



Universidad de las Ciencias Informáticas

Facultad 9

Diseño del Subsistema de edición de audio para el proyecto Captura y Catalogación de Medias

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Autor: Annes Cedeño López

Tutor: Ing. Solangel Rodríguez Vázquez

Co-Tutor: Ing. Aneli Valdés Acosta

Ciudad de la Habana, Mayo de 2010

Año 52 de la Revolución

A detailed illustration of an unrolled scroll with a textured, parchment-like surface. The scroll is held open by four wooden rollers, one at each corner. The text is centered on the scroll.

*Lo que puedes hacer, o has soñado que podrías
hacer, debes comenzarlo.*

La osadía lleva en sí, genio, poder y magia.

Goethe.

Dedicatoria

DEDICATORIA

Dedicado a:

Mis padres María y Fidel, por darme la vida. Por brindarme a lo largo de estos cinco años todo su amor, esfuerzo y dedicación. Por ofrecerme su apoyo incondicional y por hacer posible la realización de este gran sueño.

A todos mis amigos, los que siempre estaban cuando más los necesité.

Agradecimientos

AGRADECIMIENTOS

A Dios, por darme la oportunidad de poder llegar a este momento.

A mis padres, por ser mi fuente de inspiración.

A todas las personas que de una forma u otra aportaron un granito de arena en la realización de este trabajo.

A Frank, gracias por tanta paciencia y apoyo en todo este tiempo.

A todos los que me han acompañado a lo largo de la carrera, de todos aprendí algo bueno.

A mis amigas Diarilis, Nuris, Gretter y Yaimara. Gracias por los momentos vividos al comenzar esta gran lucha. Estén donde estén, siempre irán en mi corazón.

A mis amigas Dayamí y Yakelín, gracias por estar en mi vida y por la oportunidad de poderlas conocer.

A Zory, por su ayuda incondicional, sin ti no hubiese sido posible ni siquiera comenzar este trabajo.

A mi tutora Solangel, por su esfuerzo y apoyo ilimitado.

A los muchachos del proyecto.

A todos, MUCHAS GRACIAS.

RESUMEN

La edición de audio siempre ha sido importante en aras de lograr que los archivos de audio puedan ser manipulados y obtenidos del modo deseado, garantizando de esta manera una mayor calidad en los ficheros resultantes.

El proyecto Captura y Catalogación de Medias perteneciente al Departamento de Señales Digitales de la Universidad de las Ciencias Informáticas no cuenta con un producto integrado que permita la edición de audio; por tanto es una necesidad diseñar un software con estas posibilidades dada la cantidad de ficheros de audio que necesitan ser tratados y manipulados, garantizando de esta manera la máxima calidad auditiva.

El presente trabajo consiste en la realización del Análisis y Diseño de un subsistema que permita realizar ediciones de audio a diversos materiales y de esta manera incluirlos en el Departamento de Señales Digitales, considerando que es un producto de alta demanda en el mercado actual.

Para la elaboración de este subsistema se ha utilizado el lenguaje Java con la metodología Proceso Unificado de Desarrollo de Software (RUP). El subsistema de edición de audio usa las tecnologías libres siguiendo las actuales tendencias en el mundo.

PALABRAS CLAVES

Edición, audio, ficheros.

ABSTRACT

The audio editing has been always important for the sake of making audio files can be handled and obtained the desired mode, thereby ensuring a higher quality of the resulting files.

The Capture and Cataloging of Medias project of Digital Signals Department belonging to the University of Informatics Sciences does not have an integrated product that allows the audio editing and therefore is a need to design a software with these options, due to the amount of audio files witch need to be addressed and manipulated, assuring thus the maximum sound quality.

The present work consists in carrying out the analysis and design of a subsystem that allows the audio editing to various materials and in this way include them in the Digital Signals Department, considering them a high demand product in the market today.

To develop this subsystem a Java language was used jointly with the Unified Process of Software Development Methodology (RUP). The subsystem of audio editing uses the free technologies following the world current tendencies.

KEY WORDS

Edition, audio, files

Índice de Contenidos

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	7
1.1 INTRODUCCIÓN.....	7
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	7
1.3 SITUACIÓN PROBLEMÁTICA.	8
1.4 EL SONIDO DIGITAL.....	9
1.5 FORMATOS DE AUDIO.....	12
1.6 CÓDEC DE AUDIO.....	14
1.7 LA EDICIÓN DE AUDIO.....	18
1.8 LOS PROCESOS DE EDICIÓN DE AUDIO EN EL PROYECTO CCM.....	20
1.9 SOLUCIONES DE EDICIÓN DE AUDIO EXISTENTES.....	21
1.9.1 <i>Adobe Audition</i>	21
1.9.2 <i>Audacity</i>	22
1.9.3 <i>Wavosaur</i>	23
1.9.4 <i>Power Sound Editor</i>	24
1.10 CONCLUSIONES PARCIALES.....	24
CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES	25
2.1 INTRODUCCIÓN.....	25
2.2 LENGUAJE DE PROGRAMACIÓN.....	25
2.2.1 <i>Lenguaje Java</i>	26
2.3 HERRAMIENTAS DE APOYO.....	27
2.3.1 <i>Java Sound API</i>	28
2.3.2 <i>Java Media Framework (JMF)</i>	28
2.3.3 <i>JSyn (Java síntesis)</i>	29
2.3.4 <i>Ffmpeg</i>	30
2.4 SELECCIÓN DE LAS MÁS IDÓNEAS.....	30

