REPRODUCTOR DE VIDEO

Código del proyecto

Adrián Martínez Medina

18 de marzo de 2022

Contenido

MediaPlayer.java

```java
/**
 * Clase java usada para ejecutar la app
 *
 * Crea la ventana, cargando todos los componentes
 * a partir de la vista fxml que hemos creado y
 * preparado en el scene builder
 *
 * @author amartinez
 */
package mediaplayer;

// Listado de imports
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;

public class MediaPlayer extends Application {

    @Override
    public void start(Stage stage) throws Exception {

        // Creamos el parent root cargando los nodos desde el fxml
        Parent root = FXMLLoader.load(getClass().getResource("MediaPlayer.fxml"));

        /**
         * Creamos el objeto que nos dará el contenido de la ventana a partir de
         * los nodos pasados desde root
         */
        Scene scene = new Scene(root);

        // A la escena le añadimos los valores descriptivos que queremos
        stage.setTitle("Media Player hecho por Adrian");
        stage.setFullScreenExitHint("Presione ESC or Doble Click para salir de la pantalla completa");

        // Añadimos a la ventana nuestro icono personalizado
        stage.getIcons().add(new Image("images/icon.png"));

        /**
         * Añadimos a la escena la capacidad de cambiar entre modo ventana y
         * modo pantalla completa al hacer doble click sobre el video
         */
        scene.setOnMouseClicked((MouseEvent clicked) -> {
            if (clicked.getClickCount() == 2) {
                if (!stage.isFullScreen()) {
                    stage.setFullScreen(true);
                } else {
                    stage.setFullScreen(false);
                }
            }
        });

        /**
         * Añadimos el contenido guardado en la escena a la ventana y la hacemos
         * visible
         */
        stage.setScene(scene);
        stage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}
```

MediaPlayerController.java

```java
/**
 * Clase java usada para las funcionalidades de la app
 *
 * Hace a nuestra app capaz de pausar, activar,
 * parar, acelerar, ralentizar, avanzar y retroceder
 *
 * Además, también habilitamos la barra de progreso
 * del video y la de volumen. Por último también se crean
 * los keyevents para la funcionalidades
 *
 * @author amartinez
 */
package mediaplayer;

// Listado de imports
import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.util.Arrays;
import java.util.List;
import java.util.ResourceBundle;
import javafx.animation.FadeTransition;
import javafx.beans.Observable;
import javafx.beans.binding.Bindings;
import javafx.beans.property.DoubleProperty;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.MenuBar;
import javafx.scene.control.Slider;
import javafx.scene.image.Image;
import javafx.scene.input.KeyCode;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.scene.media.MediaView;
import static javafx.scene.media.MediaPlayer.Status.PLAYING;
import javafx.stage.FileChooser;
import javafx.stage.Stage;
import javafx.util.Duration;
import javax.swing.JOptionPane;

public class MediaPlayerController implements Initializable {

    // Variable que almacena la ruta al video abierto
    @FXML
    private String path;

    /**
     * Variable que almacena todas las extensiones de archivo válidas para
     * nuestra aplicación en un array
     */
    @FXML
    private List<String> extensions;

    // Variable objeto, crea el mediaPlayer
    @FXML
    private MediaPlayer mediaPlayer;

    // Variable objeto, crea el mediaView
    @FXML
    private MediaView mediaView;

    // Variable objeto, crea el slider de volumen
    @FXML
    private Slider volumeSlider;

    // Variable objeto, crea el slider de progreso de video
    @FXML
    private Slider progressBar;

    // Botones de la aplicación
```

```java
    @FXML
    private Button openFile, playVideo, stopVideo, slowVideo,
            backVideo5, normalVideo, forwardVideo5, fastVideo, setVisibilityButtons;

    // Menu bar de la apliación
    @FXML
    private MenuBar menuBar;

    // Booleana que usaremos para mostrar y ocultar los botones
    @FXML
    private boolean visible = true;

    /**
     * Método que usamos para añadir al boton de abrir archivo la capacidad de
     * hacerlo.
