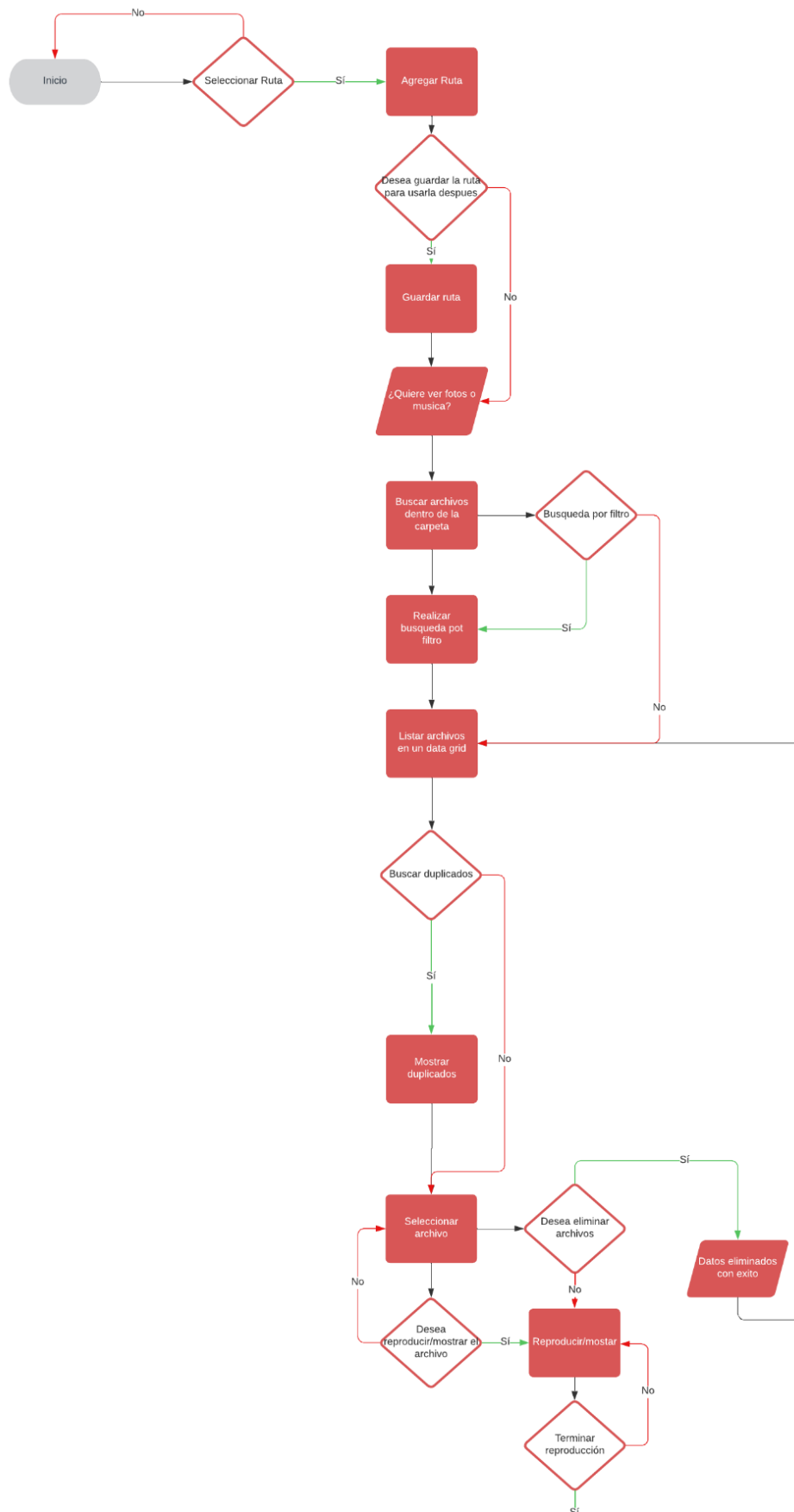
