



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

PRACTICA DE SERVICIOS

Curso: Soluciones Móviles I

Docente: Ing. Alberto Johnatan Flor Rodríguez

Alumno:

RODRIGUEZ MAMANI, Juan Rigoberto

(2017057862)

**Tacna – Perú
2020**



INDICE GENERAL

INTRODUCCIÓN	3
I. OBJETIVOS.....	4
1.1. General.....	4
1.2. Específicos.....	4
II. DESARROLLO.....	4
2.1. Practica de Reproductor de Música	4
2.1.1. Diseño	4
2.1.2. Programación.....	12
2.1.3. Resultados.....	17
2.2. Repositorio GitHub	20
CONCLUSIONES	21
BIBLIOGRAFIA.....	22



INTRODUCCIÓN

El Android Studio es la plataforma libre desarrollada por Google, ampliamente utilizada en multitud de dispositivos como móviles, tabletas, TV, wearables e Internet de las cosas. Su expansión ha sido espectacular, siendo el S.O. más utilizado en la actualidad. Tras realizar este curso conocerás los fundamentos del desarrollo de aplicaciones en Android y podrás realizar sencillas aplicaciones, que incluyan los aspectos más importantes y novedosos de esta plataforma.

Android es un sistema operativo que fue creado especialmente para teléfonos con pantalla táctil, los llamados de nueva generación o los inteligentes, las tablets comunes y las que funcionan con líneas telefónica; entrando en esta gama los relojes inteligentes, televisores y algunos aditamentos de los nuevos automóviles, esta empresa de nombre Android Inc, fue respaldada por Google, que en el año 2005 la misma google se apodera comprando por varios millones de dólares, previendo que esta nueva plataforma en la tecnología de avanzada de los estándares en los dispositivos móviles resultaba un buen negocio a futuro.

I. OBJETIVOS

1.1. General

- Realizar la practica encargada del laboratorio de Servicios.

1.2. Específicos

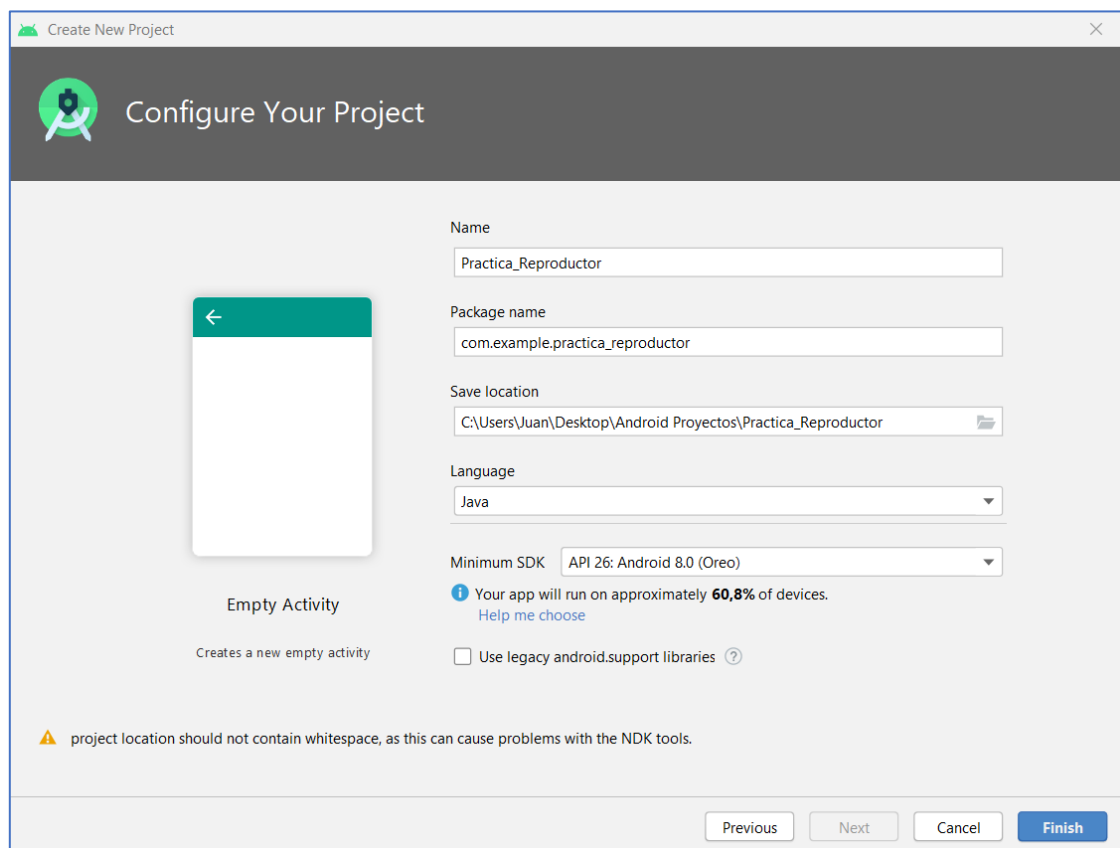
- Investigar la problemática propuesta de la practica encargada.
- Utilizar herramientas de desarrollo en Android Studio.
- Proponer una solución al problema planteado.

