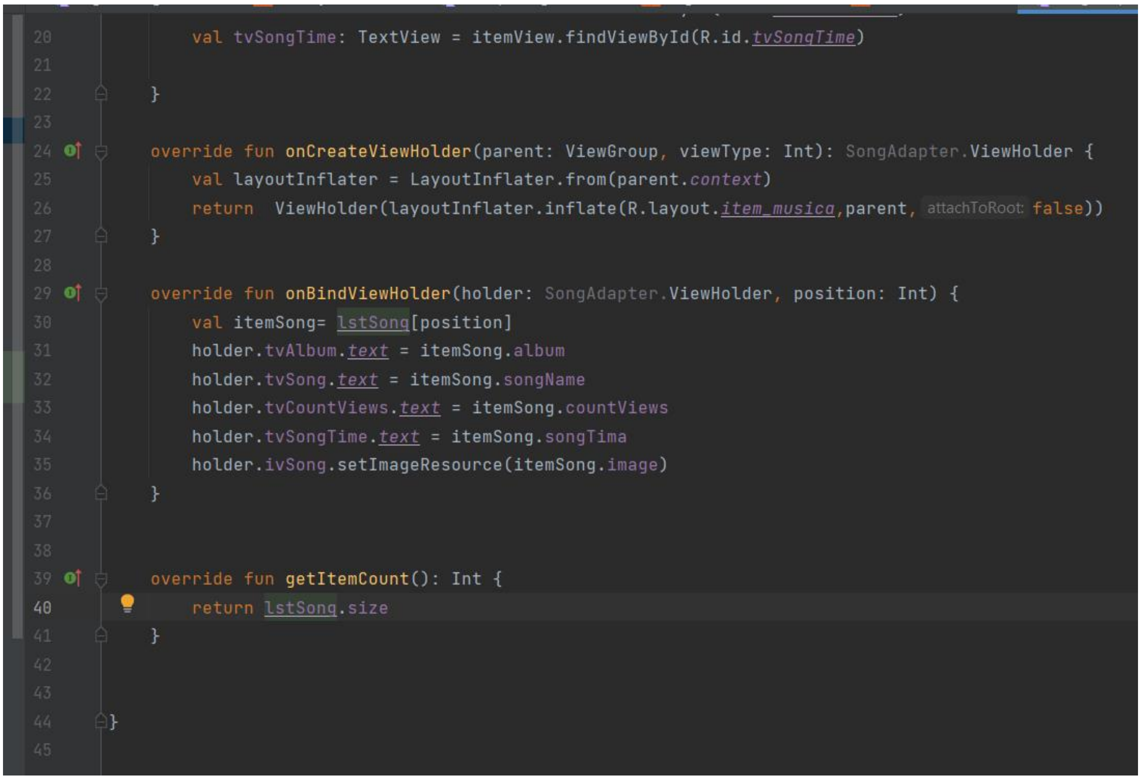
13. Arreglar los constraints:



