



REPRODUCTOR DE VIDEO

Código del proyecto

Adrián Martínez Medina

18 de marzo de 2022

Contenido

MediaPlayer.java.....	2
MediaPlayerController.java	3
MediaPlayer.fxml	10
AcercaDe.fxml	12
Style.css.....	14

MediaPlayer.java

```
1  /**
2   * Clase java usada para ejecutar la app
3   *
4   * Crea la ventana, cargando todos los componentes
5   * a partir de la vista fxml que hemos creado y
6   * preparado en el scene builder
7   *
8   * @author amartinez
9   */
10 package mediaplayer;
11
12 // Listado de imports
13 import javafx.application.Application;
14 import javafx.fxml.FXMLLoader;
15 import javafx.scene.Parent;
16 import javafx.scene.Scene;
17 import javafx.scene.input.MouseEvent;
18 import javafx.stage.Stage;
19
20 public class MediaPlayer extends Application {
21
22     @Override
23     public void start(Stage stage) throws Exception {
24
25         // Creamos el parent root cargando los nodos desde el fxml
26         Parent root = FXMLLoader.load(getClass().getResource("MediaPlayer.fxml"));
27
28         /**
29          * Creamos el objeto que nos dará el contenido de la ventana a partir de
30          * los nodos pasados desde root
31          */
32         Scene scene = new Scene(root);
33
34         // A la escena le añadimos los valores descriptivos que queremos
35         stage.setTitle("Media Player hecho por Adrian");
36         stage.setFullScreenExitHint("Presione ESC or Doble Click para salir de la pantalla completa");
37
38         /**
39          * Añadimos a la escena la capacidad de cambiar entre modo ventana y
40          * modo pantalla completa al hacer doble click sobre el video
41          */
42         scene.setOnMouseClicked((MouseEvent clicked) -> {
43             if (clicked.getClickCount() == 2) {
44                 if (!stage.isFullScreen()) {
45                     stage.setFullScreen(true);
46                 } else {
47                     stage.setFullScreen(false);
48                 }
49             }
50         });
51
52         /**
53          * Añadimos el contenido guardado en la escena a la ventana y la hacemos
54          * visible
55          */
56         stage.setScene(scene);
57         stage.show();
58
59     }
60
61     public static void main(String[] args) {
62
63         launch(args);
64
65     }
66
67 }
```