Índice de Contenidos

2.5	ENTORNO DE DESARROLLO INTEGRADO.....	31
2.5.1	<i>NetBeans</i>	31
2.6	METODOLOGÍA DE DESARROLLO DE SOFTWARE.	32
2.6.1	<i>Rational Unified Process (RUP)</i>	32
2.7	LENGUAJE DE MODELADO: UML	37
2.8	HERRAMIENTAS CASE.....	37
2.8.1	<i>Visual Paradigm</i>	38
2.9	CONCLUSIONES PARCIALES.	40
CAPÍTULO 3.	ANÁLISIS DEL SISTEMA.....	41
3.1	INTRODUCCIÓN.....	41
3.2	MODELO DE DOMINIO.....	41
3.2.1	<i>Conceptos principales del modelo de dominio</i>	41
3.2.2	<i>Diagrama de Clases del Modelo de Dominio</i>	42
3.2.3	<i>Descripción del Modelo de Dominio</i>	42
3.2.4	<i>Reglas del Negocio</i>	43
3.3	REQUERIMIENTOS.....	43
3.3.1	<i>Requerimientos Funcionales</i>	44
3.3.2	<i>Requerimientos no funcionales</i>	45
3.4	DESCRIPCIÓN DEL SISTEMA PROPUESTO.	46
3.5	DESCRIPCIÓN DEL SISTEMA. MODELOS DE CASOS DE USO DEL SISTEMA.	46
3.5.1	<i>Determinación y justificación de los actores del Sistema</i>	46
3.5.2	<i>Casos de Uso del Sistema</i>	47
3.5.3	<i>Diagrama de Casos de Uso del Sistema</i>	47
3.6	DESCRIPCIÓN DE LOS CASOS DE USO.....	48
3.7	CONCLUSIONES PARCIALES.	57
CAPÍTULO 4.	CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	58
4.1	INTRODUCCIÓN.....	58

Índice de Contenidos

4.2	PATRONES.....	58
4.2.1	<i>Patrones de Diseño.....</i>	58
4.2.2	<i>Patrones de Arquitectura.....</i>	59
4.2.3	<i>Arquitectura en Capas.....</i>	59
4.3	DIAGRAMAS DE CLASES DEL DISEÑO.....	60
4.4	PRINCIPIOS DE DISEÑO.....	64
4.4.1	<i>Estándares de interfaz de aplicación.....</i>	64
4.5	DIAGRAMAS DE INTERACCIÓN.....	64
4.5.1	<i>Diagramas de Secuencia.....</i>	65
4.5.2	<i>Diagramas de Colaboración.....</i>	65
4.6	CONCLUSIONES PARCIALES.....	65
	CONCLUSIONES GENERALES.....	67
	RECOMENDACIONES.....	68
	REFERENCIAS BIBLIOGRÁFICAS.....	69

Índice de Figuras

ÍNDICE DE FIGURAS

Figura 1: Procesos desarrollados en el proyecto CCM.....	9
Figura 2: Clasificaciones del Audio Digital.....	11
Figura 3: Proceso de edición de audio.....	17
Figura 4: Fases de RUP.....	34
Figura 5: Flujos de trabajo de RUP.....	35
Figura 6: Diagrama de clases del modelo del dominio.....	42
Figura 7: Modelo de Casos de Uso del Sistema.....	48
Figura 8: Capa de Presentación, Diseño de Clases.....	61
Figura 9: Capa Lógica del Negocio, Diseño de Clases.....	62
Figura 10: Capa Acceso a Datos, Diseño de Clases.....	62
Figura 11: Interacción General de las Clases.....	63

Índice de Tablas

ÍNDICE DE TABLAS

Tabla 1: Actor del sistema y su funcionalidad.	47
Tabla 2: Prioridad de los Casos de Uso.	47
Tabla 3: Descripción del caso de uso: Segmentar fichero de audio.	50
Tabla 4: Descripción del caso de uso: Establecer volumen.....	51
Tabla 5: Descripción del caso de uso: Normalizar audio.	52
Tabla 6: Descripción del caso de uso: Aplicar efecto.	54
Tabla 7: Descripción textual del caso de uso Importar archivo de audio.	55
Tabla 8: Descripción del caso de uso Permitir procesamiento multipista.....	56

INTRODUCCIÓN

Desde tiempos remotos el hombre ha sentido la necesidad de poderse comunicar y expresar; para lograrlo comenzó a valerse del **sonido** como medio de comunicación.

