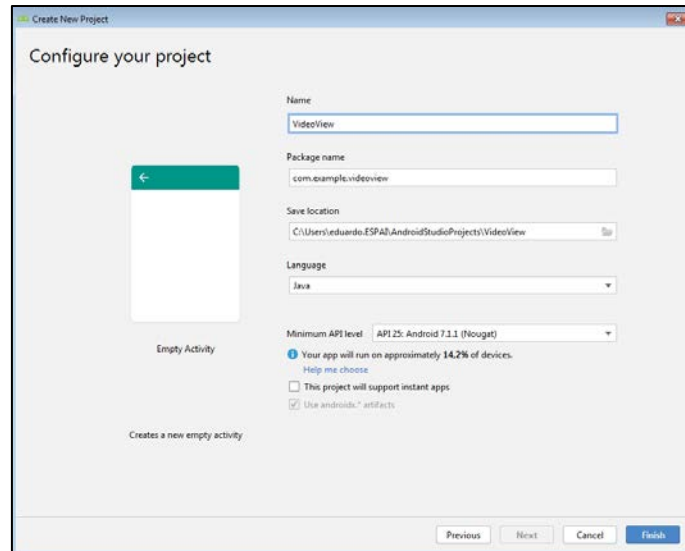


PRACTICA 10: VIDEOVIEW

Parte I: Reproducir un vídeo con VideoView

Paso 1. Crea una nueva aplicación de nombre VideoView.



Paso 2. Reemplaza el fichero `res/layout/activity_main.xml` por:

```
activity_main.xml x
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <VideoView
        android:id="@+id/surface_view"
        android:layout_width="640px"
        android:layout_height="480px"/>
</LinearLayout>
```

Paso 3. Reemplaza el siguiente código en la clase `MainActivity`:

```
public class MainActivity extends AppCompatActivity {

    private VideoView mVideoView;

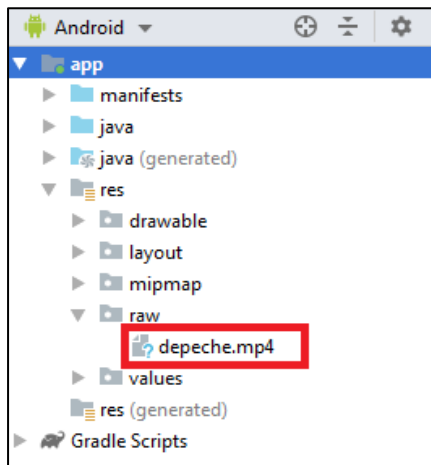
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mVideoView = (VideoView) findViewById(R.id.surface_view);

        String uriPath = "android.resource://" + getPackageName() + "/" + R.raw.depeche;
        Uri uri = Uri.parse(uriPath);
        mVideoView.setVideoURI(uri);
        //mVideoView.setVideoPath("file:///sdcard/depeche.mp4");

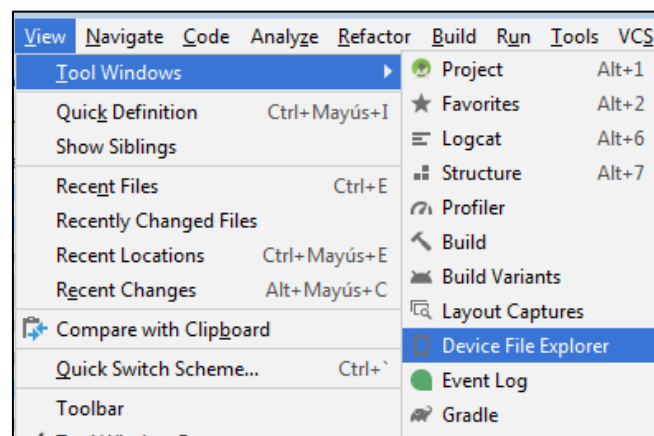
        mVideoView.start();
        mVideoView.requestFocus();
    }
}
```

En el parámetro del método `setVideoPath()` estamos indicando un fichero local.

Paso 4. Busca un fichero de video en codificación mp4. Renombra este fichero a *video.mp4*. Introdúcelo en el proyecto en el directorio raw.

