***“Tema del proyecto práctico”***

**Fundamentos de Programación**

*Universidad Técnica Particular de Loja*

Versión [1.0.1]

***CALIFICACIÓNES***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *CriterioS / calificación* | ESTADO | CALIDAD | Originalidad | Diseño | Funcionamiento | Defensa |
| Iniciado, en desarrollo, finalizado | POO, tiempos de respt. código, variables, comentarios | Enviados a hacer, copia, original | Principiante, moderado, avanzado, excelente. | No funciona, funciona con errores, funciona. | Mala, regular, buena, muy B. Excelente. |
| Calificación |  |  |  |  |  |  |

Información del Documento

|  |  |
| --- | --- |
| Título: | Reproductor de mp3 |
| Versión: | [1.0.1] |
| Archivo: | JavaFX |
| Autores: | Alexis Fabian Montoya Parra,  Luis Enrique Ortiz Rivera,  Luis Miguel Rojas Paccha |
| Estado: | Finalizado |

Lista de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Descripción |
| 1.0.1 | 31/01/2018 | ALL | Emisión Final |
|  |  |  |  |
|  |  |  |  |

Firmas y Aprobaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Elaborado Por: | *Alexis Montoya,*  *Luis Enrique,*  *Luis Rojas* | |  |
| Fecha: | 31/1/2018 | Firma: |

|  |  |  |  |
| --- | --- | --- | --- |
| Revisado Por: | *Luis Alfredo Jumbo Flores* | |  |
| Fecha: | 31/1/2018 | Firma: |

|  |  |  |  |
| --- | --- | --- | --- |
| Aprobado Por: | *Luis Alfredo Jumbo Flores* | |  |
| Fecha: | 31/1/2018 | Firma: |
|  |  |  |  |

Contenido

[1. Planteamiento 1](#_Toc505135324)

[1.1. Descripción del problema 1](#_Toc505135325)

[1.2. Descripción de la solución 1](#_Toc505135326)

[1.3. Referencias 1](#_Toc505135327)

[2. Análisis y diseño de la Solución 2](#_Toc505135328)

[2.1. Análisis 2](#_Toc505135329)

[2.2. Diseño de la solución 2](#_Toc505135330)

[3. Diseño de la UML 3](#_Toc505135331)

[4. Documentación 3](#_Toc505135332)

[5. Conclusiones 7](#_Toc505135333)

[6. Recomendaciones 7](#_Toc505135334)

***“Reproductor de mp3”***

**Programación de Algoritmos**

*Universidad Técnica Particular de Loja*

# Planteamiento

Para realizar la creación de software se necesitan conocimientos de nivel medio en diferentes interfaces de programación y es por ello que existe un sinnúmero de software que son creados para diferentes realizaciones de trabajo y diseñados con eficacia. Al pasar de los días surgen nuevos tipos de problemas que el ser humano desea resolver rápidamente; es por ello que la ingeniería en Sistemas Computacionales está en constante desarrollo e investigación para ayudar a solventar problemas que para el ser humano se hacen complejos y así optimizar el tiempo de trabajo.

Nuestra actividad propuesta para el proyecto es de “Realizar un reproductor mp3 basado en JavaFX”.

## Descripción del problema

La problemática que hoy en día hemos encontrado es las diferentes plataformas de reproductores de música, por ello hemos decidido implementar una nuevo software que facilite la reproducción de música, debido a que los usuarios desean una aplicación que faciliten el uso para reproducir música. Otro motivo es que los usuarios buscan una interfaz animada, elegante y de fácil uso. Por tal motivo, nosotros como integrantes del grupo, queremos plantear una eficaz solución de la problemática presentada anteriormente, diseñaremos un software que sea implementado para ordenadores.

## Descripción de la solución

El desarrollado de este software será bajo los estándares de adopción el cual será un aplicativo de fácil uso para el usuario, así mismo permitirá ejecutar una adecuada función y correcta organización realizando varios experimentos de la eficacia del mismo, de ser factible se implementará futuras mejoras e innovaciones bajo la acción y necesidad del usuario.

La aplicación reproducirá las canciones agregadas y agregar lista de reproducción.