     *
     * @param event
     */
    @FXML
    private void OpenFileMethod(ActionEvent event) {

        try {
            /**
             * llenamos el array de strings con las extensiones permitidas en
             * nuestra app
             */
            extensions = Arrays.asList("*.mp4", "*.3gp", "*.mkv", "*.MP4", "*.MKV", "*.3GP", "*.flv", "*.wmv");

            /**
             * Creamos tanto el objeto filechooser para elegir archivo como el
             * objeto extension filter para añadir el filtro de extensiones que
             * hemos concretado en el array anterior
             */
            FileChooser fileChooser = new FileChooser();
            FileChooser.ExtensionFilter filter = new FileChooser.ExtensionFilter("selecciona video", extensions);

            // Añadimos el filtro de extensiones al filechooser
            fileChooser.getExtensionFilters().add(filter);
            fileChooser.setTitle("Selecciona un video");

            /**
             * Abrimos la ventana para elegir archivo y este se guarda como
             * variable File
             *
             * Posteriormente guardamos en un string la ruta del archivo elegido
             * para ser usada en el objeto Media
             */
            File file = fileChooser.showOpenDialog(null);
            path = file.toURI().toString();

            /**
             * Siempre y cuando se haya elegido un archivo, se realizará el
             * código contenido en el if
             */
            if (path != null) {
                /**
                 * Creamos el objeto Media gracias al path, luego lo usamos para
                 * crear el MediaPlayer y finalmente creamos la vista del video
                 */
                Media media = new Media(path);
                mediaPlayer = new MediaPlayer(media);
                mediaView.setMediaPlayer(mediaPlayer);

                /**
                 * Las lineas que tenemos a continuación sirven para definir las
                 * dimensiones del video cargado ajustandolo al mediaView
                 */
                DoubleProperty widthProp = mediaView.fitWidthProperty();
                DoubleProperty heightProp = mediaView.fitHeightProperty();
                widthProp.bind(Bindings.selectDouble(mediaView.sceneProperty(), "width"));
                heightProp.bind(Bindings.selectDouble(mediaView.sceneProperty(), "height"));

                // Damos un valor inicial al slider del volumen del video
                volumeSlider.setValue(mediaPlayer.getVolume() * 100);

                /**
                 * Creamos el listener que pertimirá ajustar el volumen del
                 * video según donde coloquemos el punto del slider del volumen
                 */
```

```
158                 volumeSlider.valueProperty().addListener((Observable observable) -> {
159                     mediaPlayer.setVolume(volumeSlider.getValue() / 100);
160                 });
161 
162                 // Damos un valor inicial a la barra de progreso del video
163                 mediaPlayer.currentTimeProperty().addListener((ObservableValue<? extends javafx.util.Duration> observable,
    javafx.util.Duration oldValue, javafx.util.Duration newValue) -> {
164                     progressBar.setValue(newValue.toSeconds());
165                 });
166 
167                 /**
168                  * Creamos los listener que pertimirá ajustar el volumen del
169                  * video según donde coloquemos el punto del slider del volumen
170                  */
171                 progressBar.setOnMousePressed((MouseEvent event1) -> {
172                     mediaPlayer.seek(javafx.util.Duration.seconds(progressBar.getValue()));
173                 });
174                 progressBar.setOnMouseDragged((MouseEvent event1) -> {
175                     mediaPlayer.seek(javafx.util.Duration.seconds(progressBar.getValue()));
176                 });
177                 mediaPlayer.setOnReady(() -> {
178                     javafx.util.Duration total = media.getDuration();
179                     progressBar.setMax(total.toSeconds());
180                 });
181 
182                 // Por último, iniciamos la reproduccion del video al abrirlo
183                 mediaPlayer.play();
184             }
185         } catch (Exception e) {
186         }
187 
188     }
189 
190     @Override
191     public void initialize(URL url, ResourceBundle rb) {
192         // TODO
193     }
194 
195     /**
196      * Método que usamos para añadir al boton de play & pause video la capacidad
197      * de hacerlo.
198      *
199      * @param event
200      */
201     @FXML
202     private void playVideo(ActionEvent event) {
203 
204         /**
205          * Siempre y cuando se haya elegido previamente un video, se comprueba
206          * si el video está activo o pausado
207          *
208          * Si está activo, se pausa y si está pausado se reactiva
209          */
210         try {
211             if (mediaPlayer.getStatus() == PLAYING) {
212                 mediaPlayer.pause();
213             } else {
214                 mediaPlayer.play();
215             }
216         } catch (Exception e) {
217             JOptionPane.showMessageDialog(null, "Elige un video primero");
218         }
219 
220     }
221 
222     /**
223      * Método que usamos para añadir al boton de stop video la capacidad de
224      * hacerlo.