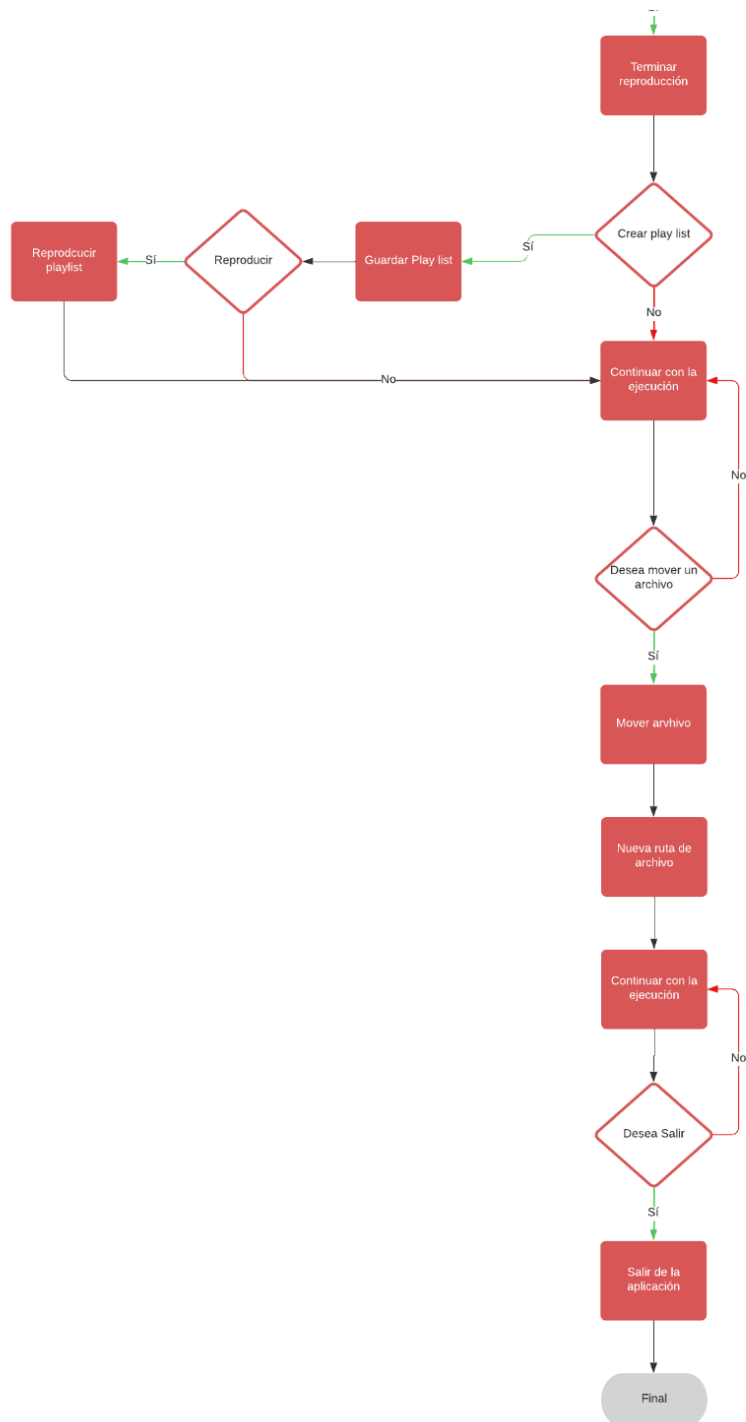