El sonido no es más que una onda que se propaga solamente si se encuentra en presencia de un medio que sirva de soporte a la perturbación. (1)

El descubrimiento de que el sonido de la voz podía ser amplificado no fue sino el primer paso de una larga carrera de investigación e invenciones que directamente condujo al sueño de que dicha voz pudiese ser grabada, reproducida y conservada por lo que no es hasta el 9 de abril de 1860 que el francés Edouard-Leon Scott de Martinville logró grabar el primer sonido, una canción popular de entonces, llamada Au clair de la lune (En el claro de la luna, en francés). (2)

La invención de Scott de Martinville no permitía reproducir el sonido grabado, y fue entonces que Thomas Alva Edison (1847-1931) revolucionó el entretenimiento hogareño, con la creación del Fonógrafo. Desde esta época el audio estuvo a la vanguardia de la tecnología. Este último dispositivo permitía la reproducción de sonido, para esto utilizaba cilindros donde se grababa el mismo y posteriormente se reproducía. Lamentablemente, las características intrínsecas de la calidad de sonido de la invención de Edison eran relativamente pobres y las primeras láminas de metal usadas en la grabación tenían una vida útil limitada. (3)

Nuevos artefactos fueron creados con el fin de mejorar la calidad del sonido y el tiempo de explotación. Emilio Berliner registraba y patentaba en 1888 una máquina parlante que también grababa y reproducía el sonido, pero con la diferencia de que no usaba el cilindro como soporte de la grabación sino, un disco plano y además la impresión se efectuaba en el surco por amplitud lateral y no como en el cilindro que se hacía de forma vertical. A esa máquina parlante Berliner la bautizó con el nombre de *Gramófono*. Las ventajas de este invento fueron evidentes comparadas con las del *Fonógrafo* y su cilindro. Mientras que con una sola toma, el *Gramófono* podía prensar miles de copias a partir de esa única matriz, el *Fonógrafo* en cambio, necesitaba para producir 500 cilindros, ejecutar 25 veces la misma obra y grabarlos directamente de manera simultánea en 20 fonógrafos. (3)

Introducción

A pesar de la amplia aceptación que tuvieron en el mercado los dispositivos edisonianos y berlinerianos, la calidad del sonido seguía siendo imperfecta, con altos niveles de ruido y muy bajos niveles de volumen en los amplificadores acústicos que se estaban utilizando.

Con el avance en el desarrollo del alto rendimiento de soluciones de audio llegaron tres grandes inventos:

El disco de vinilo LP 33rpm registrado en Colombia en la década de 1940.

La válvula y luego la base de transistores electrónica.

La grabación estéreo. (4)

Una de las características fundamentales del hombre siempre ha sido la perseverancia en buscar continuamente nuevas metas. Aunque en los años del 1857-1940 se lograron alcanzar ciertos avances en lo que respecta a la grabación, reproducción y la conservación del sonido, estas no fueron suficientes para satisfacer las necesidades que surgieron en el día a día.

El avance vertiginoso de la Informática y las Comunicaciones se ha mantenido estrechamente ligado con el desarrollo de los medios audiovisuales, en los cuales el sonido es un elemento imprescindible; surge entonces, como parte de esta revolución, el sonido digital. Se puede definir el **sonido digital** como el resultado de la codificación digital (representada por ceros y unos) de una onda sonora o de una señal analógica. (5)

Con los adelantos tecnológicos se ha logrado contar con sistemas de comunicación más eficaces, de mayor alcance, más potentes y prácticos, en este caso, el **sonido digital** se ha convertido en un medio de gran utilidad y variedad; aplicado en una gran diversidad de campos, como son: la informática, la medicina, la industria, la física, la música, incluso en el ámbito militar.

Actualmente el hombre necesita modificar el sonido para diversos fines, como son, arreglar una grabación con ruido, segmentar determinado fichero de audio, disminuir el volumen de una grabación, todos estos procesos se incluyen en lo que se conoce como edición. En informática, el término editar no es más que la acción de crear o modificar un archivo informático, visualizando su resultado en pantalla, escuchándolo en el caso del sonido.

Conjuntamente con el auge y progreso de la informática ha venido el desarrollo de las aplicaciones para la edición de sonido.

Introducción

¿Qué es un editor de sonido?

Es aquella aplicación informática usada para manipular audio digital. (6)

La edición de audio es una funcionalidad muy común entre los software de trabajo con archivos multimedia, ejemplo de ellos son: Adobe Audition, Audacity, Power Sound Editor, Audio Traverso, Jokosher, Wavosaur, Wavepad sound editor, Ardour.

