Actividad [#3] – [Diseño de Aplicación (Parte final)]

[Desarrollo de Aplicaciones Móviles II]

Ingeniería en Desarrollo de Software

Tutor: Humberto Jesús Ortega Vázquez

Alumno: Alan David López Rojas

Fecha:17/03/2023

Índice

Introducción

Las aplicaciones móviles en la actualidad son bastante reconocidas por los diferentes usos que estas prestan a los usuarios, entre ese amplio campo de las aplicaciones existe una categoría que son las redes sociales (Facebook, Twitter, etc.) o medios de comunicación, mediante estás aplicaciones se puede interactuar con otras personas con diferentes fines, entre ellos, uno de los más importantes compartir información o archivos , por lo que es una excelente manera de dar a conocer un producto o servicio.

A su vez utilizar la barra de notificaciones del dispositivo es una manera fácil y sencilla de informar al usuario: inmediatamente visibles, rápidas de implementar, las notificaciones son una herramienta eficiente para el desarrollador, porque brindan información urgente y fácil de ver, y representan acciones que se envían al usuario, una notificación está compuesta por un icono, un título en una línea superior, una segunda línea de texto que utiliza una tipografía de tamaño inferior para el mensaje, así como una fecha.

Descripción

Actualmente se observa la enorme interacción de las personas en las redes sociales, por lo cual se pide en este proyecto final ,que la aplicación que se ha ido desarrollando durante este curso de Desarrollo de aplicaciones móviles 2 para el Buen Fin, pueda compartir los archivos de audio de Ring Tones, mediante un botón de compartir, pueda ser enviado mediante diferentes vías, correo, Bluetooth y redes sociales, esto con el fin de dar a conocer más usuarios la aplicación y el nombre de la empresa. Por este motivo se tiene ha desarrollado la aplicación con el fin de compartir los archivos de audio además que se le ha puesto la acción de mandar una notificación al usuario que le informa que ha ganado un punto, esto con el fin de motivar a los usuarios a seguir compartiendo para ganar mas puntos y estos se puedan intercambiar después por algún producto o premio, permitirá que la información de la aplicación se propague rápido y más gente conozca la aplicación.

Justificación

El presente proyecto tiene como principal objetivo dar a conocer al usuario que al compartir archivos de audio de la aplicación por medio de redes sociales podrá ganar puntos, que tendrán una función la cual se especificara por la empresa, este proyecto se realiza por que existe la necesidad de implementar una función en la aplicación , que le permita a los usuarios compartir información para estar conectadas mediante redes sociales y a su vez puedan sentir que la empresa les brinda un beneficio por el uso de la aplicación. Teniendo en cuenta la importancia de compartir la información en cualquier momento y en el lugar que sea, Se considera que el desarrollo de este proyecto servirá para poder aprender a desarrollar una aplicación para compartir información y como plataforma para el desarrollo posterior de soluciones basadas en la misma lógica.

Investigación

● ¿Cómo poner notificaciones en una aplicación de Android?

Nos dirigimos a la clase Java o Kotlin de nuestro MainActivity e inicializamos nuestro botón asignándolo el evento OnClickListener. Asignamos dicho evento al botón, mediante el método setOnClickListener , tal y como se muestra a continuación:

```
public class MainActivity extends AppCompatActivity {
    Button mostrarNotificacion;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mostrarNotificacion = (Button)findViewById(R.id.btnMostrarNotificacion);
        mostrarNotificacion.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
          }
        });
    }
}
```

Ahora, procedemos a crear nuestra notificación. Para ello usaremos la clase Notification.Builder y la clase NotificationManager .

```
NotificationCompat.Builder mBuilder;
NotificationManager mNotifyMgr =(NotificationManager)
getApplicationContext().getSystemService(NOTIFICATION_SERVICE);
```

La notificación consta de varios elementos visuales como son: el icono de la misma, el título y la descripción. Existe además un componente que gestiona el Activity o pantalla que se lanza cuando un usuario hace clic sobre la notificación, hablamos del componente PendingIntent :

```
int icono = R.mipmap.ic_launcher;
Intent intent = new Intent(MainActivity.this, MensajeActivity.class);
PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0,intent, 0);
```