II. DESARROLLO

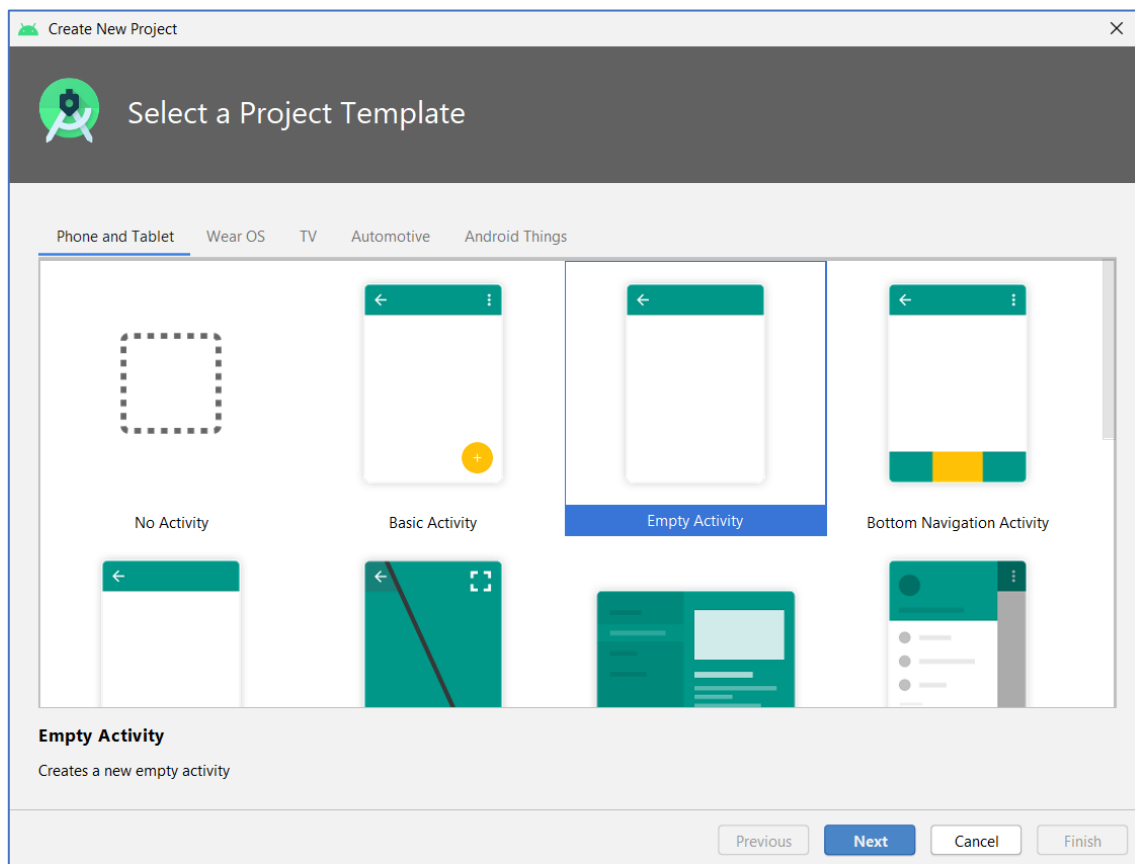
2.1. Practica de Reproductor de Música

2.1.1. Diseño

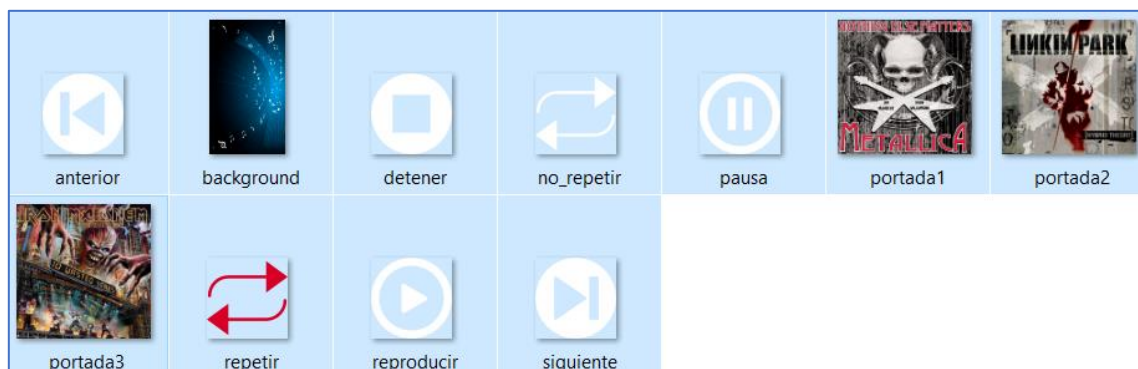
- Lo primero que haremos será crear un nuevo proyecto con el nombre: **Practica_Reproductor** luego elegiremos el Minimum SDK la cual será el **API 26**.



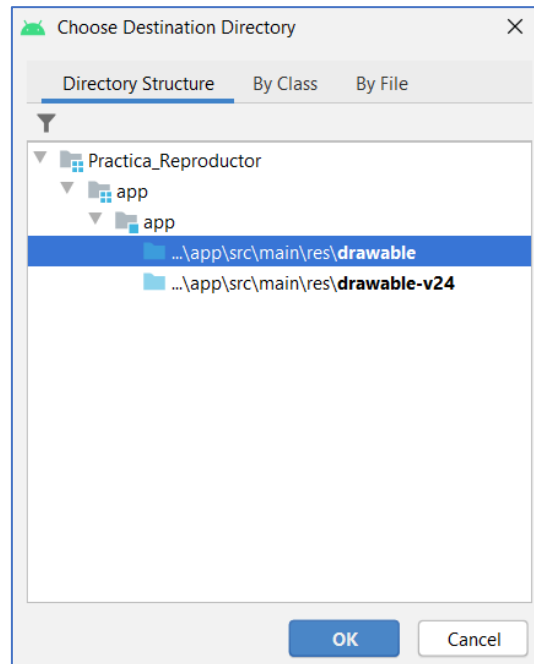
- Seguidamente elegiremos el **Empty Activity**.



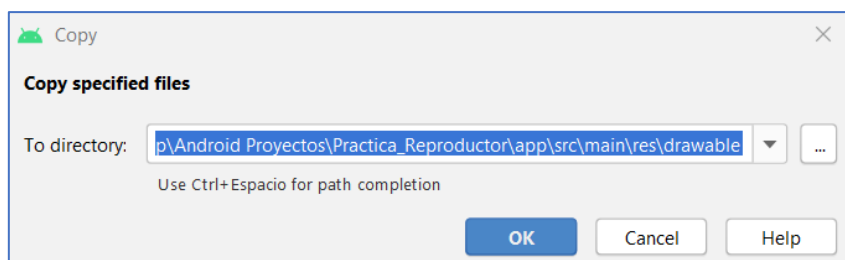
- Ahora para este paso es necesario tener algunas imágenes en nuestra computadora ya que los usaremos como recursos en nuestra aplicación (imágenes de botones, portadas de canciones).