Por ende, la solución permitirá:

* Elaboración de software que facilite al usuario reproducir la música en un entorno dinámico.
* El usuario podrá seleccionar la canción de su gusto y poder reproducirla.

## Referencias

|  |  |
| --- | --- |
| **Recursos Bibliográficos.** | |
| **[1]** | DEITEL, PAUL J. Y HARVEY M. DEITEL “CÓMO PROGRAMAR EN JAVA”. Séptima edición. PEARS ON EDUCACIÓN, México 2008. |
|  |  |
| **Recursos Electrónicos** | |
| **[2]** | https://es.stackoverflow.com/ |
| **[3]** | https://programacionjavafx.blogspot.com/ |
| **[4]** | http://naidoprograms.blogspot.com/2015/11/reproductor-mp3-con-javafx-1era-parte.html |
| **[5]** | http://naidoprograms.blogspot.com/2015/11/reproductor-mp3-con-javafx-2da-parte.html |
| **[6]** | http://naidoprograms.blogspot.com/2015/11/reproductor-mp3-con-javafx-3da-parte.html |

# Análisis y diseño de la Solución

## Análisis

El presente desarrollo del trabajo es con fines educativos por lo cual, para la implementación de un reproductor de música se necesitaran conocimientos en programación orientada a objetos. Para que el programa tenga una mejor acogida por el usuario, las interfaces que lo conforman serán presentadas y al momento de reproducir o agregar músicas a una lista de reproducción será de manera sencilla.

## Diseño de la solución

**Datos de entrada:**

Nombre de la canción a agregar en el aplicativo.

**Procesamientos de estos datos:**

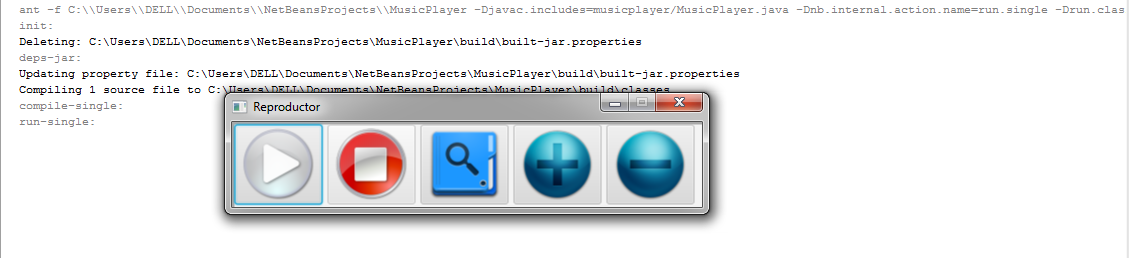
Una vez ingresado los campos requeridos, se leerá y almacenará con el objetivo de mantener un entorno correctamente organizando y debidamente procesado sin tener perdida de información proporcionada.

**Salidas o resultados esperados:**

Se genera la melodía de la canción de la música agregada a nuestro aplicativo.