En este caso, en el código anterior podemos observar que se ha definido un Activity, llamado MensajeActivity y que será el Activity que se lance cuando el usuario haga clic en la notificación

A continuación, vamos a diseñar nuestra notificación; para asignamos al objeto mBuilder anteriormente creado al método NotificationCompat.Builder() , al cual le debemos pasar como parámetro el contexto desde donde se va a lanzar la notificación, en nuestro caso, usaremos getApplicationContext () quedando la siguiente línea de código:

```
mBuilder = new NotificationCompat.Builder(getApplicationContext());
```

A continuación, vamos a proceder a añadirle a la notificación el icono, los textos y la acción:

setContentIntent (pendingIntent): Este método nos permite que al hacer click sobre la notificación se abra el Activity que definimos en el PendingIntent anteriormente creado.

setSmallIcon (icono): Este método añade a nuestra notificación el icono que inicializamos anteriormente.

setContentTitle (“Openwebinars”) : Añade el título de la notificación.

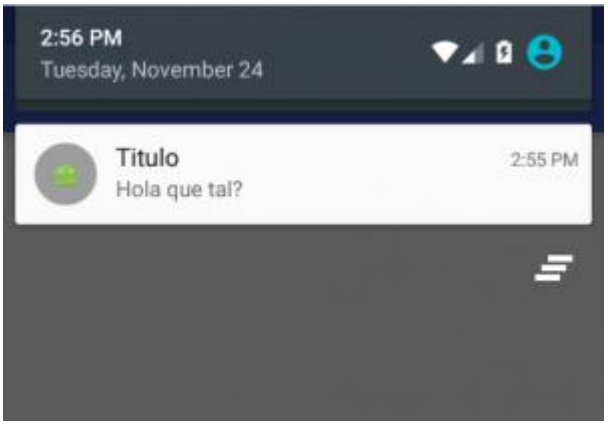
setContentText(“Tienes 1 nuevo mensaje”): Añade el mensaje descriptivo a la notificación.

setVibrate(new long[] {100, 250, 100, 500}): Esto hace que cuando salte la notificación vibre nuestro teléfono.

setAutoCancel(true): este método hace que al hacer clic sobre la notificación esta desaparezca.

Al añadir todos estos métodos nos queda un bloque de código como el siguiente:

```
mBuilder =new NotificationCompat.Builder(getApplicationContext())
        .setContentIntent(pendingIntent)
        .setSmallIcon(icono)
        .setContentTitle("Titulo")
        .setContentText("Hola que tal?")
        .setVibrate(new long[] {100, 250, 100, 500})
        .setAutoCancel(true);
```



Ya sólo falta crear la notificación. Para ello invocamos al NotificationManager que creamos arriba cuyo variable declaramos con el nombre mNotifyMgr y le añadimos el método . notify () al cual hay que pasarle como parámetros un número entero que actúa como identificador único de la notificación,

en nuestro caso le pondremos 1; y el segundo parámetro NotificationCompat.Builder creado al comienzo con el nombre de variable mBuilder , quedando la siguiente línea de código:

```
mNotifyMgr.notify(1, mBuilder.build());
```

El código complete queda así:

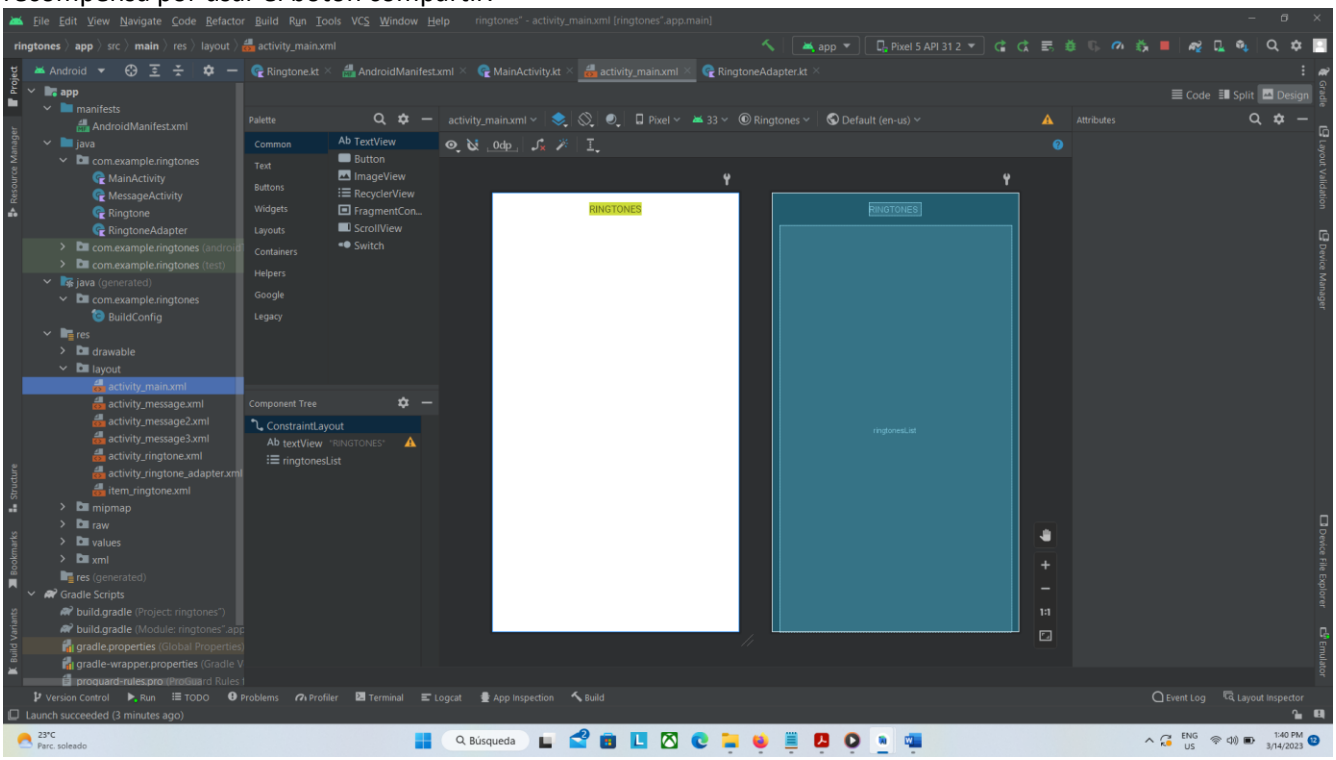
```
public class MainActivity extends AppCompatActivity {
   Button mostrarNotificacion;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      mostrarNotificacion=(Button)findViewById(R.id.btnMostrarNotificacion);
      mostrarNotificacion.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            NotificationCompat.Builder mBuilder;
            NotificationManager mNotifyMgr =(NotificationManager)
getApplicationContext().getSystemService(NOTIFICATION_SERVICE);

            int icono = R.mipmap.ic_launcher;
            Intent i=new Intent(MainActivity.this, MensajeActivity.class);
            PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, i, 0);

            mBuilder =new NotificationCompat.Builder(getApplicationContext())
                        .setContentIntent(pendingIntent)
                        .setSmallIcon(icono)
                        .setContentTitle("Titulo")
                        .setContentText("Hola que tal?")
                        .setVibrate(new long[] {100, 250, 100, 500})
                        .setAutoCancel(true);



            mNotifyMgr.notify(1, mBuilder.build());

         }
      });
   }
}[1]
```

