

Actividad | 3 | Diseño de Aplicación (Parte final).

Ingeniería del conocimiento.

Ingeniería en Desarrollo de Software.



TUTOR: Eduardo Israel Castillo García.

ALUMNO: Carlos Ariel Nicolini

FECHA: 14/04/2025

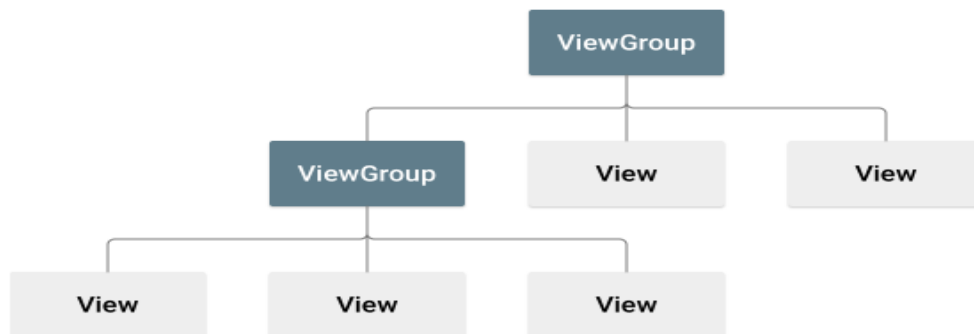
Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
• Interfaz.....	6
• Codificación	8
• Prueba de aplicación.....	11
Conclusión.....	12
Referencias.....	13

Introducción

Un diseño define la estructura de una interfaz de usuario en tu aplicación, como en un activity.

Todos los elementos de diseño se construyen mediante una jerarquía de View y ViewGroup objetos. Un View suele dibujar algo que el usuario puede ver y para interactuar. Un ViewGroup es un contenedor invisible que define la estructura de diseño de View y otros ViewGroup objetos, como se muestra en la siguiente imagen



Los objetos View a menudo se denominan widgets y pueden ser uno de muchas subclases, como Button o TextView. Los objetos ViewGroup suelen llamarse diseños y pueden ser uno de ellos, que proporcionan una estructura de diseño diferente, como LinearLayout o ConstraintLayout.

Puedes declarar un diseño de dos maneras:

- Declara elementos de la IU en XML: Android proporciona un archivo XML simple que corresponde a las clases y subclases View, como los de los widgets y diseños. También puedes usar las APIs de Android Studio Editor de diseño para compilar tu XML, con una interfaz de arrastrar y soltar.
- Crea una instancia de elementos de diseño durante el tiempo de ejecución: Tu app puede crear objetos View y ViewGroup, y manipulan sus propiedades de forma programática.

El framework de Android te brinda la flexibilidad de usar uno o ambos de estos métodos para compilar la IU de tu app.

Descripción

Contextualización:

El buen fin se acerca, por lo que se le solicita a un ingeniero en desarrollo de software el realizar una aplicación móvil en Android. Esta debe permitir manejar una galería de ringtones.

Actividad:

Continuar con el proyecto de la Actividad 2. En esta ocasión, lograr que el usuario pueda compartir el audio desde la aplicación. En este sentido, cada vez que alguno comparta deberá aparecer una notificación indicando que ha ganado un punto.

Para ponernos en contexto, en la actividad 1, se nos pidió instalar Android Studio y configurarlo para su correcto uso. Una vez hecho esto, se deberá de crear una aplicación que al descargarse contenga 10 audios en formato .mp3. En nuestro trabajo solo agregamos 5 canciones

En la actividad 2 se nos solicitó que se debería añadir la característica a la interfaz de que el usuario podrá compartir cada uno de los audios a otros dispositivos vía redes sociales, correo o Bluetooth.

Justificación

Esta actividad es la última de esta materia que a mi parecer es muy interesante y grata de realizar. Hemos podido seguir utilizando y aprendiendo del programa Android Studio que me trae fascinado con su forma de manejarse y me encanta al realizar la prueba en emulador, ver como esa idea que se tenía realizarse y funcionar tal vez de manera simple (no soy un conocedor amplio del sistema).

En esta ocasión continuaremos con lo realizado en los trabajos anteriores, que se creó un splash screen donde aparece una imagen y un texto que tienen una animación que me encanto, para luego de 6 segundos dirigirnos a la activity principal del programa donde podemos reproducir 5 canciones. También se había agregado un botón de compartir el cual nos permitía compartirlo a través de 4 redes sociales. En esta oportunidad agregaremos un mensaje al presionar el botón de bluetooth, el cual había sido agregado en la actividad anterior pero que no tenía funcionalidad.