**Diseño de la Interface.**

**(Captura de la pantalla del programa finalizado)**

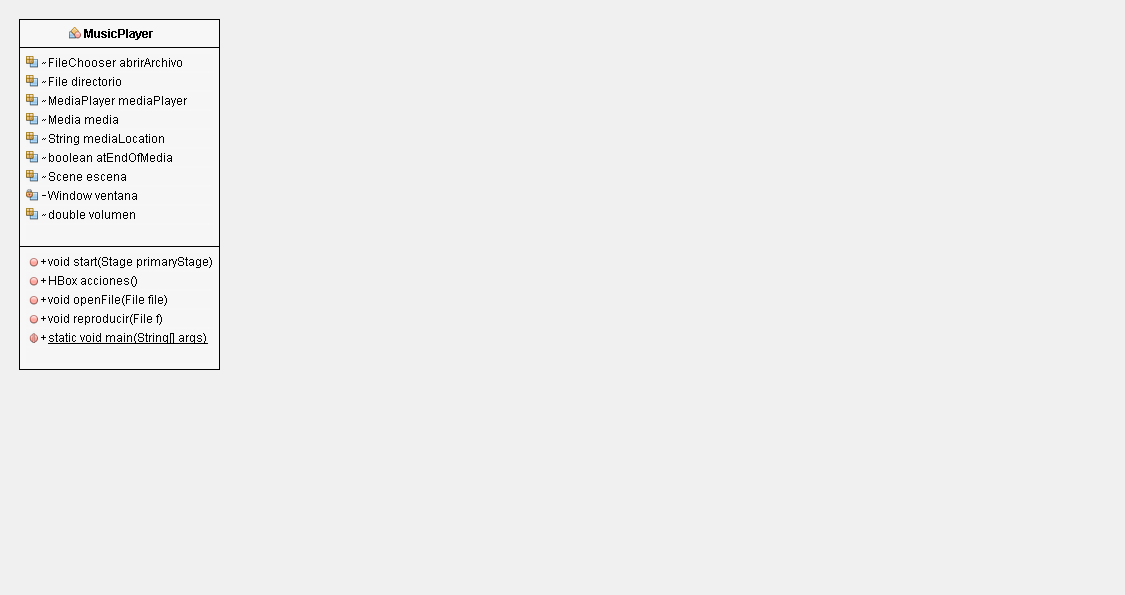
**

**Descripción de sus funcionalidades:**

Es una barra de herramientas en la cual contiene botones con su respectivo funcionamiento.

* En la ventana dispone de 5 botones, cada uno de ellos desempeña una función específica:
  + Botón play: Reproduce la canción.
  + Botón stop: Detiene la canción.
  + Botón buscar: Busca la canción y la agrega para su reproducción.
  + Botón más (volumen): Subir volumen.
  + Botón menos (Volumen): Bajar volumen.

# Diseño de la UML



Para la realización del trabajo, hemos considerado pertinente la utilización de varios métodos los cuales ayudarán a desarrollar un entorno factible. Como podemos apreciar en la imagen disponemos de 9 métodos de los cuales, el más relevante es