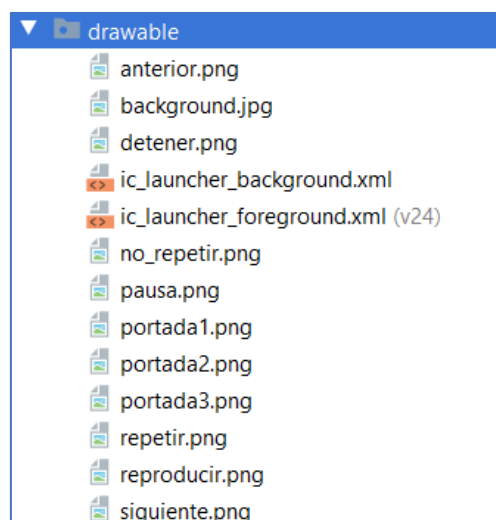
- Copiaremos dichas imágenes y lo pegaremos dentro de la carpeta drawable que se encuentra dentro de nuestro proyecto.



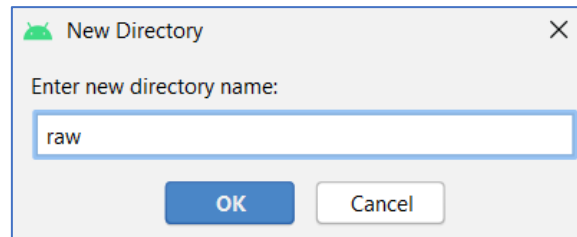
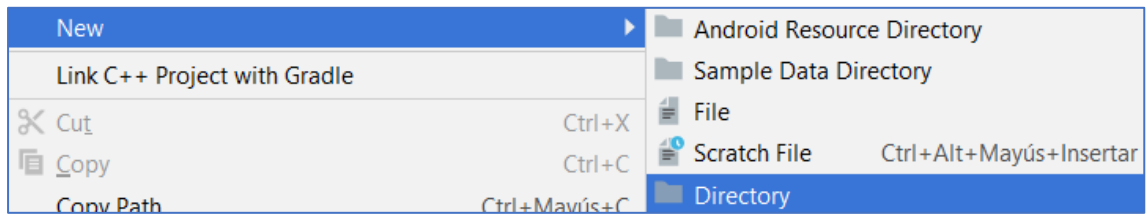
- Nos aparecerá el siguiente recuadro y le daremos clic en OK.



- Así es como debería quedarnos dicha carpeta.



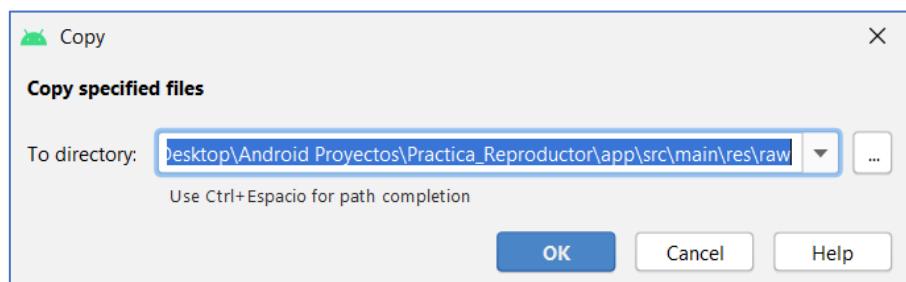
- Seguidamente crearemos un nuevo directorio en nuestro proyecto al cual llamaremos **raw**.



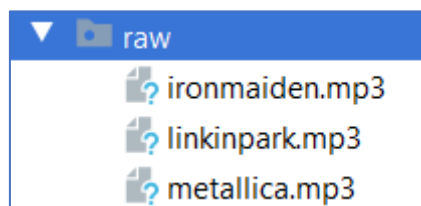
- Dentro de dicho directorio pegaremos las canciones que utilizaremos en nuestra aplicación. Para eso primero ubicaremos las canciones en nuestra computadora, le daremos Ctrl+C.



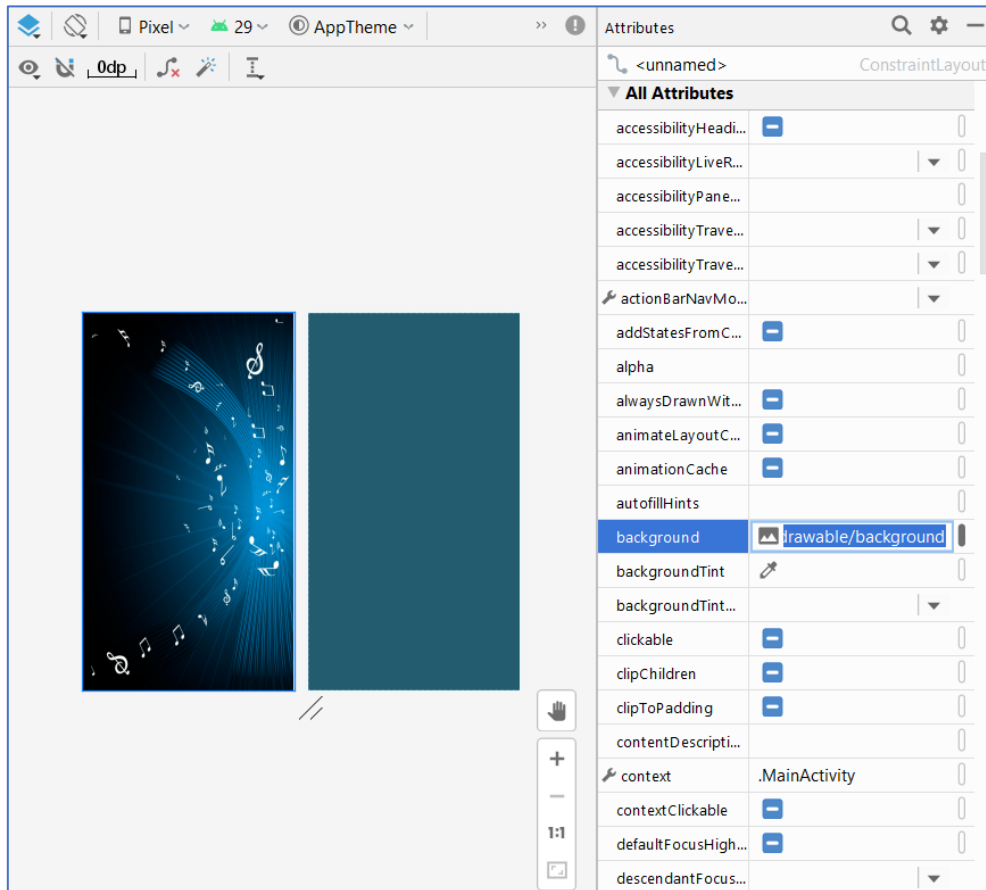
- Nos dirigiremos a nuestra carpeta recién creada llamada **raw** y pegaremos nuestros archivos Ctrl+V.