Muchas de estas aplicaciones están desarrolladas sobre software privativos y resultan costosos por lo que su uso depende de las licencias y precios de los mismos.

Cuba, en aras de obtener un mayor desarrollo de las tecnologías informáticas, propone diversas investigaciones relacionadas al desarrollo de herramientas vinculadas al uso de las tecnologías digitales. Específicamente en la Universidad de las Ciencias Informáticas (UCI), se organizaron un conjunto de proyectos orientados al desarrollo de las tecnologías digitales. En este marco se realiza el proyecto de Captura y Catalogación de Media (CCM) perteneciente al Departamento de Señales Digitales (DSD) de la Facultad 9, cuyo objetivo central es la gestión de cualquier tipo de funcionalidad que se pudiera realizar con las medias de audio y video.

Este proyecto se estructura por cuatro módulos fundamentales los cuales se mencionan a continuación: Captura y Transcripción de Audio, Administración y Configuración, Captura e Indexación de Video y Catalogación de Medias.

En el proyecto Captura y Catalogación de Medias se contará con un servidor de medias, destinado a almacenar los ficheros de audio y video con los que se trabajará en la entidad que lo implante. El origen de los ficheros de audio puede ser diverso: entrevistas realizadas en exteriores, grabaciones dentro de la propia entidad, pertenecientes a reuniones, conferencias, entre otras actividades y se agregará en un futuro la grabación de emisiones radiales. El objetivo fundamental del almacenamiento de los ficheros de audio es su posterior transcripción, proceso que consiste en llevar a texto el contenido de la grabación auditiva.

La transcripción se realiza dentro del Subsistema de Captura y Transcripción de Audio, el cual provee una interfaz con editor de texto y reproductor de audio integrados, que le permite a los transcriptores teclear mientras escuchan las grabaciones. Los ficheros de audio, además son catalogados de acuerdo a diferentes parámetros e intereses de la entidad en el subsistema de Catalogación de Medias. La catalogación la realizan un grupo de personas que necesitan escuchar el contenido de la grabación y

Introducción

catalogar según el mismo. Las grabaciones realizadas mediante la aplicación generalmente no poseen distorsiones auditivas, pero no se puede asegurar que una entrevista u otro material procedente de otro origen no contengan desperfectos como ruido, bajo volumen, entre otros, que atenten contra la buena audición y por lo tanto contra la transcripción y la catalogación.

Actualmente el subsistema de Catalogación de medias, permite la edición de video, pero no la de audio, por lo que el funcionamiento del sistema es vulnerable a los problemas que puedan presentar las grabaciones de audio. Un sistema de edición de audio resolvería la problemática descrita anteriormente y constituiría un producto de valor agregado al subsistema de Captura y Catalogación de Medias.

De lo expuesto anteriormente se ha definido el siguiente **problema científico**: Necesidad de un subsistema que permita la edición de audio en el proyecto de Captura y Catalogación de Medias del Departamento de Señales Digitales de la facultad 9.

Para lograr resolver el problema descrito anteriormente se hace necesario diseñar un subsistema que permita la edición de audio en el proyecto de CCM del Departamento de Señales Digitales de la facultad 9, lo cual constituye el **objetivo general** de esta investigación.

Siendo el **objeto de estudio** de la investigación los procesos de Edición de Audio.

El **campo de acción** estará centrado en el Subsistema de Edición de Audio en el proyecto Captura y Catalogación de Medias.

La investigación tiene la siguiente **idea a defender**: El diseño propuesto permitirá la futura implementación del subsistema que garantizará la edición de audio en el proyecto CCM.

Para dar cumplimiento al objetivo general se acometen las siguientes **tareas de investigación**:

1. Caracterizar los procesos de edición de audio.
2. Caracterizar los procesos concebidos en el proyecto Captura y Catalogación de Medias relacionados con la edición de audio.
3. Describir el estado del arte de las aplicaciones para la edición de audio.
4. Seleccionar las tecnologías y herramientas a utilizar para el desarrollo del subsistema propuesto.
5. Realizar análisis y diseño del subsistema de edición de audio.

Introducción

Cumpliendo de forma correcta las tareas planteadas se esperan obtener los siguientes **resultados**:

- Documentación que recoja la investigación previa a los temas involucrados en la solución del subsistema de edición de audio.
- Documentación UML de los artefactos resultantes del análisis y diseño del sistema de gestión de guía electrónica de programación.

Para la realización de las tareas planteadas se emplearon los siguientes **métodos científicos** de la investigación:

Métodos Teóricos

Analítico – Sintético: Se utilizó con el objetivo de analizar los diversos documentos relacionados con el proceso de edición de audio y para lograr un entendimiento más profundo del tema. Además, mediante este método se analizaron los principales componentes y funcionalidades de soluciones similares que sirven de antecedentes a este trabajo.