MediaPlayerController.java

```
1  /**
2   * Clase java usada para las funcionalidades de la app
3   *
4   * Hace a nuestra app capaz de pausar, activar,
5   * parar, acelerar, ralentizar, avanzar y retroceder
6   *
7   * Además, también habilitamos la barra de progreso
8   * del video y la de volumen. Por último también se crean
9   * los keyevents para la funcionalidades
10  *
11  * @author amartinez
12  */
13  package mediaplayer;
14
15  // Listado de imports
16  import java.io.File;
17  import java.io.IOException;
18  import java.net.URL;
19  import java.util.Arrays;
20  import java.util.List;
21  import java.util.ResourceBundle;
22  import javafx.beans.Observable;
23  import javafx.beans.binding.Bindings;
24  import javafx.beans.property.DoubleProperty;
25  import javafx.beans.value.ObservableValue;
26  import javafx.event.ActionEvent;
27  import javafx.fxml.FXML;
28  import javafx.fxml.FXMLLoader;
29  import javafx.fxml.Initializable;
30  import javafx.scene.Parent;
31  import javafx.scene.Scene;
32
33  import javafx.scene.control.Slider;
34  import javafx.scene.input.KeyCode;
35  import javafx.scene.input.KeyEvent;
36  import javafx.scene.input.MouseEvent;
37  import javafx.scene.media.Media;
38  import javafx.scene.media.MediaPlayer;
39  import javafx.scene.media.MediaView;
40  import static javafx.scene.media.MediaPlayer.Status.PLAYING;
41  import javafx.stage.FileChooser;
42  import javafx.stage.Stage;
43  import javax.swing.JOptionPane;
44
45  public class MediaPlayerController implements Initializable {
46
47      // Variable que almacena la ruta al video abierto
48      @FXML
49      private String path;
50
51      /**
52       * Variable que almacena todas las extensiones de archivo válidas para
53       * nuestra aplicación en un array
54       */
55      @FXML
56      private List<String> extensions;
57
58      // Variable objeto, crea el mediaPlayer
59      @FXML
60      private MediaPlayer mediaPlayer;
61
62      // Variable objeto, crea el mediaView
63      @FXML
```

```

63 private MediaView mediaView;
64
65 // Variable objeto, crea el slider de volumen
66 @FXML
67 private Slider volumeSlider;
68
69 // Variable objeto, crea el slider de progreso de video
70 @FXML
71 private Slider progressBar;
72
73 /**
74  * Método que usamos para añadir al boton de abrir archivo la capacidad de
75  * hacerlo.
76  *
77  * @param event
78  */
79 @FXML
80 private void OpenFileMethod(ActionEvent event) {
81
82     try {
83         /**
84          * llenamos el array de strings con las extensiones permitidas en
85          * nuestra app
86          */
87         extensions = Arrays.asList("*.mp4", "*.3gp", "*.mkv", "*.MP4", "*.MKV", "*.3GP", "*.flv", "*.wmv");
88
89         /**
90          * Creamos tanto el objeto filechooser para elegir archivo como el
91          * objeto extension filter para añadir el filtro de extensiones que
92          * hemos concretado en el array anterior
93          */
94
95         FileChooser fileChooser = new FileChooser();
96         FileChooser.ExtensionFilter filter = new FileChooser.ExtensionFilter("selecciona video", extensions);
97
98         // Añadimos el filtro de extensiones al filechooser
99         fileChooser.getExtensionFilters().add(filter);
100         fileChooser.setTitle("Selecciona un video");
101
102         /**
103          * Abrimos la ventana para elegir archivo y este se guarda como
104          * variable File
105          *
106          * Posteriormente guardamos en un string la ruta del archivo elegido
107          * para ser usada en el objeto Media
108          */
109         File file = fileChooser.showOpenDialog(null);
110         path = file.toURI().toString();
111
112         /**
113          * Siempre y cuando se haya elegido un archivo, se realizará el
114          * código contenido en el if
115          */
116         if (path != null) {
117             /**
118              * Creamos el objeto Media gracias al path, luego lo usamos para
119              * crear el MediaPlayer y finalmente creamos la vista del video
120              */
121             Media media = new Media(path);
122             mediaPlayer = new MediaPlayer(media);
123             mediaView.setMediaPlayer(mediaPlayer);
124
125             /**

```

```

125     * Las líneas que tenemos a continuación sirven para definir las
126     * dimensiones del video cargado ajustandolo al mediaView
127     */
128     DoubleProperty widthProp = mediaView.fitWidthProperty();
129     DoubleProperty heightProp = mediaView.fitHeightProperty();
130     widthProp.bind(Bindings.selectDouble(mediaView.sceneProperty(), "width"));
131     heightProp.bind(Bindings.selectDouble(mediaView.sceneProperty(), "height"));
132
133     // Damos un valor inicial al slider del volumen del video
134     volumeSlider.setValue(mediaPlayer.getVolume() * 100);
135
136     /**
137     * Creamos el listener que permitirá ajustar el volumen del
138     * video según donde coloquemos el punto del slider del volumen
139     */
140     volumeSlider.valueProperty().addListener((Observable observable) -> {
141         mediaPlayer.setVolume(volumeSlider.getValue() / 100);
142     });
143
144     // Damos un valor inicial a la barra de progreso del video
145     mediaPlayer.currentTimeProperty().addListener((ObservableValue<? extends javafx.util.Duration> observable,
146     javafx.util.Duration oldValue, javafx.util.Duration newValue) -> {
147         progressBar.setValue(newValue.toSeconds());
148     });
149
150     /**
151     * Creamos los listener que permitirá ajustar el volumen del
152     * video según donde coloquemos el punto del slider del volumen
153     */
154     progressBar.setOnMousePressed((MouseEvent event1) -> {
155         mediaPlayer.seek(javafx.util.Duration.seconds(progressBar.getValue()));
156     });
157
158     progressBar.setOnMouseDragged((MouseEvent event1) -> {
159         mediaPlayer.seek(javafx.util.Duration.seconds(progressBar.getValue()));
160     });
161     mediaPlayer.setOnReady(() -> {
162         javafx.util.Duration total = media.getDuration();
163         progressBar.setMax(total.toSeconds());
164     });
165
166     // Por último, iniciamos la reproducción del video al abrirlo
167     mediaPlayer.play();
168
169     } catch (Exception e) {
170     }
171
172     }
173
174     @Override
175     public void initialize(URL url, ResourceBundle rb) {
176         // TODO
177     }
178
179     /**
180     * Método que usamos para añadir al botón de play & pause video la capacidad
181     * de hacerlo.
182     * @param event
183     */
184     @FXML
185     private void playVideo(ActionEvent event) {
186
187         /**