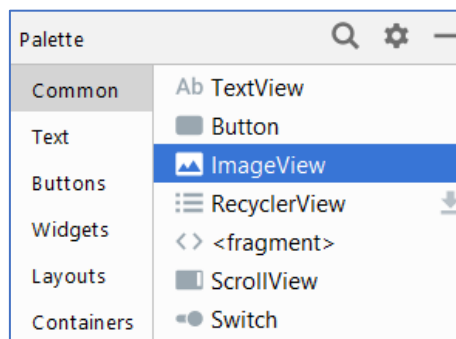
- Nuestra carpeta raw debería quedar de la siguiente forma.



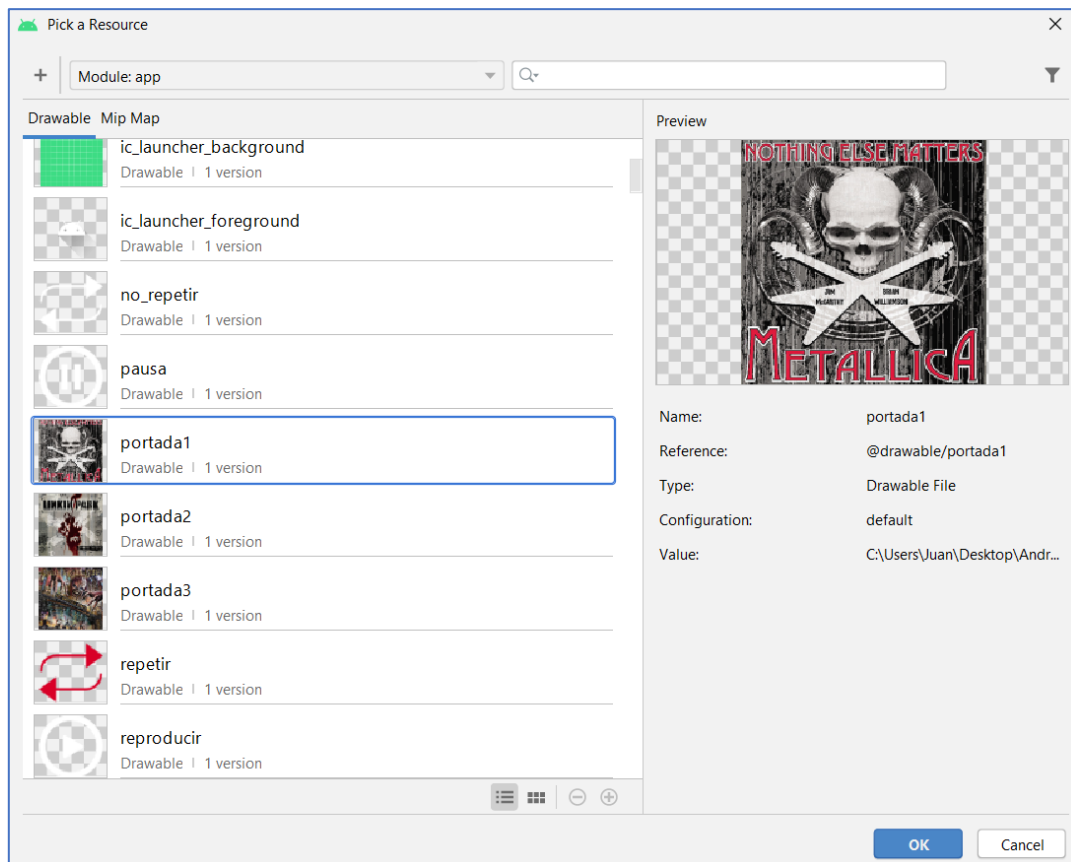
- Acto seguido, vamos a elaborar el diseño de nuestra aplicación, para eso nos dirigimos a nuestro activity y nos dirigimos a los **atributes** de la aplicación y en el apartado llamado **background** indicaremos la ruta de nuestra imagen que se encuentra en nuestro drawable.