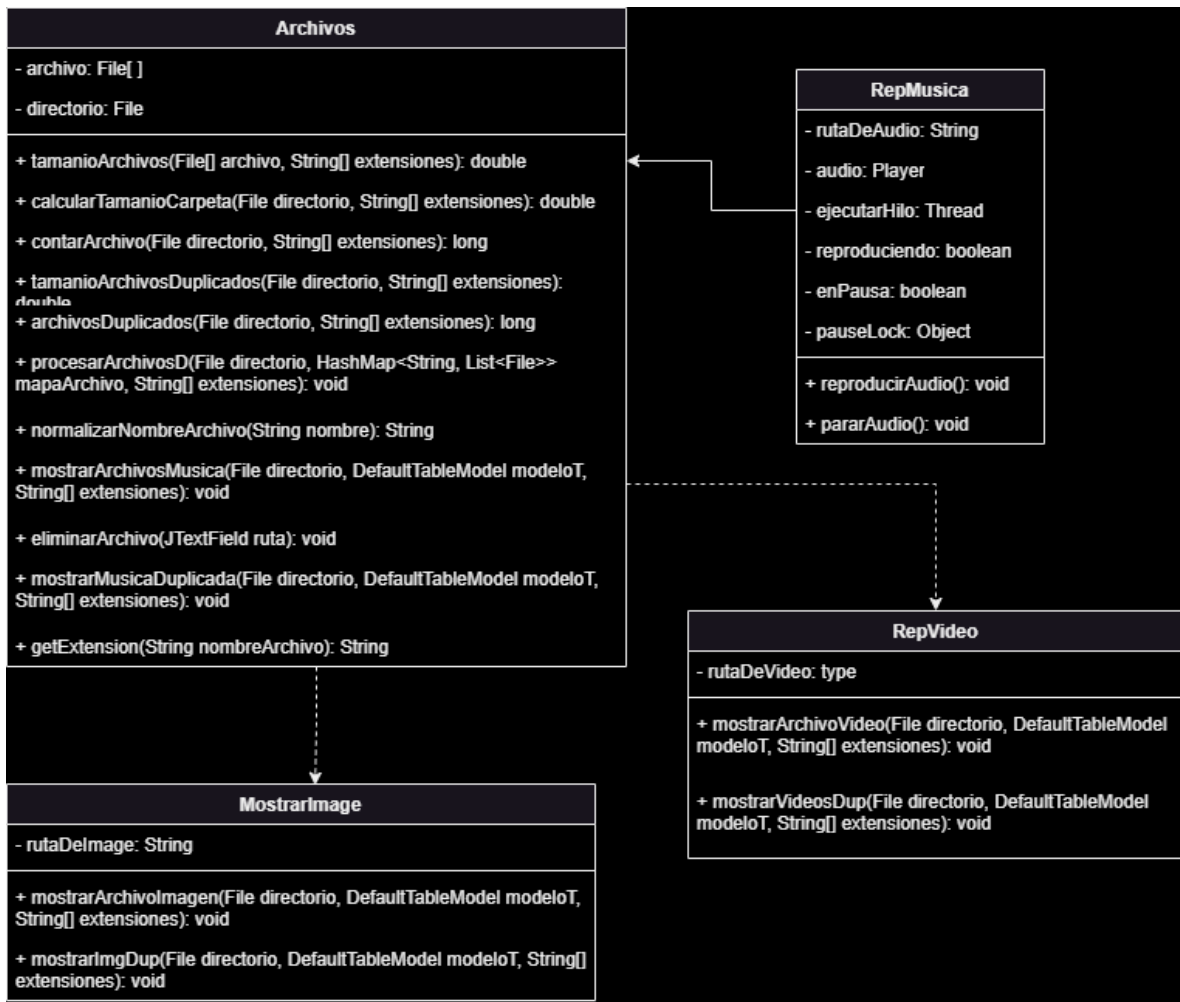
# MANUAL TÉCNICO

Christopher Alexey Robles Morales 3090-23-13774





**Se muestra anteriormente el diagrama de flujo del que se compone la aplicación Gestor Pro.**



**Aquí está el diagrama de clases creadas por mi persona, son 4 clases, su función principal es obtener la información de los archivos y la reproducción correcta de estos, hay dos clases que heredan de la clase archivos, para mantener una eficiencia en el código.**

```

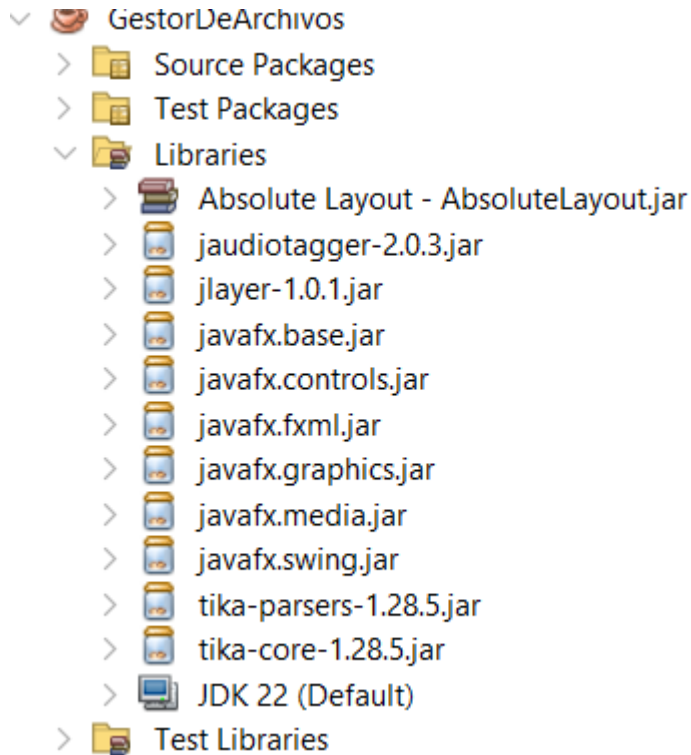
public void mostrarArchivosMusica(File directorio, DefaultTableModel modeloT, String[] extensiones){
    File[] archiv = directorio.listFiles();

    if(archiv != null){
        for(File archi : archiv){
            if(archi.isFile()){
                for(String extension : extensiones){
                    if(archi.getName().endsWith(extension)){
                        try{
                            AudioFile archivoAudio = AudioFileIO.read(archi);
                            Tag metadatos = archivoAudio.getTag();
                            AudioHeader cabecera = archivoAudio.getAudioHeader();
                            String nombreAudio = archi.getName();
                            String ext = extension;
                            String artista = metadatos != null ? metadatos.getFirst(FieldKey.ARTIST) : "Desconocido";
                            String album = metadatos != null ? metadatos.getFirst(FieldKey.ALBUM) : "Desconocido";
                            String genero = metadatos != null ? metadatos.getFirst(FieldKey.GENERO) : "Desconocido";