```

```

187     * Siempre y cuando se haya elegido previamente un video, se comprueba
188     * si el video está activo o pausado
189     *
190     * Si está activo, se pausa y si está pausado se reactiva
191     */
192     try {
193         if (mediaPlayer.getStatus() == PLAYING) {
194             mediaPlayer.pause();
195         } else {
196             mediaPlayer.play();
197         }
198     } catch (Exception e) {
199         JOptionPane.showMessageDialog(null, "Elige un video primero");
200     }
201
202 }
203
204 /**
205  * Método que usamos para añadir al boton de stop video la capacidad de
206  * hacerlo.
207  *
208  * @param event
209  */
210 @FXML
211 private void stopVideo(ActionEvent event) {
212
213     /**
214      * Siempre y cuando se haya elegido previamente un video, se detiene el
215      * video reseteandolo
216      */
217     try {
218
219         mediaPlayer.stop();
220     } catch (Exception e) {
221         JOptionPane.showMessageDialog(null, "Elige un video primero");
222     }
223 }
224
225 /**
226  * Método que usamos para añadir al boton de avanzar video la capacidad de
227  * hacerlo.
228  *
229  * @param event
230  */
231 @FXML
232 private void skip5(ActionEvent event) {
233
234     /**
235      * Siempre y cuando se haya elegido previamente un video, se adelanta el
236      * video 5 segundos
237      */
238     try {
239         mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javaafx.util.Duration.seconds(5)));
240     } catch (Exception e) {
241         JOptionPane.showMessageDialog(null, "Elige un video primero");
242     }
243 }
244
245 /**
246  * Método que usamos para añadir al boton de acelerar video la capacidad de
247  * hacerlo.

```

```

249     *
250     * @param event
251     */
252     @FXML
253     private void furtherSpeedUpVideo(ActionEvent event) {
254
255         /**
256          * Siempre y cuando se haya elegido previamente un video, se acelera el
257          * video
258          */
259         try {
260             mediaPlayer.setRate(2);
261         } catch (Exception e) {
262             JOptionPane.showMessageDialog(null, "Elige un video primero");
263         }
264     }
265
266     /**
267     * Método que usamos para añadir al boton de atrasar video la capacidad de
268     * hacerlo.
269     *
270     * @param event
271     */
272     @FXML
273     private void back5(ActionEvent event) {
274
275         /**
276          * Siempre y cuando se haya elegido previamente un video, se atrasa el
277          * video 5 segundos
278          */
279
280         try {
281             mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javaafx.util.Duration.seconds(-5)));
282         } catch (Exception e) {
283             JOptionPane.showMessageDialog(null, "Elige un video primero");
284         }
285     }
286
287     /**
288     * Método que usamos para añadir al boton de ralentizar video la capacidad
289     * de hacerlo.
290     *
291     * @param event
292     */
293     @FXML
294     private void furtherSlowDownVideo(ActionEvent event) {
295
296         /**
297          * Siempre y cuando se haya elegido previamente un video, se ralentiza
298          * el video
299          */
300         try {
301             mediaPlayer.setRate(0.5);
302         } catch (Exception e) {
303             JOptionPane.showMessageDialog(null, "Elige un video primero");
304         }
305     }
306
307     /**
308     * Método que usamos para añadir al boton de normalizar velocidad video la
309     */
310