Histórico Lógico: Se empleó para analizar el surgimiento de los procesos de edición de audio y su evolución a lo largo de la historia.

Modelación: Se empleó para mostrar los diferentes diagramas y componentes que se construyen como resultado del proceso de Ingeniería de Software.

Métodos empíricos

Entrevista: Se realizaron entrevistas con el objetivo de comprender el funcionamiento del proyecto CCM y para obtener información acerca de los requerimientos que debe cumplir el software.

Introducción

El Trabajo de Diploma ha sido estructurado de la siguiente manera:

Capítulo 1: Contempla la fundamentación teórica de la presente investigación, se exponen los principales conceptos que contribuyen al mejor entendimiento del problema en cuestión, y se realiza un análisis del estado del arte referente a los sistemas de edición de audio.

Capítulo 2: Tendencias y tecnologías actuales, se describen las metodologías de desarrollo, lenguajes de programación y tecnologías actuales para el desarrollo de la solución presentada, justificando además la selección y uso de estas.

Capítulo 3: En este capítulo se presenta la solución al problema planteado a partir de la descripción del modelo de dominio y la identificación y descripción de los casos de uso del sistema. Se describen además los conceptos fundamentales que se manejan en el sistema y se proponen los requisitos funcionales y no funcionales con los cuales el sistema debe cumplir.

Capítulo 4: Se expone de forma completa la solución propuesta. Se efectúa el diseño completo de la aplicación, tratando aspectos fundamentales como la arquitectura y los diagramas de interacción.

Capítulo 1 : Fundamentación Teórica

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN.

En el presente capítulo se abordan conceptos asociados al dominio del problema que permiten tener una mejor comprensión del mismo. Se brinda información sobre los procesos de edición de audio en el mundo actual y se explica con profundidad aspectos relacionados con la situación problemática existente.

1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.

La edición de audio consta de todo un proceso donde el sonido analógico es digitalizado, codificado y posteriormente almacenado para su edición. A este conjunto de pasos están asociados varios conceptos que son necesarios exponer para lograr un mayor entendimiento en el marco del problema.

Audio: técnica relacionada con la reproducción, grabación y transmisión del sonido. (7)

Audio Analógico: el sonido viaja en forma de ondas y a través de medios elásticos como el aire. Por la presión provocada por los objetos que vibran y desplazan partículas de aire se crea la onda. (8) La misma puede ser representada por una función matemática continua en la que varían la amplitud y el periodo, lo que significa cambios en el sonido que se escucha. La manera en la que se presenta el sonido en la naturaleza es conocida como analógica.

Audio Digital: se define como el resultado de la codificación digital (representada por ceros y unos) de una onda sonora o de una señal analógica. (5)

Sistema Digital: combinación de dispositivos diseñado para manipular información lógica o cantidades físicas que estén representadas de forma digital. Algunos de los sistemas digitales más familiares incluyen computadoras digitales, equipos de audio y video digital, así como sistemas telefónicos.

Sistemas Digitales de Edición de Audio: aplicaciones informáticas usadas para editar audio, es decir, manipular audio digital. (6)

Edición de audio: son aquellas acciones de crear o modificar un archivo informático, visualizando su resultado en pantalla, escuchándolo en caso del sonido.

Capítulo 1 : Fundamentación Teórica

Pista de audio: objeto de audio que está pensado como una presentación, total o parcial. Una pista de audio puede, pero no está obligado a corresponder a un único canal de audio (izquierdo o canal de audio derecho). (9)

1.3 SITUACIÓN PROBLEMÁTICA.

En el Departamento de Señales Digitales del Centro Geoinformática y Señales Digitales se trabaja con volúmenes considerables de archivos audiovisuales, por lo que el almacenamiento seguro y gestionable de manera eficiente de estos archivos es una preocupación constante de todo el personal perteneciente al centro. La realización de forma manual de actividades como la grabación de archivos audiovisuales, la captura de señales de video y audio, la digitalización de archivos multimedia analógicos, la transcripción de ficheros de audio y la catalogación y gestión de materiales audiovisuales son actividades que provocan una mayor cantidad de esfuerzo, tiempo e ineficiencia considerable. Por estas razones se crea el proyecto de Captura y Catalogación de Medias el cual permite la *captura de señales de audio* según planificaciones definidas, además de permitir el *almacenamiento de las medias* por lo que se pueden volver a transmitir, si se está interesado. Una vez almacenadas las medias en un servidor se les aplica un proceso de *catalogación* según su clasificación o criterio estándar por parte de la persona encargada de la tarea. Además se realiza el proceso de *transcripción y consultas de audio*. A la hora de catalogar es necesario que el audio se encuentre en perfecto estado, resultándole de este modo a la persona encargada de la tarea escuchar el contenido de la grabación sin obstáculo alguno y poder catalogar de una forma eficiente. Para lograr la buena audición de estas grabaciones es necesario contar con una aplicación que permita la *edición de audio*, mediante la cual se pueden eliminar ruidos que pudieron ser generados a la hora de la captura o la grabación del archivo, segmentar las porciones del audio más significativas y de interés a transmitir posteriormente, elevar o disminuir el volumen de la grabación, entre otros procesos que proporcionen una buena audición y con el objetivo de lograr una mayor calidad en las medias de audio.