225      *
226      * @param event
227      */
228     @FXML
229     private void stopVideo(ActionEvent event) {
230 
231         /**
232          * Siempre y cuando se haya elegido previamente un video, se detiene el
233          * video reseteandolo
234          */
235         try {
```

```
            mediaPlayer.stop();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Elige un video primero");
        }

    }

    /**
     * Método que usamos para añadir al boton de avanzar video la capacidad de
     * hacerlo.
     *
     * @param event
     */
    @FXML
    private void skip5(ActionEvent event) {

        /**
         * Siempre y cuando se haya elegido previamente un video, se adelanta el
         * video 5 segundos
         */
        try {
            mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(5)));
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Elige un video primero");
        }

    }

    /**
     * Método que usamos para añadir al boton de acelerar video la capacidad de
     * hacerlo.
     *
     * @param event
     */
    @FXML
    private void furtherSpeedUpVideo(ActionEvent event) {

        /**
         * Siempre y cuando se haya elegido previamente un video, se acelera el
         * video
         */
        try {
            mediaPlayer.setRate(2);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Elige un video primero");
        }

    }

    /**
     * Método que usamos para añadir al boton de atrasar video la capacidad de
     * hacerlo.
     *
     * @param event
     */
    @FXML
    private void back5(ActionEvent event) {

        /**
         * Siempre y cuando se haya elegido previamente un video, se atrasa el
         * video 5 segundos
         */
        try {
            mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(-5)));
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Elige un video primero");
        }

    }

    /**
     * Método que usamos para añadir al boton de ralentizar video la capacidad
     * de hacerlo.
     *
     * @param event
     */
    @FXML
    private void furtherSlowDownVideo(ActionEvent event) {
```

```
314
315         /**
316          * Siempre y cuando se haya elegido previamente un video, se ralentiza
317          * el video
318          */
319         try {
320             mediaPlayer.setRate(0.5);
321         } catch (Exception e) {
322             JOptionPane.showMessageDialog(null, "Elige un video primero");
323         }
324
325     }
326
327     /**
328      * Método que usamos para añadir al boton de normalizar velocidad video la
329      * capacidad de hacerlo.
330      *
331      * @param event
332      */
333     @FXML
334     private void normalSpeedVideo(ActionEvent event) {
335
336         /**
337          * Siempre y cuando se haya elegido previamente un video, se resetea la
338          * velocidad de reporducción del video
339          */
340         try {
341             mediaPlayer.setRate(1);
342         } catch (Exception e) {
343             JOptionPane.showMessageDialog(null, "Elige un video primero");
344         }
345
346     }
347
348     /**
349      * Método que usamos para abrir la ventana Acerca De al hacer click en el
350      * menú de la app
351      *
352      * @param event
353      */
354     @FXML
355     private void openAcercaDe(ActionEvent event) {
356         try {
357             // Creamos el parent root cargando los nodos desde el fxml
358             Parent root = FXMLLoader.load(getClass().getResource("AcercaDe.fxml"));
359
360             /**
361              * Creamos el objeto que nos dará el contenido de la ventana a
362              * partir de los nodos pasados desde root
363              */
364             Scene scene = new Scene(root);
365
366             // Creamos la ventana acerca de
367             Stage acercade = new Stage();
368
369             // Añadimos a la ventana nuestro icono personalizado
370             acercade.getIcons().add(new Image("images/icon.png"));
371
372             /**
373              * Añadimos el contenido guardado en la escena a la ventana y la
374              * hacemos visible
375              */
376             acercade.setScene(scene);
377             acercade.show();
378         } catch (IOException e) {
379         }
380     }
381
382     /**
383      * Método que usamos para mostrar y ocultar todos los controles con el fin
384      * de que el video sea perfectamente visible sin molestias a gusto del
385      * usuario
386      *
387      * @param event
388      */
389     @FXML
390     private void hideShowControls(ActionEvent event) {
391
```

```
        try {
            /**
             * Objetos transiciones que nos permiten hacer visibles
             * e invisibles posteriormente a los controles de nuestra
             * aplicación cambiando su visibilidad
             */
            FadeTransition fadeTransitionVisibility = new FadeTransition(Duration.millis(200), setVisibilityButtons);
            FadeTransition fadeTransitionOpenFile = new FadeTransition(Duration.millis(200), openFile);