```



```

311     * capacidad de hacerlo.
312     *
313     * @param event
314     */
315     @FXML
316     private void normalSpeedVideo(ActionEvent event) {
317
318         /**
319          * Siempre y cuando se haya elegido previamente un video, se resetea la
320          * velocidad de reproducción del video
321          */
322         try {
323             mediaPlayer.setRate(1);
324         } catch (Exception e) {
325             JOptionPane.showMessageDialog(null, "Elige un video primero");
326         }
327     }
328 }
329
330 /**
331  * Método que usamos para abrir la ventana Acerca De al hacer click en el
332  * menú de la app
333  *
334  * @param event
335  */
336     @FXML
337     private void openAcercaDe(ActionEvent event) {
338         try {
339             // Creamos el parent root cargando los nodos desde el fxml
340             Parent root = FXMLLoader.load(getClass().getResource("AcercaDe.fxml"));
341
342             /**
343              * Creamos el objeto que nos dará el contenido de la ventana a
344
345              * partir de los nodos pasados desde root
346              */
347             Scene scene = new Scene(root);
348
349             // Creamos la ventana acerca de
350             Stage acercade = new Stage();
351
352             /**
353              * Añadimos el contenido guardado en la escena a la ventana y la
354              * hacemos visible
355              */
356             acercade.setScene(scene);
357             acercade.show();
358         } catch (IOException e) {
359         }
360     }
361
362     /**
363      * Método que contiene todos los keylisteners de la app, haciendo posible la
364      * ejecución de las funciones descritas anteriormente segun la combinación
365      * de teclas pulsada.
366      *
367      * @param ke
368      */
369     @FXML
370     private void handleKeyPressed(KeyEvent ke) {
371
372         if (ke.getCode().equals(KeyCode.P) && ke.isControlDown()) {
373             try {
374                 if (mediaPlayer.getStatus() == PLAYING) {
375                     mediaPlayer.pause();
376                 }
377             }
378         }
379     }

```

```

375         } else {
376             mediaPlayer.play();
377         }
378     } catch (Exception e) {
379         JOptionPane.showMessageDialog(null, "Elige un video primero");
380     }
381 }
382
383 if (ke.getCode().equals(KeyCode.S) && ke.isControlDown()) {
384     try {
385         mediaPlayer.stop();
386     } catch (Exception e) {
387         JOptionPane.showMessageDialog(null, "Elige un video primero");
388     }
389 }
390
391 if (ke.getCode().equals(KeyCode.RIGHT) && ke.isControlDown()) {
392     try {
393         mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(5)));
394     } catch (Exception e) {
395         JOptionPane.showMessageDialog(null, "Elige un video primero");
396     }
397 }
398
399 if (ke.getCode().equals(KeyCode.LEFT) && ke.isControlDown()) {
400     try {
401         mediaPlayer.seek(mediaPlayer.getCurrentTime().add(javafx.util.Duration.seconds(-5)));
402     } catch (Exception e) {
403         JOptionPane.showMessageDialog(null, "Elige un video primero");
404     }
405 }
406
407 if (ke.getCode().equals(KeyCode.UP) && ke.isControlDown()) {
408     try {
409         mediaPlayer.setRate(2);
410     } catch (Exception e) {
411         JOptionPane.showMessageDialog(null, "Elige un video primero");
412     }
413 }
414
415 if (ke.getCode().equals(KeyCode.DOWN) && ke.isControlDown()) {
416     try {
417         mediaPlayer.setRate(0.5);
418     } catch (Exception e) {
419         JOptionPane.showMessageDialog(null, "Elige un video primero");
420     }
421 }
422
423 if (ke.getCode().equals(KeyCode.N) && ke.isControlDown()) {
424     try {
425         mediaPlayer.setRate(1);
426     } catch (Exception e) {
427         JOptionPane.showMessageDialog(null, "Elige un video primero");
428     }
429 }
430
431 if (ke.getCode().equals(KeyCode.A) && ke.isControlDown()) {
432     try {
433         Parent root = FXMLLoader.load(getClass().getResource("AcercaDe.fxml"));
434
435         Scene scene = new Scene(root);
436         Stage acerca = new Stage();
437
438         acerca.setScene(scene);
439         acerca.show();
440     } catch (IOException e) {
441     }
442 }
443
444 }
445
446 }