Capítulo 1 : Fundamentación Teórica

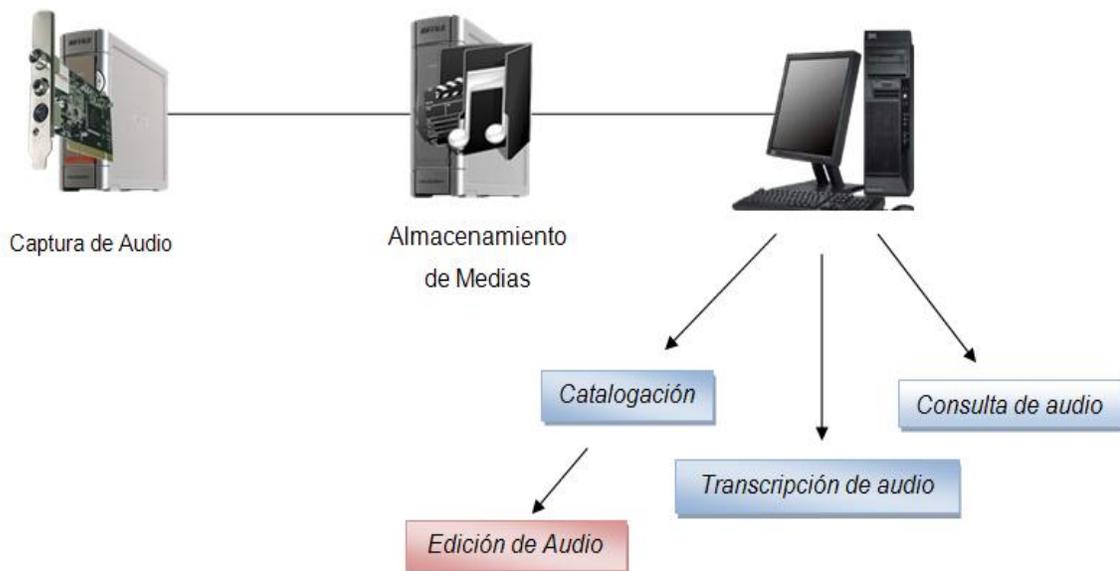


Figura 1: Procesos desarrollados en el proyecto CCM

1.4 EL SONIDO DIGITAL.

Para comprender las características del sonido digital, se hará una pequeña introducción de las características del sonido y sus conceptos más básicos.

El sonido son vibraciones de las partículas de aire que se propagan a través del mismo, gracias a estas partículas que se mueven, las vibraciones llegan a nuestros oídos. La audición en los seres humanos, ocurre siempre que una vibración tenga una frecuencia comprendida entre unos 15 y 20.000 hercios, y su intensidad sea la suficiente para llegar al oído interno. Cuando las vibraciones pasan estos márgenes se habla de ultrasonidos y no son perceptibles al ser humano.

La palabra digital indica la presencia de procesos numéricos para concretar un hecho (imagen, sonido, etc.), los sistemas de audio digital tienen circuitos eléctricos para guardar el registro de la música, en pocas palabras lo que hacen estos circuitos es grabar una larga cadena de números (digitalización o muestreo) con un dispositivo llamado convertor análogo digital (ADC), que se encarga de monitorear la evolución de la onda y asignarle a cada momento un valor numérico, luego ese valor numérico es decodificado por un convertor llamado digital-análogo (DAC).

El proceso de conversión analógica-digital y viceversa se explica con más profundidad a continuación.

Capítulo 1 : Fundamentación Teórica

Primeramente se captura el sonido haciendo uso del micrófono. Cuando las ondas de sonido llegan al micrófono, el movimiento mecánico se traduce en una señal eléctrica.

Esta señal se denomina señal analógica porque es una señal continua en el tiempo y análoga al sonido original.

Conversión analógica-digital (adc): Dada una señal analógica, se van tomando valores discretos de su amplitud a intervalos de tiempo pequeños, evidentemente será más fiable la reproducción cuantas más muestras por segundo se tomen. A estos valores obtenidos se les asigna un valor digital que la computadora puede entender y procesar como se requiera. A cada muestra obtenida se le asigna un equivalente binario ya que es en este sistema en el que trabajan las computadoras, su unidad de información es el bit.