                            int duracion = cabecera.getTrackLength();
                            String anio = metadatos != null ? metadatos.getFirst(FieldKey.YEAR) : "N/A";
                            String ruta = archi.getAbsolutePath();
                            double tamañoByte = archi.length();
                            double tamañoMB = tamañoByte / (1024 * 1024);
                            String espacioMB = String.format("%.2f MB", tamañoMB);

                            Object[] contenedorArchivo = new Object[]{
                                nombreAudio, ext, artista, album, genero, duracion, anio, ruta, espacioMB
                            };
                            //System.out.println("Agregando: " + archi.getName());
                            modeloT.addRow(contenedorArchivo);
                        }catch(HeadlessException | IOException | CannotReadException | InvalidAudioFormatException | ReadOnlyFileException | KeyNotFoundException | TagException e){
                            JOptionPane.showMessageDialog(null, e.toString());
                        }
                        break; // no se siguen verificando otras extensiones
                    }
                }
            }
        }
    }
}

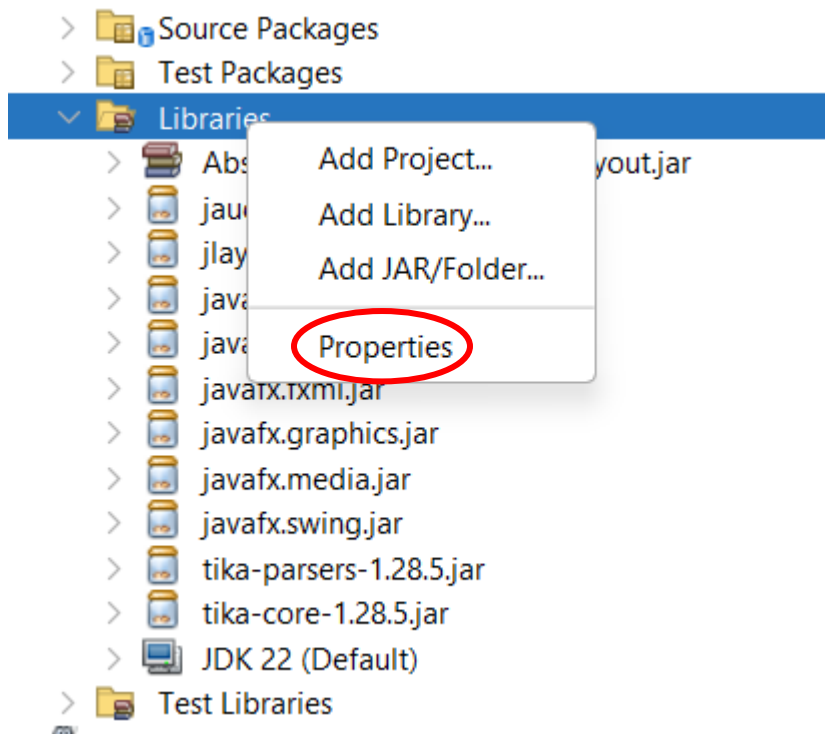
```

**Aquí se muestra un fragmento de código en cual es un método recursivo para poder llevar la tabla de música, para los demás tipos de archivos la lógica es la misma en el único punto en el que varían es en la obtención de sus metadatos para lo cual se utilizan sus librerías específicas.**