Me divertí mucho en este trabajo, si tarde un poco por problemas de salud, pero este trabajo me ayudo a despejarme un poco y divertirme con mi hijo al realizarlo con ideas que él me proporciono.

Agradecido desde siempre con el profesor, los compañeros y todo el personal que siempre está ayudando en todo.

Este trabajo fue subido al siguiente enlace de GitHub

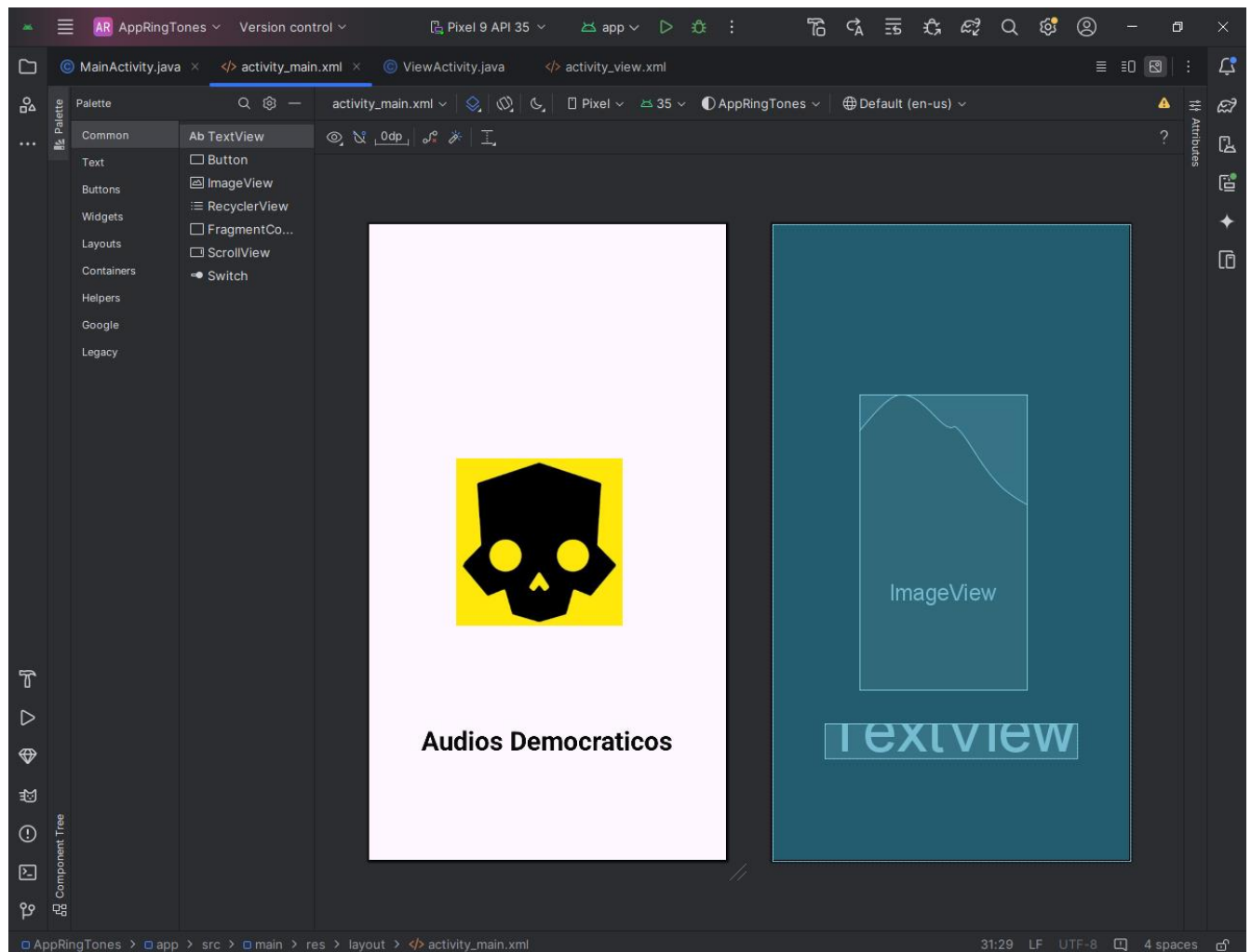
<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-II>