            FadeTransition fadeTransitionPlayVideo = new FadeTransition(Duration.millis(200), playVideo);
            FadeTransition fadeTransitionStopVideo = new FadeTransition(Duration.millis(200), stopVideo);
            FadeTransition fadeTransitionSlowVideo = new FadeTransition(Duration.millis(200), slowVideo);
            FadeTransition fadeTransitionBackVideo5 = new FadeTransition(Duration.millis(200), backVideo5);
            FadeTransition fadeTransitionNormalVideo = new FadeTransition(Duration.millis(200), normalVideo);
            FadeTransition fadeTransitionForwardVideo5 = new FadeTransition(Duration.millis(200), forwardVideo5);
            FadeTransition fadeTransitionFastVideo = new FadeTransition(Duration.millis(200), fastVideo);
            FadeTransition fadeTransitionVolumeSlider = new FadeTransition(Duration.millis(200), volumeSlider);
            FadeTransition fadeTransitionProgressBar = new FadeTransition(Duration.millis(200), progressBar);
            FadeTransition fadeTransitionMenuBar = new FadeTransition(Duration.millis(200), menuBar);

            if (visible) {
                // Ponemos la visibilidad de los controles a 0
                fadeTransitionVisibility.setToValue(0.2);
                fadeTransitionOpenFile.setToValue(0);
                fadeTransitionPlayVideo.setToValue(0);
                fadeTransitionStopVideo.setToValue(0);
                fadeTransitionSlowVideo.setToValue(0);
                fadeTransitionBackVideo5.setToValue(0);
                fadeTransitionNormalVideo.setToValue(0);
                fadeTransitionForwardVideo5.setToValue(0);
                fadeTransitionFastVideo.setToValue(0);
                fadeTransitionVolumeSlider.setToValue(0);
                fadeTransitionProgressBar.setToValue(0);
                fadeTransitionMenuBar.setToValue(0);

                // Ejecutamos la transicion
                fadeTransitionVisibility.play();
                fadeTransitionOpenFile.play();
                fadeTransitionPlayVideo.play();
                fadeTransitionStopVideo.play();
                fadeTransitionSlowVideo.play();
                fadeTransitionBackVideo5.play();
                fadeTransitionNormalVideo.play();
                fadeTransitionForwardVideo5.play();
                fadeTransitionFastVideo.play();
                fadeTransitionVolumeSlider.play();
                fadeTransitionProgressBar.play();
                fadeTransitionMenuBar.play();

                /**
                 * al completarse la última transición cambiamos el
                 * valor de la booleana visible para que al volver
                 * a hacer clic se lleve a cabo el efecto contrario
                 * a hacer clic se lleve a cabo el efecto contrario
                 */
                fadeTransitionMenuBar.setOnFinished(e -> {
                    visible = false;
                });
            }
            if (!visible) {
                // Ponemos la visibilidad de los controles a 1
                fadeTransitionVisibility.setToValue(1);
                fadeTransitionOpenFile.setToValue(1);
                fadeTransitionPlayVideo.setToValue(1);
                fadeTransitionStopVideo.setToValue(1);
                fadeTransitionSlowVideo.setToValue(1);
                fadeTransitionBackVideo5.setToValue(1);
                fadeTransitionNormalVideo.setToValue(1);
                fadeTransitionForwardVideo5.setToValue(1);
                fadeTransitionFastVideo.setToValue(1);
                fadeTransitionVolumeSlider.setToValue(1);
                fadeTransitionProgressBar.setToValue(1);
                fadeTransitionMenuBar.setToValue(1);

                // Ejecutamos la transicion
                fadeTransitionVisibility.play();
                fadeTransitionOpenFile.play();
                fadeTransitionPlayVideo.play();
                fadeTransitionStopVideo.play();
                fadeTransitionSlowVideo.play();
                fadeTransitionBackVideo5.play();
                fadeTransitionNormalVideo.play();
                fadeTransitionForwardVideo5.play();
                fadeTransitionFastVideo.play();
                fadeTransitionVolumeSlider.play();
                fadeTransitionProgressBar.play();
                fadeTransitionMenuBar.play();

```

```
478                 /**
479                  * al completarse la última transición cambiamos el
480                  * valor de la booleana visible para que al volver
481                  * a hacer clic se lleve a cabo el efecto contrario
482                  */
483                 fadeTransitionMenuBar.setOnFinished(e -> {
484                     visible = true;
485                 });
486             }
487         } catch (Exception e) {
488         }
489 
490     }
491 
492     /**
493      * Método que contiene todos los keylisteners de la app, haciendo posible la
494      * ejecución de las funciones descritas anteriormente segun la combinación
495      * de teclas pulsada.