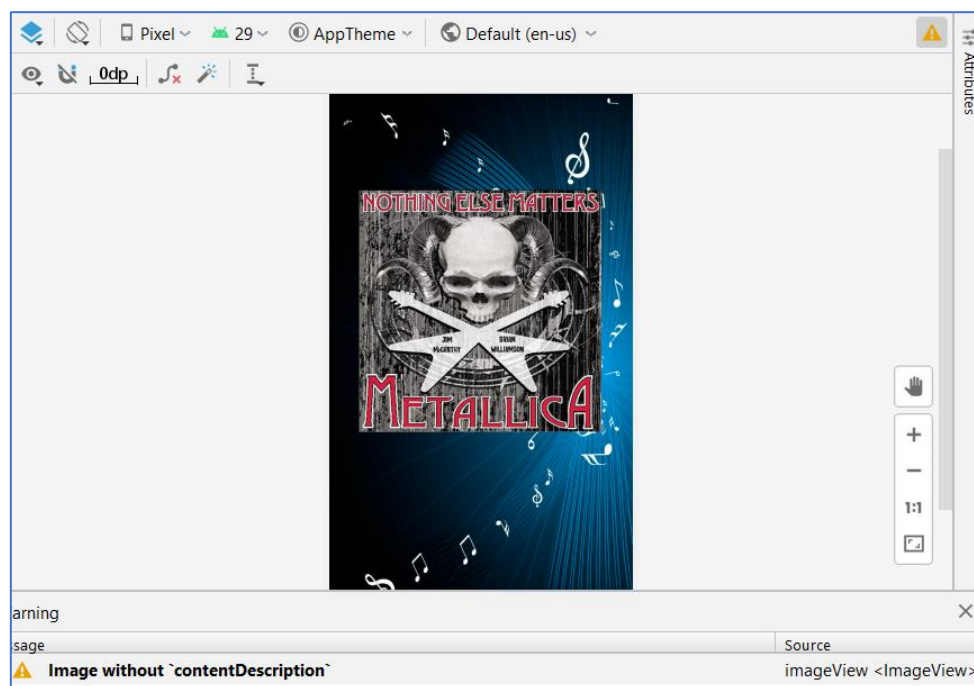
- Luego agregaremos una imagen la cual mostrará la portada de la música en reproducción. Para nos vamos a las herramientas y arrastraremos el componente llamado **ImageView**.



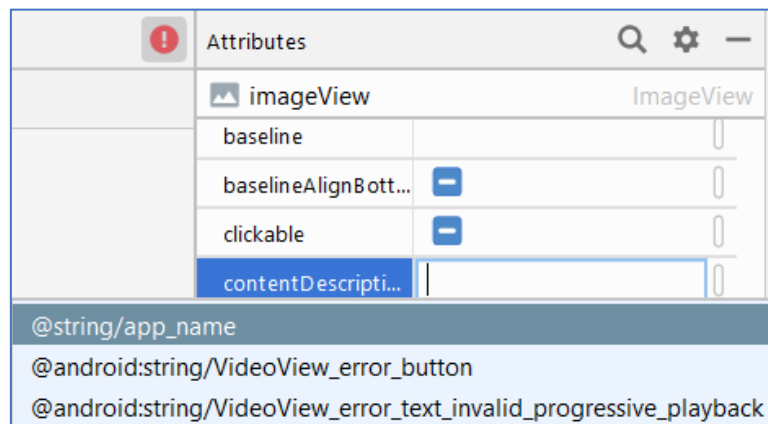
- Ahora elegiremos la imagen que se mostrará en nuestro **ImageView**, en este caso elegiremos la imagen de la **portada1**.



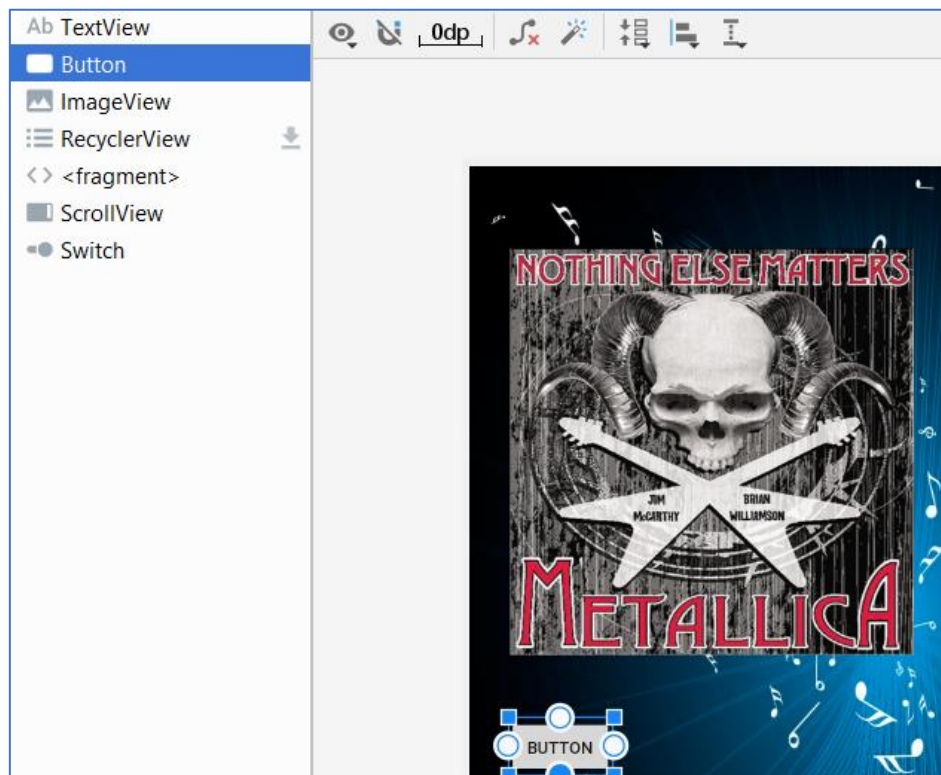
- Luego acomodaremos la posición de la imagen de tal manera que nos quede así.



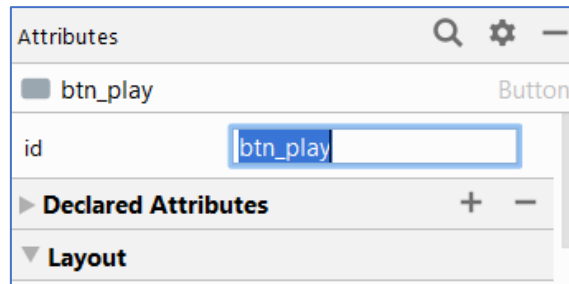
- Luego de haber hecho esto, lo más probable es que Android nos muestre una advertencia de diseño, esto lo solucionaremos agregando una referencia de tipo String en su atributo **contentDescription**. Para esto debemos ir a los **attributes** y en el apartado de contentDescription y presionaremos **Ctrl+Space** luego nos aparecerán varias referencias del propio Android en el cual nosotros elegiremos la que se llama **@string/app_name** y con esto habremos solucionado dicha advertencia.