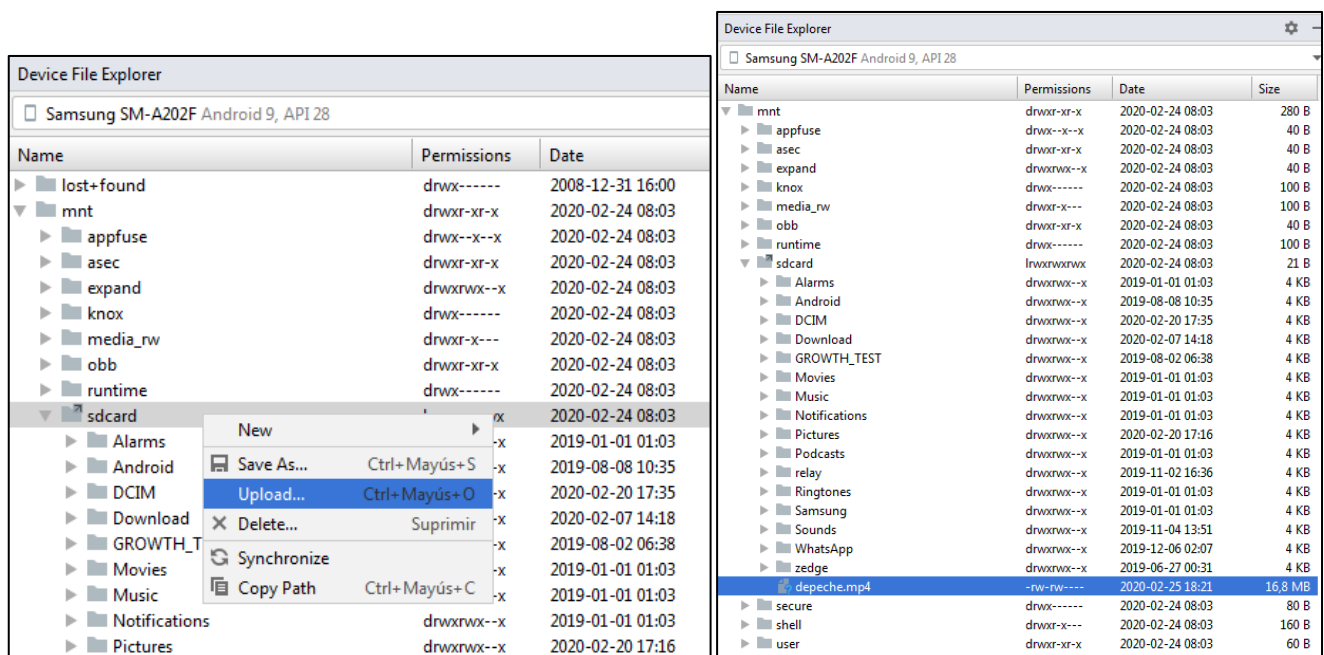


Paso 5. Es necesario almacenar un fichero de video en el simulador. Para ello, utiliza la opción del menú *View > Tools Windows/Device File Explorer*. Te mostrará el sistema de ficheros del emulador (para ello el emulador debe de estar ejecutándose).



Paso 6. Selecciona la carpeta *mnt/sdcard* y utiliza el botón donde aparece un teléfono con una flecha para almacenar un nuevo fichero. Indica el fichero *video.mp4*.

NOTA: el dispositivo ha de disponer de almacenamiento externo.



Paso 7. Ejecuta la aplicación y observa el resultado.



Paso 8. Modifica el fichero xml para que el vídeo aparezca centrado y ocupe toda la pantalla del teléfono, tal y como se muestra en la imagen de arriba

```
activity_main.xml x
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <VideoView
        android:id="@+id/surface_view"
        android:layout_alignParentTop="true"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```



Paso 9. Añade la siguiente línea antes de la llamada al método `start()`;
`videoView.setMediaController(new MediaController(this));`

```
public class MainActivity extends AppCompatActivity {  
  
    private VideoView mVideoView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mVideoView = (VideoView) findViewById(R.id.surface_view);  
  
        String uriPath = "android.resource://" + getPackageName() + "/" + R.raw.depeche;  
        Uri uri = Uri.parse(uriPath);  
        mVideoView.setVideoURI(uri);  
        //mVideoView.setVideoPath("file:///sdcard/depeche.mp4");  
  
        mVideoView.setMediaController(new MediaController(context, this));  
  
        mVideoView.start();  
        mVideoView.requestFocus();  
    }  
}
```

De esta forma permitimos que el usuario pueda controlar la reproducción del vídeo mediante el objeto `MediaController`.

Paso 10. Observa el resultado.