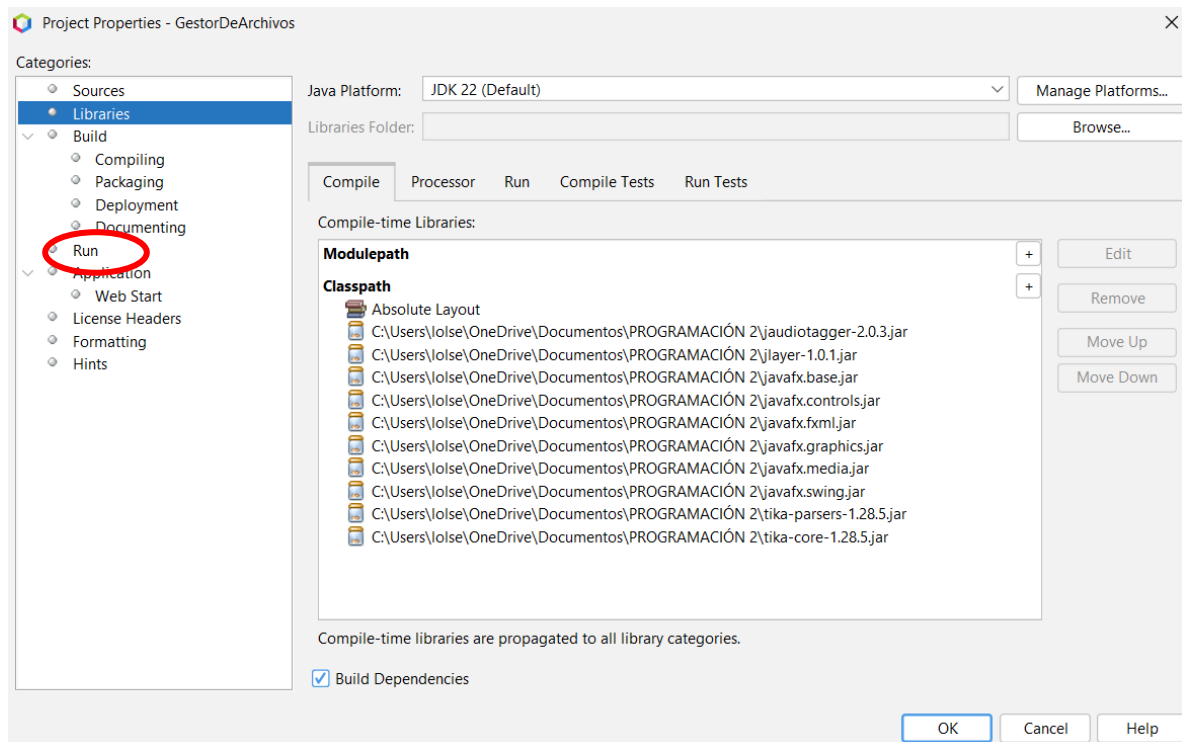


**Estas son las librerías necesarias para el funcionamiento correcto del proyecto, pueden descargarse a través de internet sin mayor problema buscando la versión y el nombre de la librería, se debe de tener en cuenta que deben de ser archivos con la terminación “.jar” de lo contrario no funcionarán.**