- Una vez que hayamos agregado la imagen de la portada, lo que haremos ahora será agregar los botones de nuestro reproductor. Para esto nos vamos a las herramientas y buscaremos el componente llamado **Button** y lo arrastraremos a nuestro diseño.



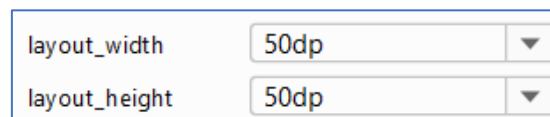
- Seleccionaremos nuestro **button** y en el apartado de **attributes** cambiaremos si **id** por **btn_play**.



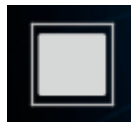
- Acto seguido modificaremos el aspecto de nuestro botón, para eso en attributes nos dirigiremos a su layout.



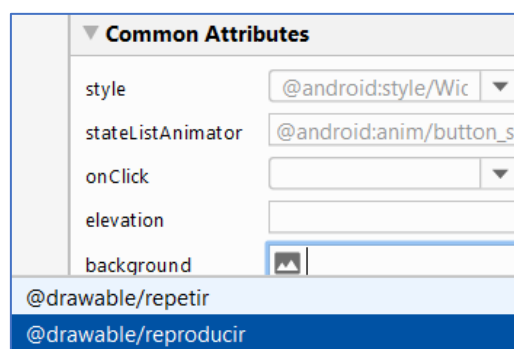
- Y cambiaremos sus valores por estos nuevos.



- De tal forma que el nuevo diseño del botón será el siguiente.



- Ahora vamos a agregar una imagen para eso nos vamos al **background** del botón.



- Y elegiremos la imagen de nuestro drawable llamado **reproducir**.



- Debemos repetir el mismo proceso para los otros botones, no debemos olvidar de cambiar el **id** de los nuevos botones por los siguientes.

btn_play, **btn_stop**, **btn_repetir**, **btn_anterior** y **btn_siguiente**.



- Y con esto habremos terminado de diseñar nuestra aplicación, la cuál se encuentra lista para ser programada.

2.1.2. Programación

- En este paso comenzaremos a programar nuestra aplicación, como primer paso nos dirigiremos a nuestro **MainActivity** y declararemos las siguientes variables.

```
Button play_pause, btn_repetir;  
MediaPlayer mp;  
ImageView iv;  
int repetir = 2, posicion = 0;  
  
MediaPlayer vectormp [] = new MediaPlayer [3];
```

- Dentro de nuestro **onCreate** agregaremos el siguiente código. Básicamente lo que hace es enlazar los botones de nuestro diseño con las respectivas variables antes creadas, además estamos estableciendo las posiciones de las músicas con las que vamos a trabajar.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    play_pause = (Button)findViewById(R.id.btn_play);
    btn_repetir = (Button)findViewById(R.id.btn_repetir);
    iv = (ImageView)findViewById(R.id.imageView);

    vectormp[0] = MediaPlayer.create( context: this, R.raw.metallica);
    vectormp[1] = MediaPlayer.create( context: this, R.raw.Linkinpark);
    vectormp[2] = MediaPlayer.create( context: this, R.raw.ironmaiden);
}
```

- Seguidamente crearemos nuestros métodos que darán funcionalidad a nuestra aplicación.
- **Método PlayPause.** Con este método lo que estamos diciendo es que cambie el estado del botón a **Pausa** si se está reproduciendo una música o que cambie a **Play** si la música está en pausa.

```
//Metodo para PlayPause
public void PlayPause(View view){
    if(vectormp[posicion].isPlaying()){
        vectormp[posicion].pause();
        play_pause.setBackgroundResource(R.drawable.reproducir);
        Toast.makeText( context: this, text: "Pausa", Toast.LENGTH_SHORT).show();
    }else{
        vectormp[posicion].start();
        play_pause.setBackgroundResource(R.drawable.pausa);
        Toast.makeText( context: this, text: "Play", Toast.LENGTH_SHORT).show();
    }
}
```

- **Método Stop.** Este método lo que hará será indicar a la aplicación que deje de reproducir la pista actual, además cambia la apariencia del botón de Stop a Play como también de la imagen que se está mostrando.

```
//Metodo Stop
public void Stop(View view){
    if(vectormp[posicion] != null){
        vectormp[posicion].stop();
        vectormp[0] = MediaPlayer.create( context: this, R.raw.metallica);
        vectormp[1] = MediaPlayer.create( context: this, R.raw.Linkinpark);
        vectormp[2] = MediaPlayer.create( context: this, R.raw.ironmaiden);
        posicion = 0;
        play_pause.setBackgroundResource(R.drawable.reproducir);
        iv.setImageResource(R.drawable.portada1);
        Toast.makeText( context: this, text: "Stop", Toast.LENGTH_SHORT).show();
    }
}
```

- **Método Repetir.** Este método lo que hará será repetir la pista actual además de cambiar la imagen de los botones Repetir y No Repetir.

```
//Metodo Repetir
public void Repetir(View view){
    if(repetir == 1){
        btn_repetir.setBackgroundResource(R.drawable.no_repetir);
        Toast.makeText( context: this, text: "No repetir", Toast.LENGTH_SHORT).show();
        vectormp[posicion].setLooping(false);
        repetir = 2;
    }else{
        btn_repetir.setBackgroundResource(R.drawable.repetir);
        Toast.makeText( context: this, text: "Repetir", Toast.LENGTH_SHORT).show();
        vectormp[posicion].setLooping(true);
        repetir = 1;
    }
}
```

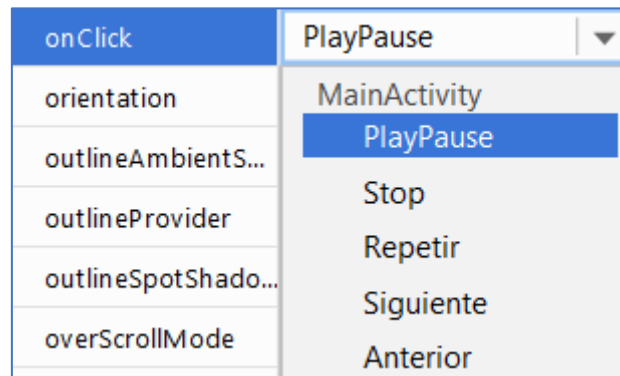
- **Método Siguiente.** Con este método lograremos que la aplicación vea la posición actual de la canción y pase a la siguiente, además de cambiar la imagen de la portada según la posición de la canción que se está reproduciendo.