Desarrollo

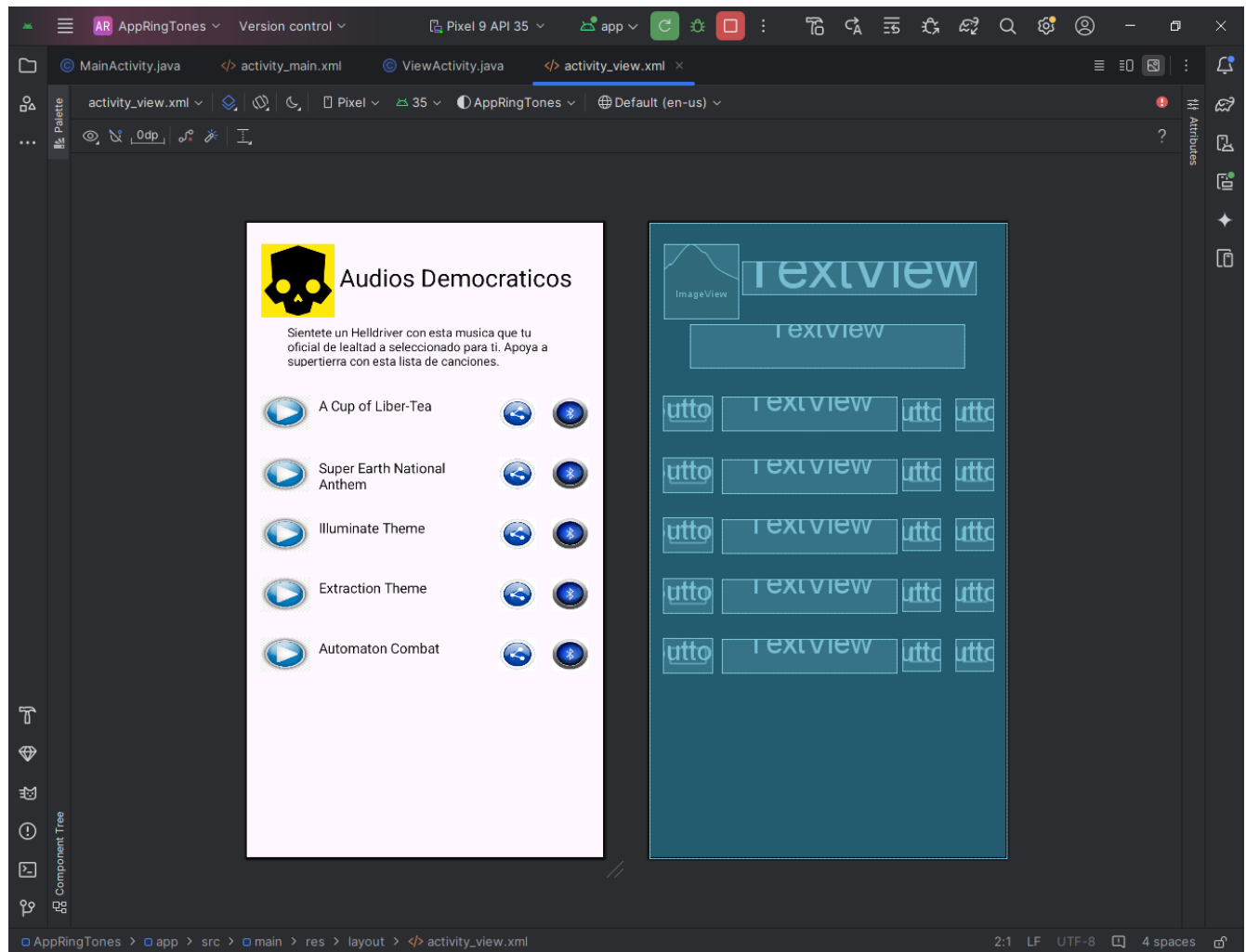
Interfaz

En este ejercicio final continuaremos con el proyecto realizado en las clases anteriores. En esta ocasión se agregara una acción al botón de bluetooth.

A continuación mostramos como estaba realizado el Splash Screen (activity_main.xml), la cual se compone gráficamente de un icono y un texto.



En la actividad principal (activity_view.xml) se muestra ya el botón de bluetooth pero hasta antes de este trabajo no contaba con una acción al presionarlo. Se agregó un texto informativo al presionarlo y un botón de si o no con sus respectivas acciones.



Codificación

A continuación mostraremos las imágenes de los códigos realizados y una breve explicación de que realizan como fue visto en la clase.