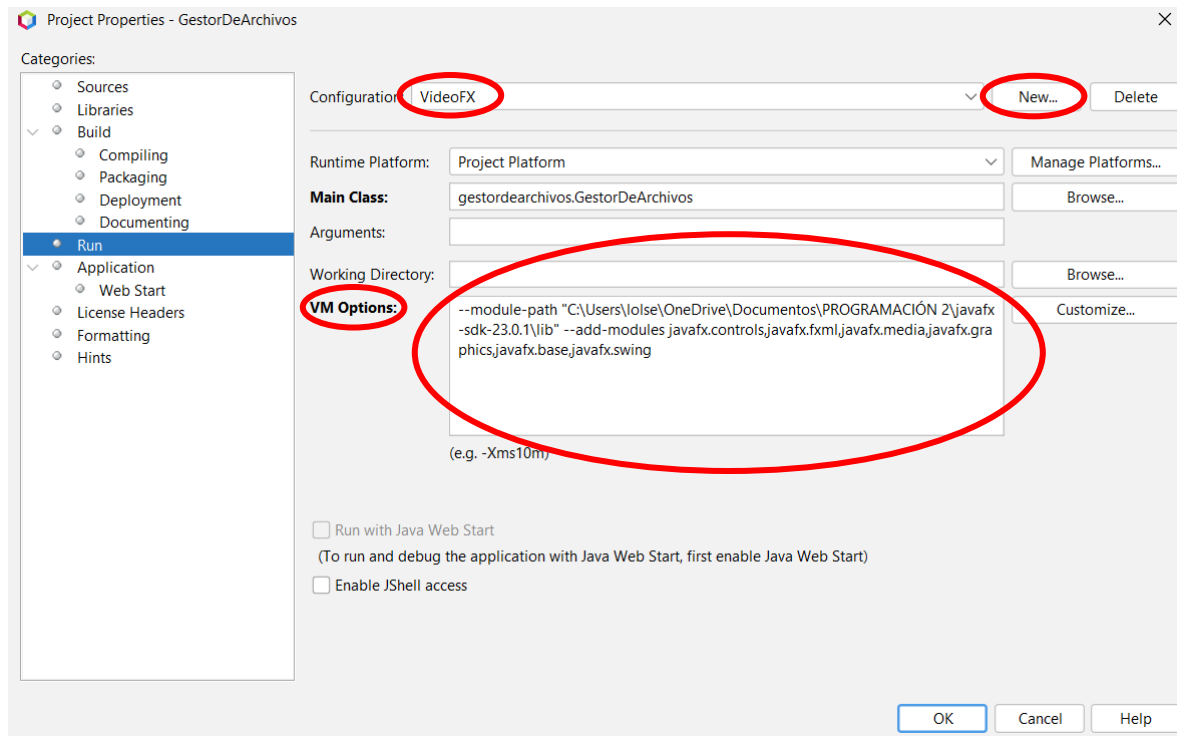
**Además de la importación de las librerías para que el proyecto pueda funcionar de manera correcta se debe de hacer una importación de la carpeta hacia el classpath, a continuación se muestra como:**



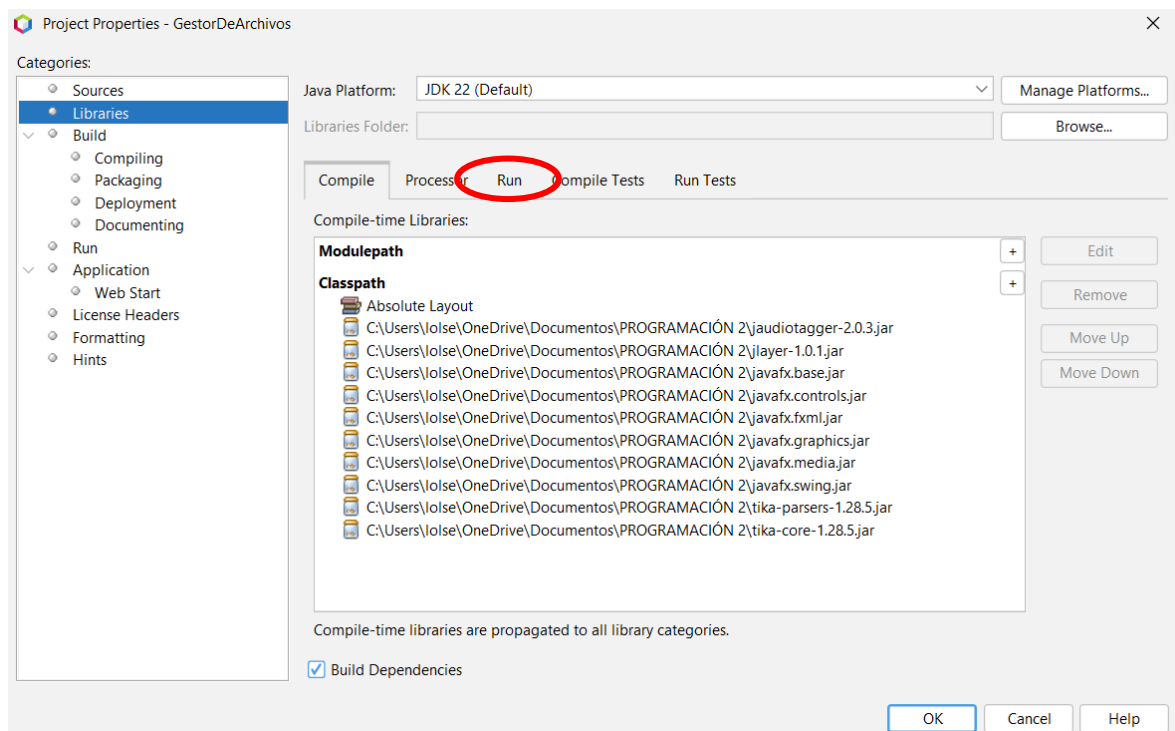
**Paso uno, dar click izquierdo en Libraries y presionar Properties.**