```
//Metodo Siguiente
public void Siguiente(View view){
    if(posicion < vectormp.length -1){
        if(vectormp[posicion].isPlaying()){
            vectormp[posicion].stop();
            posicion++;
            vectormp[posicion].start();
            if(posicion == 0){
                iv.setImageResource(R.drawable.portada1);
            }else if(posicion == 1){
                iv.setImageResource(R.drawable.portada2);
            }else if(posicion == 2){
                iv.setImageResource(R.drawable.portada3);
            }
        }else{
            posicion++;
            if(posicion == 0){
                iv.setImageResource(R.drawable.portada1);
            }else if(posicion == 1){
                iv.setImageResource(R.drawable.portada2);
            }else if(posicion == 2){
                iv.setImageResource(R.drawable.portada3);
            }
        }
    }else{
        Toast.makeText( context: this, text: "No hay mas canciones", Toast.LENGTH_SHORT).show();
    }
}
```

- **Método Anterior.** Este método es similar a la anterior con la única diferencia que, en lugar de sumar posiciones, vamos a restarlas.

```
//Metodo Anterior
public void Anterior(View view){
    if(posicion >= 1){
        if(vectormp[posicion].isPlaying()){
            vectormp[posicion].stop();
            vectormp[0] = MediaPlayer.create( context: this, R.raw.metallica);
            vectormp[1] = MediaPlayer.create( context: this, R.raw.Linkinpark);
            vectormp[2] = MediaPlayer.create( context: this, R.raw.ironmaiden);
            posicion--;
            if(posicion == 0){
                iv.setImageResource(R.drawable.portada1);
            }else if(posicion == 1){
                iv.setImageResource(R.drawable.portada2);
            }else if(posicion == 2){
                iv.setImageResource(R.drawable.portada3);
            }
            vectormp[posicion].start();
        }else{
            posicion--;
            if(posicion == 0){
                iv.setImageResource(R.drawable.portada1);
            }else if(posicion == 1){
                iv.setImageResource(R.drawable.portada2);
            }else if(posicion == 2){
                iv.setImageResource(R.drawable.portada3);
            }
        }
    }else{
        Toast.makeText( context: this, text: "No hay mas canciones", Toast.LENGTH_SHORT).show();
    }
}
```


- Para finalizar el desarrollo de la aplicación ya solo nos quedará asignar los métodos creados con los botones antes diseñados. Para esto nos dirigiremos al diseño de nuestro **activity** y haremos click sobre el botón Play, y en el apartado de **attributes** buscaremos la opción de **onClick** y elegiremos el método **PlayPause**.