14. En songadapter

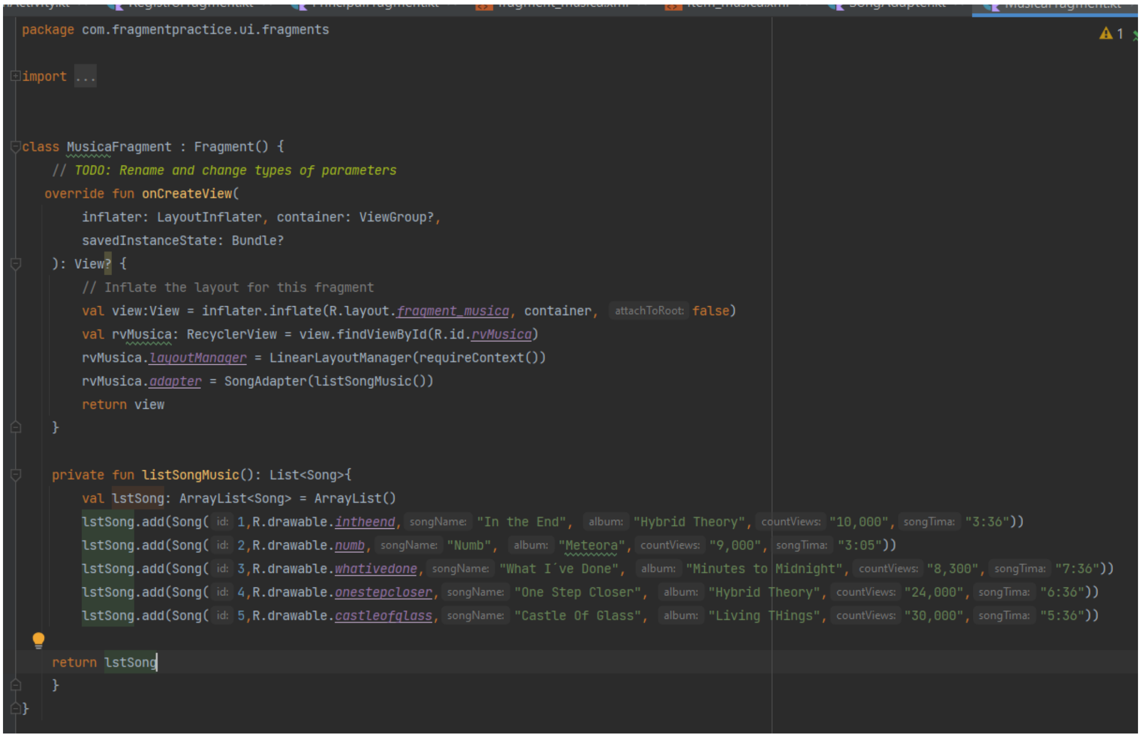


```
package com.fragmentpractice.adapter

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.appcompat.view.menu.ActionMenuItemView
import androidx.recyclerview.widget.RecyclerView
import com.fragmentpractice.R
import com.fragmentpractice.model.Song

class SongAdapter(private var lstSong: List<Song>)
    :RecyclerView.Adapter<SongAdapter.ViewHolder>(){
    class ViewHolder(itemView: View): RecyclerView.ViewHolder(itemView){
        val ivSong: ImageView = itemView.findViewById(R.id.ivSong)
        val tvSong: TextView = itemView.findViewById(R.id.tvSong)
        val tvAlbum: TextView = itemView.findViewById(R.id.tvAlbum)
        val tvCountViews: TextView = itemView.findViewById(R.id.tvCountViews)
        val tvSongTime: TextView = itemView.findViewById(R.id.tvSongTime)

    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): SongAdapter.ViewHolder {
        val layoutInflater = LayoutInflater.from(parent.context)
        return  ViewHolder(layoutInflater.inflate(R.layout.item_musica,parent, attachToRoot: false))
    }
```



```kotlin
        val tvSongTime: TextView = itemView.findViewById(R.id.tvSongTime)

    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): SongAdapter.ViewHolder {
        val layoutInflater = LayoutInflater.from(parent.context)
        return  ViewHolder(layoutInflater.inflate(R.layout.item_musica,parent, attachToRoot: false))
    }

    override fun onBindViewHolder(holder: SongAdapter.ViewHolder, position: Int) {
        val itemSong= lstSong[position]
        holder.tvAlbum.text = itemSong.album
        holder.tvSong.text = itemSong.songName
        holder.tvCountViews.text = itemSong.countViews
        holder.tvSongTime.text = itemSong.songTima
        holder.ivSong.setImageResource(itemSong.image)
    }


    override fun getItemCount(): Int {
        return lstSong.size
    }


}
```

15. En MusicaFragment



```kotlin
package com.fragmentpractice.ui.fragments

import ...


class MusicaFragment : Fragment() {
    // TODO: Rename and change types of parameters
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        val view:View = inflater.inflate(R.layout.fragment_musica, container,  attachToRoot: false)
        val rvMusica: RecyclerView = view.findViewById(R.id.rvMusica)
        rvMusica.layoutManager = LinearLayoutManager(requireContext())
        rvMusica.adapter = SongAdapter(listSongMusic())
        return view
    }

    private fun listSongMusic(): List<Song>{
        val lstSong: ArrayList<Song> = ArrayList()
        lstSong.add(Song( id: 1,R.drawable.intheend, songName: "In the End",  album: "Hybrid Theory", countViews: "10,000", songTima: "3:36"))
        lstSong.add(Song( id: 2,R.drawable.numb, songName: "Numb",  album: "Meteora", countViews: "9,000", songTima: "3:05"))
        lstSong.add(Song( id: 3,R.drawable.whativedone, songName: "What I´ve Done",  album: "Minutes to Midnight", countViews: "8,300", songTima: "7:36"))
        lstSong.add(Song( id: 4,R.drawable.onestepcloser, songName: "One Step Closer",  album: "Hybrid Theory", countViews: "24,000", songTima: "6:36"))
        lstSong.add(Song( id: 5,R.drawable.castleofglass, songName: "Castle Of Glass",  album: "Living THings", countViews: "30,000", songTima: "5:36"))

    return lstSong
    }
}
```