[1] Campos Miguel, Cómo hacer notificaciones push en Android fácil, 2016

Desarrollo

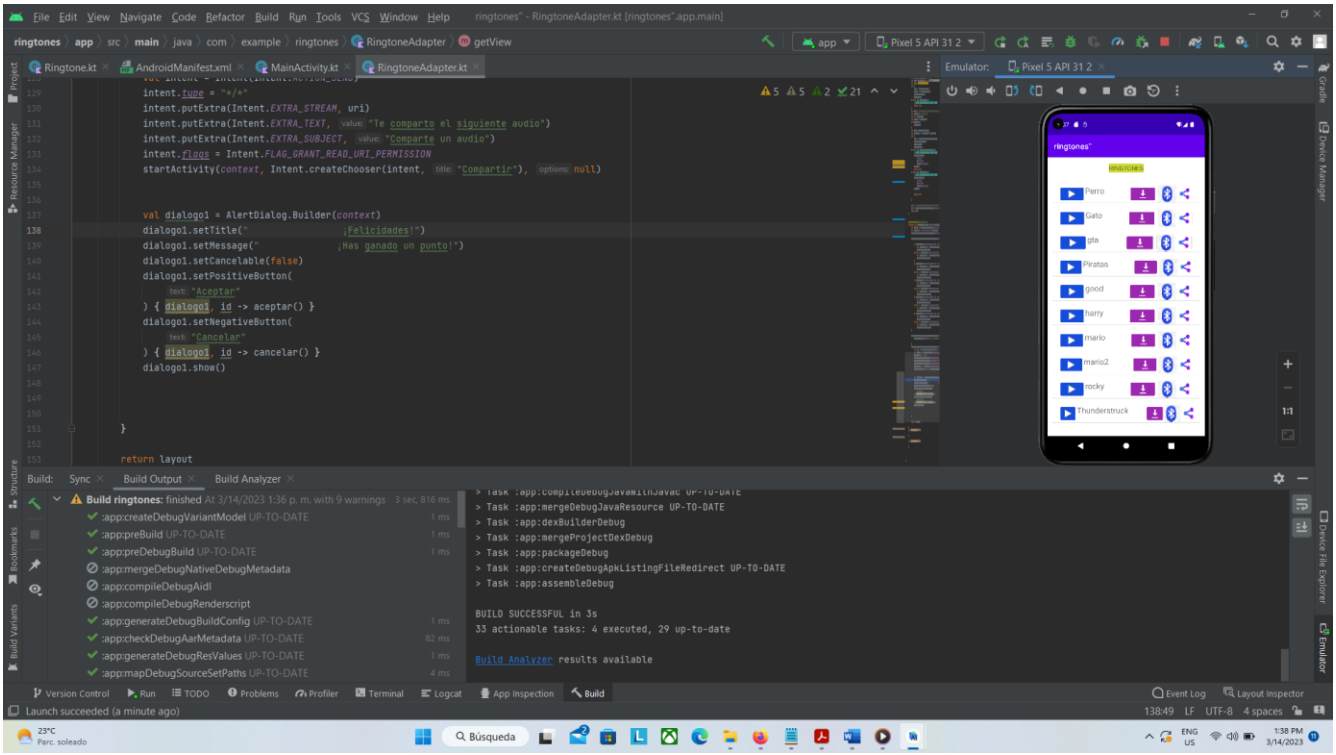
❖ Interfaz

Se utiliza la misma interfaz de las actividades anteriores, solo agregando el cuadro de notificación de la recompensa por usar el botón compartir.