Conversión digital-analógica (dac): El proceso inverso es mucho menos complejo ya que solo se trata de ir poniendo los valores de las muestras en el mismo orden que fueron tomados y unos filtros electrónicos se encargan de convertir esa señal resultante de valores discretos en una señal analógica.

Calidad Del Sonido Digital

Frecuencia o tasa de muestreo: el número de veces por segundo que una señal analógica se mide y se convierte en un número binario, con el fin de convertir la señal analógica a una señal digital.

Bit de profundidad: el bit de profundidad determina el nivel de detalle que se puede obtener de cada frecuencia obtenida durante el muestreo. Cuanto mayor sea la profundidad de bits, mayor será la gama dinámica de los niveles de amplitud que se podrán capturar. La gama dinámica es obviamente importante, da el nivel de rango dinámico que los oídos pueden percibir. Pero su significado real es que, cuando el número de posibles niveles dinámico está limitado por el uso de un convertidor con una menor profundidad de bits, la medición de la señal analógica se convierte en inexacta. Esta inexactitud es percibida como ruido. (10)

Códec: Codificador-decodificador. Un dispositivo para convertir el vídeo analógico y las señales de audio en un formato digital para su transmisión, así como para convertir las señales digitales recibidas en analógico. Un sistema matemático para la compresión (codificación) y descompresión (decodificación) un archivo de audio o vídeo. Los códecs pueden ser de hardware o software, o ambos. Los códecs de

Capítulo 1 : Fundamentación Teórica

hardware suelen ser más eficientes, pero el problema está en que no todos los usuarios tendrán el hardware necesario para reproducir un archivo.

Bitrate: Relativo a la velocidad de un dispositivo, velocidad con que los dígitos binarios se pueden transferir. Puede medirse en bits por segundo (bit/s), cantidad de información transmitida en un segundo (11). El bitrate puede ser constante (CBR) para todo el archivo o variar (VBR) en dependencia de la cantidad de información a guardar por segundo. Con VBR, la tasa de bits de codificación aumenta las secciones de captura de datos complejos y, disminuye la compresión de las secciones menos complejas (12).

¿Cómo reducir el tamaño de los archivos de audio manteniendo la calidad de la grabación?

Con el objetivo de reducir el tamaño de los archivos de audio y logrando una alta calidad en los mismos se han creado diversos códec de audio y se han experimentado distintas combinaciones de frecuencias de muestreo con diferente bitrate, dando paso de esta manera a la creación de diferentes formatos de audio digital.

Los formatos de audio se pueden agrupar en dos tipos: analógico y digital. El audio digital por su parte se puede clasificar en audio digital sin compresión y audio digital con compresión, y este último en dos nuevas clasificaciones, audio digital con compresión sin pérdida y audio digital con compresión con pérdida.

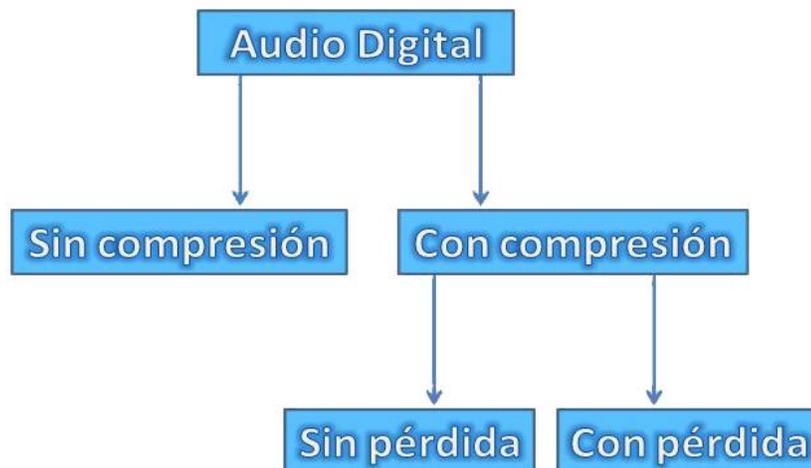


Figura 2: Clasificaciones del Audio Digital

Capítulo 1 : Fundamentación Teórica

Para la confección de un editor de audio es necesario tener conocimiento sobre las características de los distintos formatos con que puede operar el mismo. De esta forma se podrá identificar el formato de audio con el cual el editor podrá realizar las modificaciones y ajustes necesarios para obtener el archivo de la manera deseada. Se detallan a continuación las principales características de los formatos de audio.

1.5 FORMATOS DE AUDIO.

Un formato de audio es un contenedor multimedia que guarda una grabación de audio (música, discurso, etc.). Lo que hace a un archivo distinto del otro son sus propiedades; cómo se almacenan los datos, sus capacidades de reproducción, y cómo puede utilizarse el archivo en un sistema de administración de archivos. La diferencia radica en que la calidad de la música digitalizada diverge según el formato, la calidad y la compresión ejercida sobre unos archivos de audio. Existen diferentes tipos de formato según la compresión del audio:

Formato WAV (compresión sin pérdida).