496      *
497      * @param ke
498      */
499     @FXML
500     private void handleKeyPressed(KeyEvent ke) {
501 
502         if (ke.getCode().equals(KeyCode.P) && ke.isControlDown()) {
503             try {
504                 if (mediaPlayer.getStatus() == PLAYING) {
505                     mediaPlayer.pause();
506                 } else {
507                     mediaPlayer.play();
508                 }
509             } catch (Exception e) {
510                 JOptionPane.showMessageDialog(null, "Elige un video primero");
511             }
512         }
513 
514         if (ke.getCode().equals(KeyCode.S) && ke.isControlDown()) {
515             try {
516                 mediaPlayer.stop();
517             } catch (Exception e) {
518                 JOptionPane.showMessageDialog(null, "Elige un video primero");
519             }
520         }
521 
522         if (ke.getCode().equals(KeyCode.RIGHT) && ke.isControlDown()) {
523             try {
524                 mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(5)));
525             } catch (Exception e) {
526                 JOptionPane.showMessageDialog(null, "Elige un video primero");
527             }
528         }
529 
530         if (ke.getCode().equals(KeyCode.LEFT) && ke.isControlDown()) {
531             try {
532                 mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(-5)));
533             } catch (Exception e) {
534                 JOptionPane.showMessageDialog(null, "Elige un video primero");
535             }
536         }
537 
538         if (ke.getCode().equals(KeyCode.UP) && ke.isControlDown()) {
539             try {
540                 mediaPlayer.setRate(2);
541             } catch (Exception e) {
542                 JOptionPane.showMessageDialog(null, "Elige un video primero");
543             }
544         }
545 
546         if (ke.getCode().equals(KeyCode.DOWN) && ke.isControlDown()) {
547             try {
548                 mediaPlayer.setRate(0.5);
549             } catch (Exception e) {
550                 JOptionPane.showMessageDialog(null, "Elige un video primero");
551             }
552         }
553 
554         if (ke.getCode().equals(KeyCode.N) && ke.isControlDown()) {
555             try {
556                 mediaPlayer.setRate(1);
557             } catch (Exception e) {
558                 JOptionPane.showMessageDialog(null, "Elige un video primero");
559             }
560         }
561 
```

```
        if (ke.getCode().equals(KeyCode.A) && ke.isControlDown()) {
            try {
                Parent root = FXMLLoader.load(getClass().getResource("AcercaDe.fxml"));

                Scene scene = new Scene(root);
                Stage acercade = new Stage();

                acercade.setTitle("Media Player hecho por Adrian");
                acercade.getIcons().add(new Image("images/icon.png"));

                acercade.setScene(scene);
                acercade.show();
            } catch (IOException e) {
            }
        }

        if (ke.getCode().equals(KeyCode.H) && ke.isControlDown()) {
            try {
                FadeTransition fadeTransitionVisibility = new FadeTransition(Duration.millis(200), setVisibilityButtons);
                FadeTransition fadeTransitionOpenFile = new FadeTransition(Duration.millis(200), openFile);
                FadeTransition fadeTransitionPlayVideo = new FadeTransition(Duration.millis(200), playVideo);
                FadeTransition fadeTransitionStopVideo = new FadeTransition(Duration.millis(200), stopVideo);
                FadeTransition fadeTransitionSlowVideo = new FadeTransition(Duration.millis(200), slowVideo);
                FadeTransition fadeTransitionBackVideo5 = new FadeTransition(Duration.millis(200), backVideo5);
                FadeTransition fadeTransitionNormalVideo = new FadeTransition(Duration.millis(200), normalVideo);
                FadeTransition fadeTransitionForwardVideo5 = new FadeTransition(Duration.millis(200), forwardVideo5);
                FadeTransition fadeTransitionFastVideo = new FadeTransition(Duration.millis(200), fastVideo);
                FadeTransition fadeTransitionVolumeSlider = new FadeTransition(Duration.millis(200), volumeSlider);
                FadeTransition fadeTransitionProgressBar = new FadeTransition(Duration.millis(200), progressBar);
                FadeTransition fadeTransitionMenuBar = new FadeTransition(Duration.millis(200), menuBar);

                if (visible) {
                    fadeTransitionVisibility.setToValue(0.2);
                    fadeTransitionOpenFile.setToValue(0);
                    fadeTransitionPlayVideo.setToValue(0);
                    fadeTransitionStopVideo.setToValue(0);
                    fadeTransitionSlowVideo.setToValue(0);
                    fadeTransitionBackVideo5.setToValue(0);
                    fadeTransitionNormalVideo.setToValue(0);
                    fadeTransitionForwardVideo5.setToValue(0);
                    fadeTransitionFastVideo.setToValue(0);