```

MediaPlayer.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Button?>
5 <?import javafx.scene.control.Menu?>
6 <?import javafx.scene.control.MenuBar?>
7 <?import javafx.scene.control.MenuItem?>
8 <?import javafx.scene.control.Slider?>
9 <?import javafx.scene.effect.InnerShadow?>
10 <?import javafx.scene.layout.BorderPane?>
11 <?import javafx.scene.layout.HBox?>
12 <?import javafx.scene.layout.StackPane?>
13 <?import javafx.scene.layout.VBox?>
14 <?import javafx.scene.media.MediaView?>
15
16 <BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
17 onKeyPressed="#handleKeyPressed" prefHeight="400.0" prefWidth="600.0" stylesheets="@style.css"
18 xmlns="http://javafx.com/javafx/17" xmlns:fx="http://javafx.com/fxml/1"
19 fx:controller="mediaplayer.MediaPlayerController">
20     <center>
21         <StackPane fx:id="pane" prefHeight="300.0" prefWidth="609.0"
22         BorderPane.alignment="CENTER">
23             <children>
24                 <MediaView fx:id="mediaView" fitHeight="300.0" fitWidth="600.0">
25                     <effect>
26
27                         <InnerShadow />
28                     </effect>
29                 </MediaView>
30                 <VBox alignment="BOTTOM_CENTER" prefHeight="39.0" prefWidth="609.0">
31                     <children>
32                         <Slider fx:id="progressBar" prefHeight="0.0" prefWidth="600.0" />
33                         <HBox alignment="BOTTOM_CENTER" opacity="0.8" prefHeight="39.0"
34                         prefWidth="576.0">
35                             <children>
36                                 <Button fx:id="openFile" alignment="BOTTOM_LEFT"
37                                 mnemonicParsing="false" onAction="#OpenFileMethod" prefWidth="35.0">
38                                     <HBox.margin>
39                                         <Insets bottom="7.0" right="14.0" />
40                                     </HBox.margin>
41                                 </Button>
42                                 <Button fx:id="playVideo" alignment="BOTTOM_LEFT"
43                                 mnemonicParsing="false" onAction="#playVideo" prefWidth="35.0">
44                                     <HBox.margin>
45                                         <Insets bottom="7.0" right="7.0" />
46                                     </HBox.margin>
47                                 </Button>
48                                 <Button fx:id="stopVideo" alignment="BOTTOM_LEFT"
49                                 mnemonicParsing="false" onAction="#stopVideo" prefWidth="35.0">
50                                     <HBox.margin>
51                                         <Insets bottom="7.0" right="14.0" />
52                                     </HBox.margin>
53                                 </Button>
54                             </children>
55                         </HBox>
56                     </children>
57                 </VBox>
58             </children>
59         </StackPane>
60     </center>
61 </BorderPane>
```