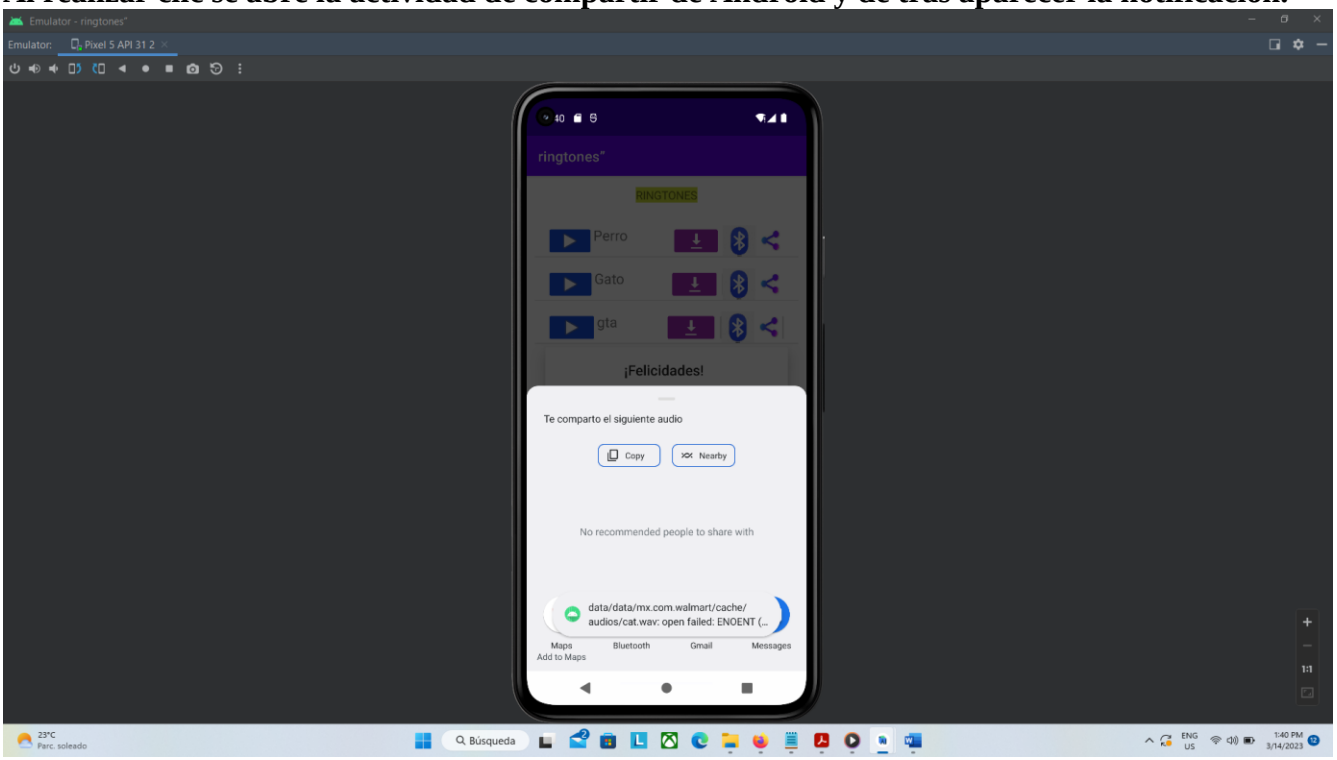
❖ Codificación

El código de la notificación se realiza en el setOnClickListener del mismo botón de compartir, así se realizan ambas acciones.