                    fadeTransitionVolumeSlider.setToValue(0);
                    fadeTransitionProgressBar.setToValue(0);
                    fadeTransitionMenuBar.setToValue(0);

                    fadeTransitionVisibility.play();
                    fadeTransitionOpenFile.play();
                    fadeTransitionPlayVideo.play();
                    fadeTransitionStopVideo.play();
                    fadeTransitionSlowVideo.play();
                    fadeTransitionBackVideo5.play();
                    fadeTransitionNormalVideo.play();
                    fadeTransitionForwardVideo5.play();
                    fadeTransitionFastVideo.play();
                    fadeTransitionVolumeSlider.play();
                    fadeTransitionProgressBar.play();
                    fadeTransitionMenuBar.play();

                    fadeTransitionMenuBar.setOnFinished(e -> {
                        visible = false;
                    });
                }
                if (!visible) {
                    fadeTransitionVisibility.setToValue(1);
                    fadeTransitionOpenFile.setToValue(1);
                    fadeTransitionPlayVideo.setToValue(1);
                    fadeTransitionStopVideo.setToValue(1);
                    fadeTransitionSlowVideo.setToValue(1);
                    fadeTransitionBackVideo5.setToValue(1);
                    fadeTransitionNormalVideo.setToValue(1);
                    fadeTransitionForwardVideo5.setToValue(1);
                    fadeTransitionFastVideo.setToValue(1);
                    fadeTransitionVolumeSlider.setToValue(1);
                    fadeTransitionProgressBar.setToValue(1);
                    fadeTransitionMenuBar.setToValue(1);

```

```
                    fadeTransitionVisibility.play();
                    fadeTransitionOpenFile.play();
                    fadeTransitionPlayVideo.play();
                    fadeTransitionStopVideo.play();
                    fadeTransitionSlowVideo.play();
                    fadeTransitionBackVideo5.play();
                    fadeTransitionNormalVideo.play();
                    fadeTransitionForwardVideo5.play();
                    fadeTransitionFastVideo.play();
                    fadeTransitionVolumeSlider.play();
                    fadeTransitionProgressBar.play();
                    fadeTransitionMenuBar.play();

                    fadeTransitionMenuBar.setOnFinished(e -> {
                        visible = true;
                    });
                }
            } catch (Exception e) {
            }
        }

    }

}
```

MediaPlayer.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.Slider?>
<?import javafx.scene.effect.InnerShadow?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.media.MediaView?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" onKeyPressed="#handleKeyPressed" prefHeight="400.0" prefWidth="600.0" stylesheets="@style.css" xmlns="http://javafx.com/javafx/17" xmlns:fx="http://javafx.com/fxml/1" fx:controller="mediaplayer.MediaPlayerController">
    <center>
        <StackPane fx:id="pane" prefHeight="300.0" prefWidth="609.0" BorderPane.alignment="CENTER">
            <children>
                <MediaView fx:id="mediaView" fitHeight="300.0" fitWidth="600.0">
                    <effect>
                        <InnerShadow />
                    </effect>
                </MediaView>
                <VBox alignment="BOTTOM_CENTER" prefHeight="39.0" prefWidth="609.0">
                    <children>
                        <Slider fx:id="progressBar" prefHeight="0.0" prefWidth="600.0" />
                        <HBox alignment="BOTTOM_CENTER" opacity="0.8" prefHeight="39.0" prefWidth="576.0">
                            <children>
                                <Button fx:id="openFile" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#OpenFileMethod" prefWidth="35.0">
                                    <HBox.margin>
                                        <Insets bottom="7.0" right="14.0" />
                                    </HBox.margin>
                                </Button>
                                <Button fx:id="playVideo" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#playVideo" prefWidth="35.0">
                                    <HBox.margin>
                                        <Insets bottom="7.0" right="7.0" />
                                    </HBox.margin>
                                </Button>
                                <Button fx:id="stopVideo" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#stopVideo" prefWidth="35.0">
                                    <HBox.margin>
                                        <Insets bottom="7.0" right="14.0" />
                                    </HBox.margin>
                                </Button>
                                <Button fx:id="slowVideo" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#furtherSlowDownVideo" prefWidth="35.0">
                                    <HBox.margin>