```

43         </HBox.margin>
44     </Button>
45     <Button fx:id="slowVideo" alignment="BOTTOM_LEFT"
mnemonicParsing="false" onAction="#furtherSlowDownVideo" prefWidth="35.0">
46         <HBox.margin>
47             <Insets bottom="7.0" right="7.0" />
48         </HBox.margin>
49     </Button>
50     <Button fx:id="backVideo5" alignment="BOTTOM_LEFT"
mnemonicParsing="false" onAction="#back5" prefWidth="35.0">
51         <HBox.margin>
52             <Insets bottom="7.0" right="7.0" />
53         </HBox.margin>
54     </Button>
55     <Button fx:id="normalVideo" alignment="BOTTOM_LEFT"
mnemonicParsing="false" onAction="#normalSpeedVideo" prefWidth="35.0">
56         <HBox.margin>
57             <Insets bottom="7.0" right="7.0" />
58         </HBox.margin>
59     </Button>
60     <Button fx:id="forwardVideo5" alignment="BOTTOM_LEFT"
mnemonicParsing="false" onAction="#skip5" prefWidth="35.0">
61         <HBox.margin>
62             <Insets bottom="7.0" right="7.0" />
63         </HBox.margin>
64     </Button>
65     <Button fx:id="fastVideo" alignment="BOTTOM_LEFT"
mnemonicParsing="false" onAction="#furtherSpeedUpVideo" prefWidth="35.0">
66         <HBox.margin>
67             <Insets bottom="7.0" right="21.0" />
68         </HBox.margin>
69     </Button>
70     <Slider fx:id="volumeSlider" stylesheets="@style.css">
71         <HBox.margin>
72             <Insets bottom="10.0" />
73         </HBox.margin>
74     </Slider>
75 </children>
76 </HBox>
77 </children>
78 </VBox>
79 <MenuBar prefHeight="0.0" prefWidth="600.0" StackPane.alignment="TOP_CENTER">
80     <menus>
81         <Menu mnemonicParsing="false" text="Ayuda">
82             <items>
83                 <MenuItem fx:id="menuAcercaDe" mnemonicParsing="false"
onAction="#openAcercaDe" text="Acerca De" />
84             </items>
85         </Menu>
86     </menus>
87 </MenuBar>
88 </children>
89 </StackPane>
90 </center>
91 </BorderPane>

```

AcercaDe.fxml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Hyperlink?>
5 <?import javafx.scene.control.Separator?>
6 <?import javafx.scene.layout.BorderPane?>
7 <?import javafx.scene.layout.ColumnConstraints?>
8 <?import javafx.scene.layout.GridPane?>
9 <?import javafx.scene.layout.RowConstraints?>
10 <?import javafx.scene.text.Font?>
11 <?import javafx.scene.text.Text?>
12
13 <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">
14     <top>
15         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Acerca De" underline="true"
BorderPane.alignment="CENTER">
16             <font>
17                 <Font name="Arial Black" size="36.0" />
18             </font>
19             <BorderPane.margin>
20                 <Insets top="20.0" />
21             </BorderPane.margin>
22         </Text>
23     </top>
24     <bottom>
25         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="(c) Copyright amartinez Junio, 2022.
Todos los derechos reservados" textAlignment="CENTER" BorderPane.alignment="CENTER">
26             <BorderPane.margin>
27                 <Insets bottom="10.0" />
28             </BorderPane.margin>
29             <font>
30                 <Font name="Arial Black" size="12.0" />
31             </font>
32
33         </Text>
34     </bottom>
35     <center>
36         <GridPane BorderPane.alignment="CENTER">
37             <columnConstraints>
38                 <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
39             </columnConstraints>
40             <rowConstraints>
41                 <RowConstraints maxHeight="99.0" minHeight="0.0" prefHeight="0.0" vgrow="SOMETIMES" />
42                 <RowConstraints maxHeight="274.0" minHeight="10.0" prefHeight="274.0" vgrow="SOMETIMES" />
43                 <RowConstraints maxHeight="61.0" minHeight="0.0" prefHeight="0.0" vgrow="SOMETIMES" />
44             </rowConstraints>
45             <children>
46                 <GridPane GridPane.rowIndex="1">
47                     <columnConstraints>
48                         <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
49                     </columnConstraints>
50                     <rowConstraints>
51                         <RowConstraints maxHeight="62.0" minHeight="10.0" prefHeight="62.0"
vgrow="SOMETIMES" />
52                         <RowConstraints maxHeight="68.0" minHeight="0.0" prefHeight="23.0"
vgrow="SOMETIMES" />
53                     </rowConstraints>
54                 </GridPane>
55             </children>
56         </GridPane>
57     </center>
58 </BorderPane>
```

```

52         <RowConstraints maxHeight="73.0" minHeight="0.0" prefHeight="23.0"
vgrow="SOMETIMES" />
53         <RowConstraints maxHeight="148.0" minHeight="10.0" prefHeight="124.0"
vgrow="SOMETIMES" />
54         <RowConstraints maxHeight="115.0" minHeight="10.0" prefHeight="42.0"
vgrow="SOMETIMES" />
55     </rowConstraints>
56     <children>
57         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Reproductor de video hecho
por Adrián Martínez Medina">
58             <font>
59                 <Font name="Arial Black" size="14.0" />
60             </font>
61             <GridPane.margin>
62                 <Insets left="20.0" />
63             </GridPane.margin>
64         </Text>
65         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Versión: 1.0"
wrappingWidth="108.0" GridPane.rowIndex="1">
66             <font>
67                 <Font name="Arial Black" size="14.0" />
68             </font>
69             <GridPane.margin>
70                 <Insets left="20.0" />
71             </GridPane.margin>
72         </Text>
73         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Build: 20220318-001"
wrappingWidth="204.599609375" GridPane.rowIndex="2">
74             <font>
75                 <Font name="Arial Black" size="14.0" />
76             </font>
77             <GridPane.margin>
78                 <Insets left="20.0" />
79             </GridPane.margin>
80         </Text>
81         <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">
82             <font>
83                 <Font name="Arial Black" size="14.0" />
84             </font>
85             <GridPane.margin>
86                 <Insets left="20.0" />
87             </GridPane.margin>
88         </Text>
89         <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Esta aplicación ha sido
desarrollada con JavaFX:" y="1.0" GridPane.rowIndex="4">
90             <font>
91                 <Font name="Arial Black" size="14.0" />
92             </font>
93             <GridPane.margin>
94                 <Insets left="20.0" />
95             </GridPane.margin>
96         </Text>
97         <Hyperlink prefHeight="28.0" prefWidth="316.0"
text="https://github.com/AdriMartinezMedina/TFG_MediaPlayer" GridPane.rowIndex="3"
GridPane.valignment="BOTTOM">
98             <GridPane.margin>
99                 <Insets bottom="28.0" left="250.0" />
100             </GridPane.margin>
101         </Hyperlink>