- Haremos los mismos pasos para los demás botones y luego de haber hecho esto, podremos compilar nuestra aplicación.



- ¡Y listo! Ahora podremos disfrutar de nuestra aplicación de música y también podemos probar que todos los botones se encuentran totalmente funcionales.

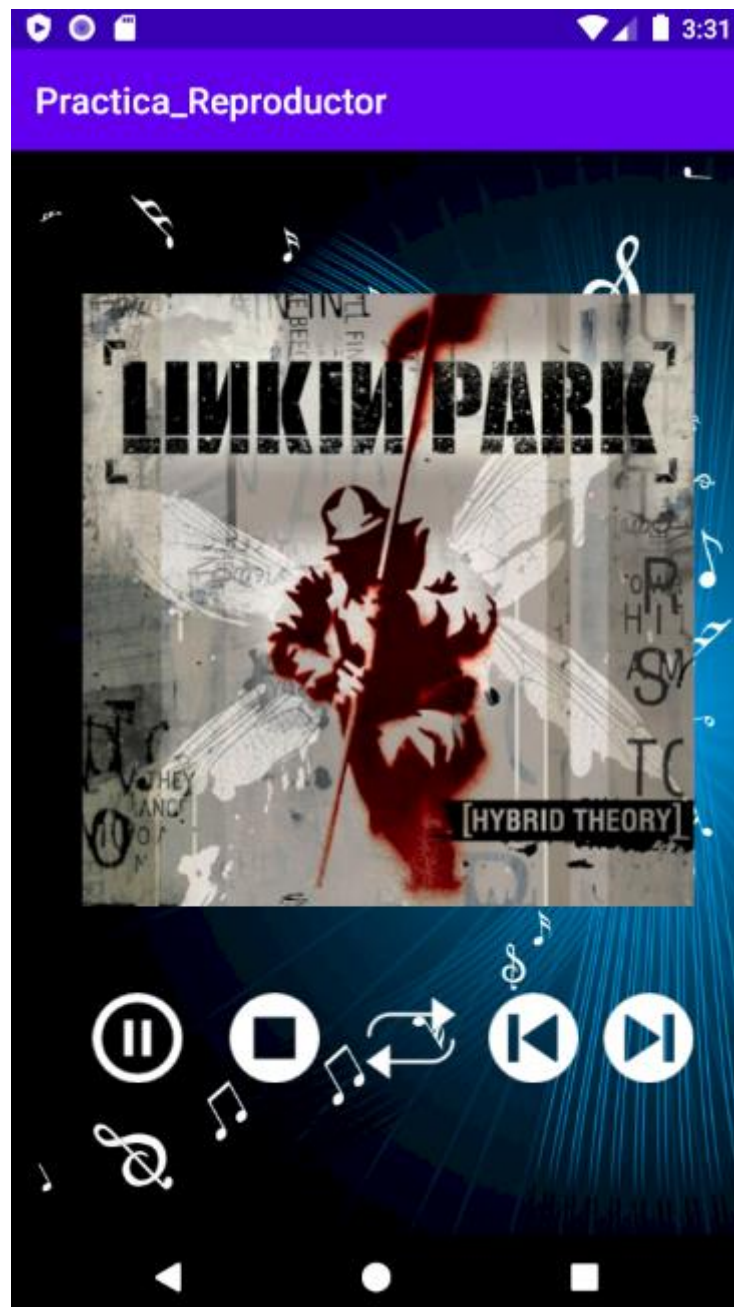
2.1.3. Resultados

- Canción 1: Metallica - Nothing Else Matters.



Publicación	20 de abril de 1992
Formato	Sencillo en CD · casete · vinilo
Grabación	30 de mayo de 1991 <ul style="list-style-type: none">• One on One Studios (North Hollywood, California, E.E. U.U.)
Género(s)	Heavy metal · Power ballad
Duración	6:29

- Canción 2: Linkin Park - In The End.



Publicación	Worldwide: December 2000 UK: August 2001
Formato	Sencillo en CD, Sencillo en DVD, Descarga digital, Streaming
Grabación	1998 - 1999: (Nueva Orleans, Estados Unidos)
Género(s)	Nu metal, ¹ rap rock, rock alternativo, hard rock
Duración	3:37

- Canción 3: Iron Maiden - Wasted Years.



Publicación	6 de septiembre 1986
Formato	Vinilo de 7", 12" y CD
Género(s)	Heavy metal
Duración	5:07
Discográfica	EMI
Autor(es)	Adrian Smith
Productor(es)	Martin Birch

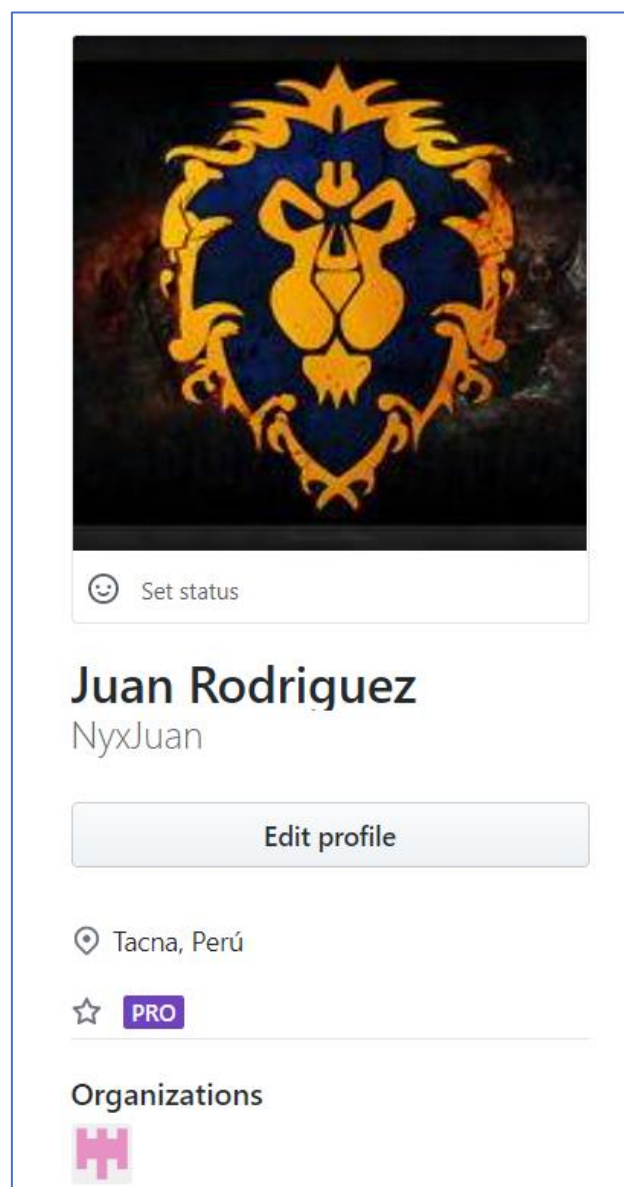
2.2. Repositorio GitHub

Practica V Servicios.

NOTA:

- En caso de que falle la aplicación, intentar limpiar el Emulador dándole clic en **WIPE DATA** o intentar probar con otro emulador.
- Otorgar permisos a la aplicación.
- Intentar correr con el emulador utilizando el **API 26** en adelante.

[https://github.com/NyxJuan/Moviles I Practica V ReproductorMusica](https://github.com/NyxJuan/Moviles_I_Practica_V_ReproductorMusica)





CONCLUSIONES

- Se logró realizar la practica encargada del laboratorio de Servicios.
- Se pudo Investigar la problemática propuesta de la practica encargada, con la finalidad de poder proponer posibles soluciones.
- Se utilizó varias herramientas para el desarrollo de la aplicación en Android Studio, los cuales fueron útiles para el desarrollo de la solución.
- Se pudo proponer la solución del problema planteado, esto de acuerdo con la problemática planteada del laboratorio de servicios.



BIBLIOGRAFIA

- [1]. Gutiérrez, G. (2019). *Android*. Recuperado el 11 de mayo de 2020, de Concepto Definición. Sitio web: <https://conceptodefinicion.de/android/>

- [2]. Google. (2019). *Google Cloud Platform*. Recuperado el 11 de junio de 2020, de Google. Sitio web: <https://console.cloud.google.com/>

- [3]. ShareIcon. (2018). *Share Icon*. Recuperado el 11 de junio de 2020, de ShareIcon. Sitio web: <https://www.shareicon.net/>

- [4]. La Geekipedia De Ernesto. (2018). Curso Android desde cero #40 | Reproductor de música. Recuperado el 12 de junio de 2020, de YouTube. Sitio web: <https://www.youtube.com/watch?v=IAfGGzZoMwk>