                                        <Insets bottom="7.0" right="7.0" />
                                    </HBox.margin>
                                </Button>
                                <Button fx:id="backVideo5" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#back5" prefWidth="35.0">
```

```
                                    <HBox.margin>
                                        <Insets bottom="7.0" right="7.0" />
                                    </HBox.margin>
                                </Button>
                            <Button fx:id="normalVideo" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#normalSpeedVideo"
prefWidth="35.0">
                               <HBox.margin>
                                  <Insets bottom="7.0" right="7.0" />
                               </HBox.margin>
                            </Button>
                                    <Button fx:id="forwardVideo5" alignment="BOTTOM_LEFT" mnemonicParsing="false" onAction="#skip5"
prefWidth="35.0">
                                        <HBox.margin>
                                            <Insets bottom="7.0" right="7.0" />
                                        </HBox.margin>
                                    </Button>
                                    <Button fx:id="fastVideo" alignment="BOTTOM_LEFT" mnemonicParsing="false"
onAction="#furtherSpeedUpVideo" prefWidth="35.0">
                                        <HBox.margin>
                                            <Insets bottom="7.0" right="21.0" />
                                        </HBox.margin>
                                    </Button>
                                    <Slider fx:id="volumeSlider" stylesheets="@style.css">
                                        <HBox.margin>
                                            <Insets bottom="10.0" />
                                        </HBox.margin>
                                    </Slider>
                            <Button fx:id="setVisibilityButtons" alignment="BOTTOM_LEFT" mnemonicParsing="false"
onAction="#hideShowControls" prefWidth="35.0">
                               <HBox.margin>
                                  <Insets bottom="7.0" left="21.0" />
                               </HBox.margin>
                            </Button>
                                </children>
                            </HBox>
                        </children>
                    </VBox>
                    <MenuBar fx:id="menuBar" prefHeight="0.0" prefWidth="600.0" StackPane.alignment="TOP_CENTER">
                        <menus>
                            <Menu mnemonicParsing="false" text="Ayuda">
                                <items>
                                    <MenuItem fx:id="menuAcercaDe" mnemonicParsing="false" onAction="#openAcercaDe" text="Acerca De" />
                                </items>
                            </Menu>
                        </menus>
                    </MenuBar>
                </children>
            </StackPane>
        </center>
</BorderPane>
```

AcercaDe.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Hyperlink?>
<?import javafx.scene.control.Separator?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.text.Font?>
<?import javafx.scene.text.Text?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" onKeyPressed="#handleKeyPressed" prefHeight="400.0" prefWidth="600.0" stylesheets="@style.css" xmlns="http://javafx.com/javafx/17" xmlns:fx="http://javafx.com/fxml/1" fx:controller="mediaplayer.MediaPlayerController">
   <top>
      <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Acerca De" underline="true" BorderPane.alignment="CENTER">
         <font>
            <Font name="Arial Black" size="36.0" />
         </font>
         <BorderPane.margin>
            <Insets top="20.0" />
         </BorderPane.margin>
      </Text>
   </top>
   <bottom>
      <Text strokeType="OUTSIDE" strokeWidth="0.0" text="(c) Copyright amartinez Junio, 2022. Todos los derechos reservados" wrappingWidth="568.724609375" BorderPane.alignment="CENTER">
         <BorderPane.margin>
            <Insets bottom="10.0" />
         </BorderPane.margin>
         <font>
            <Font name="Arial Black" size="12.0" />
         </font>
      </Text>
   </bottom>
   <center>
      <GridPane BorderPane.alignment="CENTER">
        <columnConstraints>
          <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
        </columnConstraints>
        <rowConstraints>
          <RowConstraints maxHeight="99.0" minHeight="0.0" prefHeight="0.0" vgrow="SOMETIMES" />
          <RowConstraints maxHeight="274.0" minHeight="10.0" prefHeight="274.0" vgrow="SOMETIMES" />
          <RowConstraints maxHeight="61.0" minHeight="0.0" prefHeight="0.0" vgrow="SOMETIMES" />
        </rowConstraints>
         <children>
            <GridPane GridPane.rowIndex="1">
              <columnConstraints>