```



```

102         <Hyperlink layoutX="20.0" layoutY="209.0" prefHeight="23.0" prefWidth="104.0"
text="https://openjfx.io/" GridPane.halignment="RIGHT" GridPane.rowIndex="4">
103             <GridPane.margin>
104                 <Insets right="100.0" />
105             </GridPane.margin>
106         </Hyperlink>
107     </children>
108 </GridPane>
109 <Separator prefWidth="200.0" GridPane.rowIndex="2" />
110 <Separator layoutX="10.0" layoutY="268.0" prefWidth="200.0" />
111 </children>
112 </GridPane>
113 </center>
114 </BorderPane>

```

Style.css

```

1 #pane {
2     -fx-background-color: black;
3 }
4
5 #openFile {
6     -fx-background-image: url("../images/of.png");
7     -fx-background-size: 20 20;
8     -fx-background-repeat: no-repeat;
9     -fx-background-position: center;
10 }
11
12 #playVideo {
13     -fx-background-image: url("../images/playpause.png");
14     -fx-background-size: 20 20;
15     -fx-background-repeat: no-repeat;
16     -fx-background-position: center;
17 }
18
19 #stopVideo {
20     -fx-background-image: url("../images/nogo.png");
21     -fx-background-size: 20 20;
22     -fx-background-repeat: no-repeat;
23     -fx-background-position: center;
24 }
25

```

```
26 #slowVideo {
27     -fx-background-image: url("../images/goback.png");
28     -fx-background-size: 20 20;
29     -fx-background-repeat: no-repeat;
30     -fx-background-position: center;
31 }
32
33 #fastVideo {
34     -fx-background-image: url("../images/goforward.png");
35     -fx-background-size: 20 20;
36     -fx-background-repeat: no-repeat;
37     -fx-background-position: center;
38 }
39
40 #normalVideo {
41     -fx-background-image: url("../images/go.png");
42     -fx-background-size: 20 20;
43     -fx-background-repeat: no-repeat;
44     -fx-background-position: center;
45 }
46
47 #backVideo5 {
48     -fx-background-image: url("../images/b5.png");
49     -fx-background-size: 20 20;
50     -fx-background-repeat: no-repeat;
51     -fx-background-position: center;
52 }
53
54 #forwardVideo5 {
55     -fx-background-image: url("../images/f5.png");
56     -fx-background-size: 20 20;
57     -fx-background-repeat: no-repeat;
58     -fx-background-position: center;
59 }
```