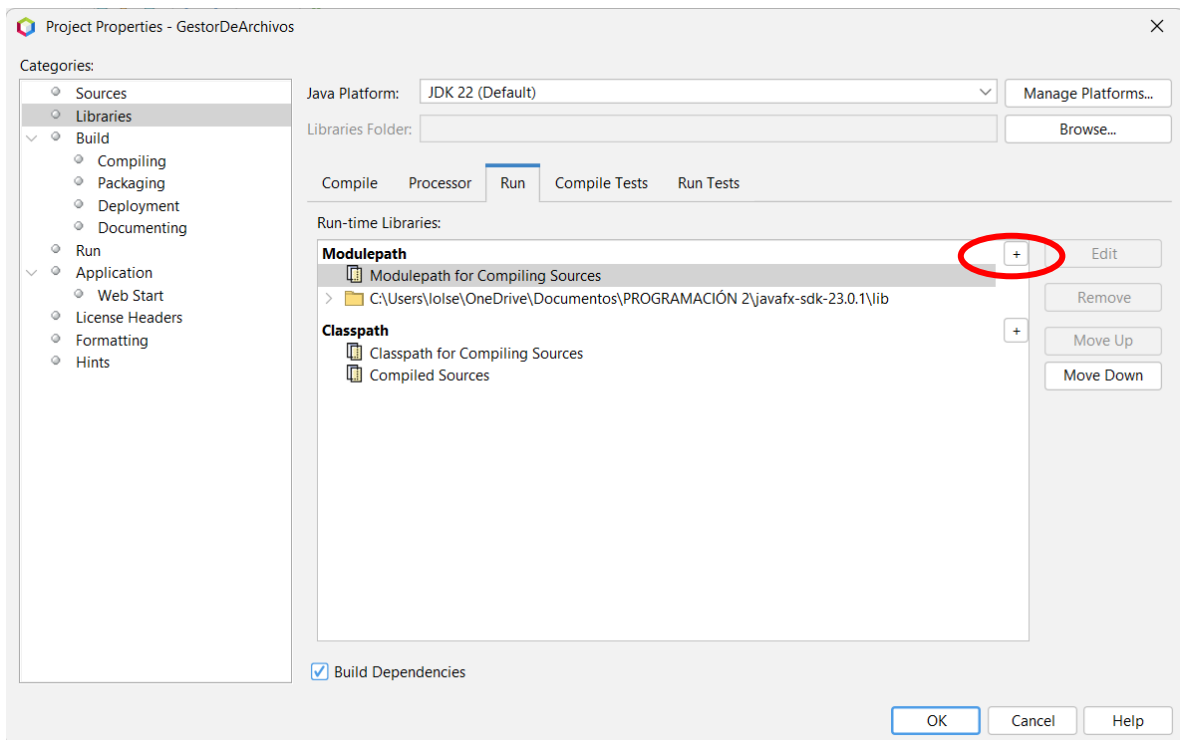
**Dentro de esta ventana seleccionamos la opción "Run".**