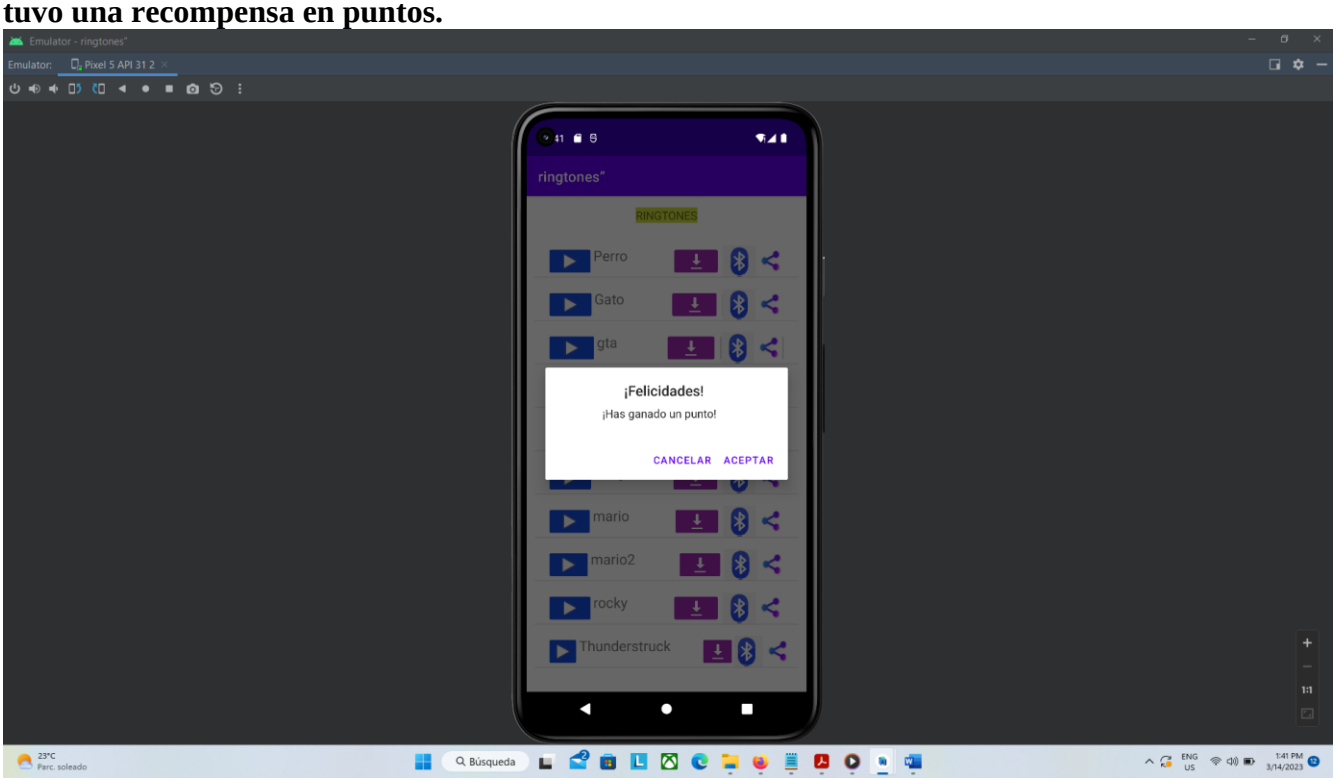


❖ Prueba de la aplicación

Al realizar clic se abre la actividad de compartir de Android y de tras aparecer la notificación.



Después de compartir se puede visualizar el mensaje que se envía al usuario que le indica que tuvo una recompensa en puntos.



Conclusión

La aplicación desarrollada permite al usuario escuchar Ringtones y descargarlos a su dispositivo móvil para que puedan usarlos, además de compartir archivos de audio por medio de internet o Bluetooth y le permite al usuario saber que ha tenido un premio por haber compartido el ringtone, durante esta actividad pude comprender que desarrollar aplicaciones móviles, con funcionalidad de compartir archivos, es una herramienta cada vez más utilizada para dar conocer a información de primera mano, a la hora que sea y en cualquier lugar, ya necesaria genera gran expectativa en los usuarios de este tipo de tecnologías por la posibilidad de tener y compartir la información en cualquier momento.

Sobre el desarrollo con kotlin, es fácil de aprender, sobre todo teniendo buena base con Java, tiene mucha información en la web y da muchas posibilidades al desarrollador, esto me ha obligado a estudiar nuevas herramientas y pensar diferentes opciones para realizar una misma tarea, lo cual creo que también es de gran importancia para mi carrera como desarrollador.

Respecto a la conclusión personal que saco, debo decir que la actividad sirve para acercarnos un poco las nuevas tecnologías de la información abarcan cada vez un campo más grande y aprender a desarrollar software que haga a las personas sentir nuevas experiencias y el acceso a la información es de vital importancia.

Se anexa link de Google drive con el proyecto en formato RAR, ya que es muy grande para subir a la plataforma.
https://drive.google.com/file/d/1gvInTAbER4wXn9h7XSLQbcDc8LsIpQ57/view?usp=share_link
Lo subí a One Drive debido a que Google lo detecta como aplicación para acceder a datos, lo bloqueo pero ya se solicitó revisión.
https://1drv.ms/u/s!Aj0cqUIiha6LhLURNPtJRl9ze2jxbA?e=aZuvp6

Bibliografía

Campos Miguel, Cómo hacer notificaciones push en Android fácil, 2016,En línea: https://openwebinars.net/blog/como-hacer-notificaciones-push-en-android-facil/

Codelaby, Enviar notificaciones simples en Android usando Kotlin (Android O), 2022, En línea: https://dev4phones.wordpress.com/2022/05/05/enviar-notificaciones-simples-en-android-usando-kotlin-android-o/

Developers, Cómo crear una notificación, En línea: https://developer.android.com/training/notify-user/build-notification?hl=es-419