# Documentación

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180 | **package** ReproductorMP3;  **import** **java.io.File**;  **import** **javafx.application.Application**;  **import** **javafx.event.ActionEvent**;  **import** **javafx.event.EventHandler**;  **import** **javafx.scene.Scene**;  **import** **javafx.scene.control.Button**;  **import** **javafx.scene.image.Image**;  **import** **javafx.scene.image.ImageView**;  **import** **javafx.scene.layout.HBox**;  **import** **javafx.scene.media.Media**;  **import** **javafx.scene.media.MediaPlayer**;  **import** **javafx.scene.media.MediaPlayer.Status**;  **import** **javafx.scene.media.MediaView**;  **import** **javafx.scene.paint.Color**;  **import** **javafx.stage.Stage**;  **import** **javafx.stage.FileChooser**;  **import** **javafx.stage.Window**;  **import** **javax.swing.JOptionPane**;  /\*\*  \*  \* @author Integrantesdelgrupo  \*/  **public** **class** **MusicPlayer** **extends** Application {  //Variables ha utilizar  **final** FileChooser abrirArchivo = **new** FileChooser();  File directorio = **new** File("C:/Users/DELL/Documents/NetBeansProjects/MusicPlayer/src/musicplayer/Faded.mp3");  MediaPlayer mediaPlayer;  Media media;  String mediaLocation;  **boolean** atEndOfMedia = **false**;  Scene escena;  **private** Window ventana;  **double** volumen = **0.1**;  **@Override**  **public** **void** **start**(Stage primaryStage) {  primaryStage.setTitle("Reproductor");  escena = **new** Scene(acciones(), **640**, **75**, Color.DARKGRAY);//se instancia el objeto escena |||||y se le envia comom parametro el metodo acciones  reproducir(directorio); //se le pasa el medoto la direccion almacendad en directorio  primaryStage.setScene(escena);// se envia la ventana "escena" al marco primaryStage  primaryStage.show();//se le establece estabilidad al llamar a este metodo;  primaryStage.setResizable(**false**);  }  **public** HBox **acciones**() {//se crea los botones y se asigna imagenes  Image playimagen = **new** Image(getClass().getResourceAsStream("play.png"));  Button iniciar = **new** Button("", **new** ImageView(playimagen));  Image stopimagen = **new** Image(getClass().getResourceAsStream("stop.png"));  Button detener = **new** Button("", **new** ImageView(stopimagen));  Image buscarimagen = **new** Image(getClass().getResourceAsStream("buscar.png"));  Button buscar = **new** Button("", **new** ImageView(buscarimagen));  Image masimagen = **new** Image(getClass().getResourceAsStream("mas.png"));  Button masvolumen = **new** Button("", **new** ImageView(masimagen));  Image menosimagen = **new** Image(getClass().getResourceAsStream("menos.png"));  Button menosvolumen = **new** Button("", **new** ImageView(menosimagen));  Image agregarimagen = **new** Image(getClass().getResourceAsStream("agregar.png"));  Button agregar = **new** Button("", **new** ImageView(agregarimagen));  Image listaimagen = **new** Image(getClass().getResourceAsStream("lista.png"));  Button lista = **new** Button("", **new** ImageView(listaimagen));  HBox botones = **new** HBox(**5**);//Espacio entre cada boton  botones.getChildren().addAll(iniciar, detener, buscar, masvolumen, menosvolumen, agregar, lista);//Activar y/o desactivar botones  botones.setLayoutX(**2.5**);//mover botones  botones.setLayoutY(**2.5**);//mover botones  iniciar.setOnAction(**new** EventHandler<ActionEvent>() {//Action permormad  **@Override**  **public** **void** **handle**(ActionEvent e  ) {  Status status = mediaPlayer.getStatus();  **if** (status == Status.UNKNOWN || status == Status.HALTED) {  **return**;  }  **if** (status == Status.PAUSED || status == Status.READY || status == Status.STOPPED) {  **if** (atEndOfMedia) {  mediaPlayer.seek(mediaPlayer.getStartTime());  }  mediaPlayer.play();  } **else** {  mediaPlayer.pause();  }  }  });// Final  detener.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  mediaPlayer.stop();  }  });  buscar.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  directorio = abrirArchivo.showOpenDialog(ventana);  **if** (directorio != **null**) {  openFile(directorio);  }  }  });  agregar.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  directorio = abrirArchivo.showOpenDialog(ventana);  **if** (directorio != **null**) {  openFile(directorio);  }  }  });  lista.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  openFile(playList);  }  });  masvolumen.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  **if** (volumen < **1.0**) {  volumen = volumen + **0.05**;  mediaPlayer.setVolume(volumen);  }  }  });  menosvolumen.setOnAction(**new** EventHandler<ActionEvent>() {  **@Override**  **public** **void** **handle**(ActionEvent e) {  **if** (volumen > **0.0**) {  volumen = volumen - **0.05**;  mediaPlayer.setVolume(volumen);  }  }  });  **return** botones;  }  **public** **void** **openFile**(File file) {//Abrir archivos  **try** {  mediaPlayer.stop();  reproducir(file);  } **catch** (Exception e) {  JOptionPane.showMessageDialog(**null**, "Error al abrir archivo: " + e);  }  }  **public** **void** **reproducir**(File f) {//reproducir archivos  **try** {  mediaLocation = f.toURI().toURL().toExternalForm();  Media media = **new** Media(mediaLocation);  mediaPlayer = **new** MediaPlayer(media);  mediaPlayer.setAutoPlay(**true**);  mediaPlayer.setVolume(volumen);  MediaView mediaView = **new** MediaView(mediaPlayer);  } **catch** (Exception e) {  JOptionPane.showMessageDialog(**null**, "Error al arbrir archivo: " + e);  }  }  /\*\*  \* @param args the command line arguments  \*/  **public** **static** **void** **main**(String[] args) {  launch(args);  }  } |

# Conclusiones

* El compromiso del grupo en las entregas respectivas a las fechas del cronograma fue fundamental para un buen desempeño en el desarrollo del proyecto.
* Mantener un sistema libre de riesgos y problemas al momento de utilizarlo.

# Recomendaciones

* Al momento de continuar con el presente proyecto se recomienda aprovechar las funcionalidades ya desarrolladas y expandirlas de tal forma que el aplicativo no solo proporcionen la reproducción de música sino también de almacenamiento de los datos y poder mostrarlas en la misma interfaz, este beneficio abre la puerta a muchas ideas innovadoras que pueden ser aprovechadas por el desarrollador de futuras implementaciones.
* Implementar nuevos métodos las cuales ayuden a un mejor rendimiento y presentación del trabajo realizado.