                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
              </columnConstraints>
              <rowConstraints>
                <RowConstraints maxHeight="62.0" minHeight="10.0" prefHeight="62.0" vgrow="SOMETIMES" />
                <RowConstraints maxHeight="68.0" minHeight="0.0" prefHeight="23.0" vgrow="SOMETIMES" />
                <RowConstraints maxHeight="73.0" minHeight="0.0" prefHeight="23.0" vgrow="SOMETIMES" />
                  <RowConstraints maxHeight="148.0" minHeight="10.0" prefHeight="124.0" vgrow="SOMETIMES" />
                  <RowConstraints maxHeight="115.0" minHeight="10.0" prefHeight="42.0" vgrow="SOMETIMES" />
              </rowConstraints>
               <children>
                  <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Reproductor de video hecho por Adrián Martínez Medina">
                     <font>
                        <Font name="Arial Black" size="14.0" />
                     </font>
                     <GridPane.margin>
                        <Insets left="20.0" />
```

```
                     </GridPane.margin>
                  </Text>
                  <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Versión: 1.0" wrappingWidth="108.0" GridPane.rowIndex="1">
                     <font>
                        <Font name="Arial Black" size="14.0" />
                     </font>
                     <GridPane.margin>
                        <Insets left="20.0" />
                     </GridPane.margin>
                  </Text>
                  <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Build: 20220318-001" wrappingWidth="204.599609375"
   GridPane.rowIndex="2">
                     <font>
                        <Font name="Arial Black" size="14.0" />
                     </font>
                     <GridPane.margin>
                        <Insets left="20.0" />
                     </GridPane.margin>
                  </Text>
                  <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Toda la documentación junto al proyecto está disponible en un
   repositorio de Github con acceso gratuito, para acceder haz click en el siguiente link:" wrappingWidth="482.0" y="1.0"
   GridPane.rowIndex="3">
                     <font>
                        <Font name="Arial Black" size="14.0" />
                     </font>
                     <GridPane.margin>
                        <Insets left="20.0" />
                     </GridPane.margin>
                  </Text>
                  <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Esta aplicación ha sido desarrollada con JavaFX:" y="1.0"
   GridPane.rowIndex="4">
                     <font>
                        <Font name="Arial Black" size="14.0" />
                     </font>
                     <GridPane.margin>
                        <Insets left="20.0" />
                     </GridPane.margin>
                  </Text>
                  <Hyperlink prefHeight="28.0" prefWidth="316.0" text="https://github.com/AdriMartinezMedina/TFG_MediaPlayer"
   GridPane.rowIndex="3" GridPane.valignment="BOTTOM">
                     <GridPane.margin>
                        <Insets bottom="28.0" left="250.0" />
                     </GridPane.margin>
                  </Hyperlink>
                  <Hyperlink layoutX="20.0" layoutY="209.0" prefHeight="23.0" prefWidth="104.0" text="https://openjfx.io/"
   GridPane.halignment="RIGHT" GridPane.rowIndex="4">
                     <GridPane.margin>
                        <Insets right="100.0" />
                     </GridPane.margin>
                  </Hyperlink>
               </children>
            </GridPane>
            <Separator prefWidth="200.0" GridPane.rowIndex="2" />
            <Separator layoutX="10.0" layoutY="268.0" prefWidth="200.0" />
         </children>
      </GridPane>
   </center>
</BorderPane>
```

Style.css

```css
#pane {
    -fx-background-color: black;
}

#openFile {
    -fx-background-image: url("../images/of.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#playVideo {
    -fx-background-image: url("../images/playpause.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#stopVideo {
    -fx-background-image: url("../images/nogo.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#slowVideo {
    -fx-background-image: url("../images/goback.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#fastVideo {
    -fx-background-image: url("../images/goforward.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#normalVideo {
    -fx-background-image: url("../images/go.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#backVideo5 {
    -fx-background-image: url("../images/b5.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#forwardVideo5 {
    -fx-background-image: url("../images/f5.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}

#setVisibilityButtons {
    -fx-background-image: url("../images/eye.png");
    -fx-background-size: 20 20;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
```