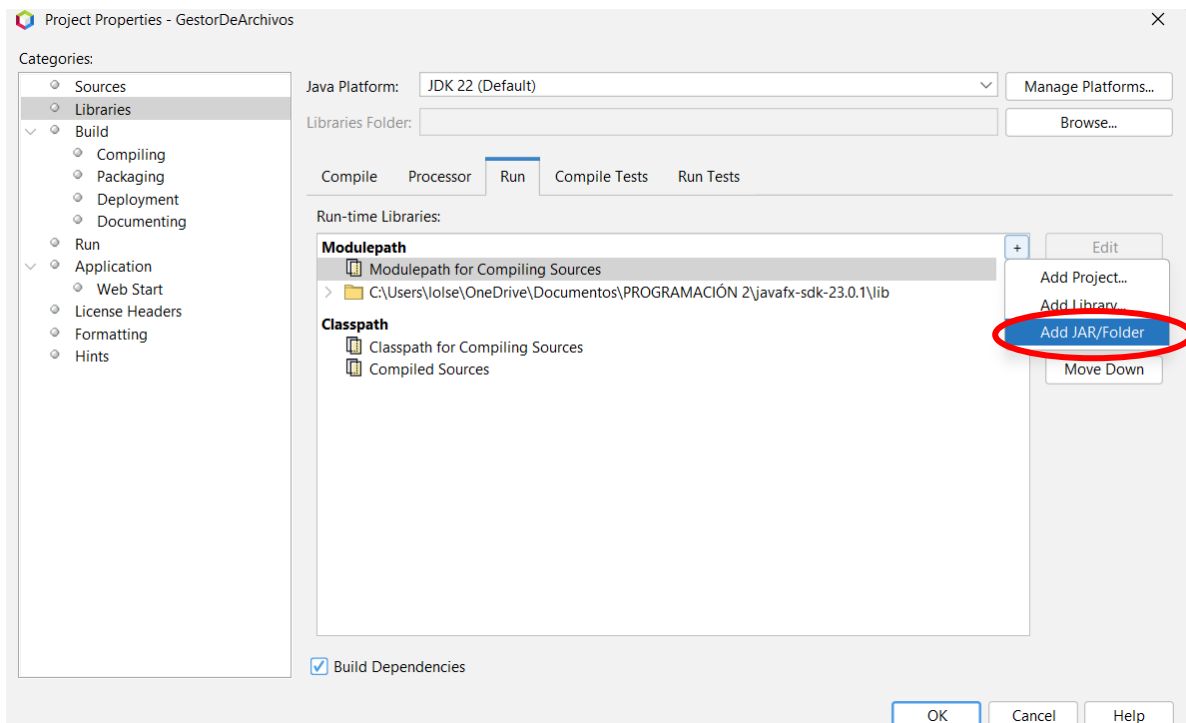
**Creemos una nueva configuración y la nombre “VideoFX” y luego en donde dice “VM Options” colocamos -- module-path “la ruta/de/nuestra/carpeta/lib/de/FX” y en --add-modules colocamos las librerías de javaFX para que funcione de manera correcta, también se debe de hacer aquí:**



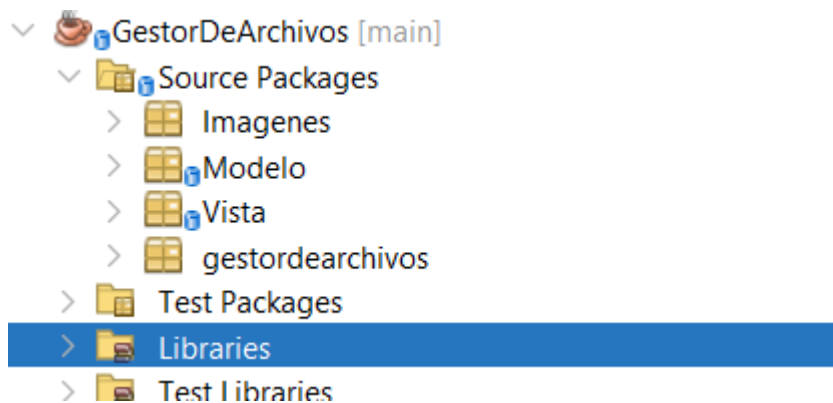
**Estando aquí solamente colocamos dentro de Modulepath le damos al más y añadimos la ruta de la carpeta donde están estas librerías:**



**Y damos en add JAR/Folder y seleccionamos nuestra carpeta**



**Una vez hecho esto se podría utilizar las librerías de javaFX de manera correcta y sin errores.**



**El proyecto se dividió en 3 paquetes para mantener de forma estructurada el programa, en Images, se encuentran todas las imágenes que utilizan los iconos, botones y labels.**

**En el apartado de modelo van todas las clases que tiene la funcionalidad, para reproducir o llenar las tablas, en el paquete de vista están toda la parte visual del archivo, aquí mismo también se colocaron dentro de los JFrame las funcionalidades de los botones y demás opciones.**

**El paquete de gestordearchivos es el archivo main del proyecto.**

```
JFileChooser jFileChooser1 = new JFileChooser();
jFileChooser1.setMultiSelectionEnabled(true); // Permitir seleccionar múltiples archivos
jFileChooser1.setSelectionMode(JFileChooser.FILES_AND_DIRECTORIES); // filtro que nos permite elegir que es lo que se selecciona
int respuesta = jFileChooser1.showOpenDialog(null);

if (respuesta == JFileChooser.APPROVE_OPTION) {

    File nombre = jFileChooser1.getSelectedFile();
    archivoSeleccionado = jFileChooser1.getSelectedFiles();
}
```

**Este fragmento de código es la lógica para el funcionamiento correcto del JFileChooser, se declara un objeto de tipo FileChooser, luego se van integrando las características que tendrá, por ejemplo se puede agregar un filtro para ver si agrega un archivo o una carpeta o ambas.**