ViewActivity.java

En el configuramos para que realice lo siguiente:

- Se declaran los botones de share y se les asigna una variable Button.
- Se configura los botones blue para que cuando el usuario haga click en ellos, se muestre un mensaje con dos botones (si o no). Si se presiona si, la actividad se cierra y si presiona no, el cuadro se cierra y permanece en la pantalla.

```
2 usages
Button play1, play2, play3, play4, play5, share1, share2, share3, share4, share5, blue1, blue2, blue3, blue4, blue5;

4 usages
MediaPlayer mp1, mp2, mp3, mp4, mp5;

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

    play1 = (Button)findViewById(R.id.play1);
    play2 = (Button)findViewById(R.id.play2);
    play3 = (Button)findViewById(R.id.play3);
    play4 = (Button)findViewById(R.id.play4);
    play5 = (Button)findViewById(R.id.play5);
    share1 = (Button)findViewById(R.id.share1);
    share2 = (Button)findViewById(R.id.share2);
    share3 = (Button)findViewById(R.id.share3);
    share4 = (Button)findViewById(R.id.share4);
    share5 = (Button)findViewById(R.id.share5);
    blue1 = (Button)findViewById(R.id.blue1);
    blue2 = (Button)findViewById(R.id.blue2);
    blue3 = (Button)findViewById(R.id.blue3);
    blue4 = (Button)findViewById(R.id.blue4);
    blue5 = (Button)findViewById(R.id.blue5);
}
```



```

blue1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder alerta = new AlertDialog.Builder(context: ViewActivity.this);
        alerta.setMessage("Muchas Felicidades, otra victoria para el lado correcto de la historia, has ganado un punto, sigue part
        .setCancelable(false)
        .setPositiveButton(text: "Si", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        })
        .setNegativeButton(text: "No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        });
        AlertDialog titulo = alerta.create();
        titulo.setTitle("FELICIDADES");
        titulo.show();
    }
});

```

Activity_view.xml

En el configuramos para que realice lo siguiente:

- Se agregaron los botones blue con sus códigos, se le defino un ancho de 43dp y una altura de 37dp.
- Se le puso un icono de imagen que está en la ruta @drawable/blue.
- Está centrado horizontalmente pero desplazado hacia la derecha y ligeramente hacia la parte superior.

Botón blue1:

```

<Button
    android:id="@+id/blue1"
    android:layout_width="43dp"
    android:layout_height="37dp"
    android:foreground="@drawable/blue"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.956"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.29" />

```

Botón blue2:

```
<Button
    android:id="@+id/blue2"
    android:layout_width="43dp"
    android:layout_height="37dp"
    android:foreground="@drawable/blue"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.956"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.39" />
```

Botón blue3:

```
<Button
    android:id="@+id/blue3"
    android:layout_width="43dp"
    android:layout_height="37dp"
    android:foreground="@drawable/blue"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.956"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.49" />
```

Botón blue4:

```
<Button
    android:id="@+id/blue4"
    android:layout_width="43dp"
    android:layout_height="37dp"
    android:foreground="@drawable/blue"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.956"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.59" />
```

Botón blue5:

```
<Button
    android:id="@+id/blue5"
    android:layout_width="43dp"
    android:layout_height="37dp"
    android:foreground="@drawable/blue"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.956"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.69" />
```

Prueba de aplicación

En esta parte probamos la aplicación realizada y adjuntaremos imágenes de su comportamiento. Al presionar cualquier botón de bluetooth nos saldrá el mensaje proporcionado en el código como se muestra en la siguiente imagen. Al presionar si se cerrara la aplicación, al presionar no, se quedara en ella.



Conclusión

En esta materia y este ejercicio nos han enseñado lo importante que es no solo conocer el funcionamiento del Android Studio y el idioma que usa al realizar un desarrollo, sino que un correcto diseño de las aplicaciones es crucial para el correcto funcionamiento de la app, para la experiencia del usuario final, para tener una mayor visibilidad de la app y para la retención de usuarios al sentirse a gusto y contentos con ella. Un buen diseño ayuda y facilita la navegación, mejora su usabilidad y hace que la aplicación sea más atractiva.

Si es verdad que se necesita tener bien en mente primero que es lo que se desea realizar y hacerlo de una manera correcta, que sea agradable al uso y vista, que no sea muy complicada de usar y que sea una experiencia grata para el usuario para que siga usándola y la recomiende. Para esto es necesario tener un amplio conocimiento en el lenguaje y sus formas, tener conocimiento del software en este caso Android Studio el cual me encanta, ojala podamos en las futuras clases poder seguir utilizándolo.

Espero que el trabajo cumpla con lo solicitado, desde ya muchas gracias.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-II>

Referencias

Introducción a Android Studio. (n.d.). Android Developers. Retrieved April 20, 2025, from

<https://developer.android.com/studio/intro?hl=es-419>

Helldivers 2. (n.d.). Logopedia; Fandom, Inc. Retrieved April 20, 2025, from

https://logos.fandom.com/wiki/Helldivers_2

Aspectos fundamentales de la app. (n.d.). Android Developers. Retrieved April 21, 2025, from

<https://developer.android.com/guide/components/fundamentals?hl=es-419>

Diseños en vistas. (n.d.). Android Developers. Retrieved April 21, 2025, from

<https://developer.android.com/develop/ui/views/layout/declaring-layout?hl=es-419>