Su nombre original es formato de audio con forma de onda (WAVEform audio format en inglés) y es un formato de audio digital sin comprimir, admite archivos mono-canal y estéreos a diferentes velocidades de muestreo. La extensión contenedora que lo caracteriza es wav.

Es un formato óptimo para ser utilizado por profesionales de esta rama, ya que no tiene ningún tipo de compresión, además es ideal también para el almacenamiento. Otra de sus ventajas es que por sus características es soportado por casi todos los códec de sonido y por todos los sistemas de MPEG, es reconocido fácilmente por utilizarse principalmente con formatos sin compresión. Para que el formato tenga calidad suficiente de CD es necesario que la frecuencia de muestreo sea 44100 Hz como mínimo y a 16 bits, por lo que cada minuto de grabación ocuparía aproximadamente 10 Mb de disco duro lo cual es bastante grande, esa es la desventaja fundamental de este formato, y la otra desventaja es que solo puede grabar hasta 6,6 horas y se debe a que en la cabecera del fichero se indica la longitud del mismo con un número entero de 32 bit, lo que limita el tamaño del fichero a 4 GB.

Capítulo 1 : Fundamentación Teórica

Formato MP3 o MPEG-1 Layer 3 (con compresión con pérdida)

La creación de MP3 no se atribuye a una sola persona, es la unión de muchos conceptos y técnicas de codificación de distintas empresas en la etapa del 1979 a 1992 que quedó oficialmente registrado por la Moving Picture Experts Group (MPEG) y registrado en la Organización Internacional de Normalización (ISO), aunque no se puede dejar de mencionar los importantes aportes de Karlheinz Brandenburg, el Instituto Fraunhofer, Universidad de Hannover y en menor medida las empresas AT & T y Thomson.

MPEG en su sección de audio, creó una familia genérica de tres formatos de códec (Layer-1, -2, -3). Se especifica que el Nivel 3 (MP3) es un códec más eficiente, lo que dio lugar a su adopción generalizada como una forma de almacenar música en las relativamente pequeñas unidades de disco duro de la era de la PC y transferir archivos de música a través de Internet. Utilizando la codificación de MP3, el bitrate de CD se puede reducir por un factor de 11:1, obteniéndose 12 Mbit/s para 96 kbit/s y manteniendo una buena semejanza del sonido original. El mp3 estándar es de 44,1 KHz y un bitrate de 128 kbps por la relación de calidad/tamaño. (13)

Formato Ogg Vorbis (compresión con pérdida).

Ogg Vorbis es un formato de compresión de audio relativamente nuevo. Es más o menos comparable a la de otros formatos utilizados para almacenar y reproducir música digital, como MP3. Es diferente de los demás formatos porque es completamente libre, de código abierto y no patentado. Ogg es el formato de la *Xiph.org Foundation* contenedor de audio y metadatos mientras que Vorbis es el sistema de compresión de audio diseñado para obtener ogg. Fue publicado oficialmente en febrero de 2003. Con un bitrate y una frecuencia de muestreo igual, un archivo mp3 ocupa el mismo espacio aproximadamente que un archivo ogg. (14)

Formato AAC (Advanced Audio Coding) (compresión con pérdida).

AAC fue desarrollado por el grupo que incluye a MPEG Dolby, Fraunhofer (FhG), AT & T, Sony y Nokia, las empresas que también han participado en el desarrollo de códecs de audio tales como MP3 y AC3 (también conocido como Dolby Digital). El códec de AAC de QuickTime se basa en las nuevas técnicas de procesamiento de señales de las tecnologías de Dolby Laboratories. Debido a su increíble calidad en una amplia gama de velocidades de transmisión de datos y la ratificación como un estándar en la industria,

Capítulo 1 : Fundamentación Teórica

AAC audio está ganando amplia adopción en el mercado. Por ejemplo, todas las ventas de música en iTunes Music Store utiliza AAC, para reproducir en el escritorio o en el iPod.

Entre las ventajas que presenta el formato AAC se encuentran:

- Soporte para audio multicanal, ofreciendo hasta 48 canales de alta frecuencia.
- Mayor resolución de audio, produciendo tasas de muestreo hasta 96 kHz.
- Mejora de la eficiencia de decodificación, que requieren menos potencia de procesamiento para decodificar. (15)

1.6 CÓDEC DE AUDIO.

Asociado a cada uno de los formatos se encuentran los códec de audio. Estos cumplen fundamentalmente la función de reducir la cantidad de datos digitales necesarios para reproducir una señal auditiva. Lo que comúnmente se denomina compresión de datos, pero aplicado a un fin muy concreto. Por ello, existen fundamentalmente dos aplicaciones de los códec de audio:

Almacenamiento: útil para reproductores multimedia que pueden reproducir sonido almacenado, por ejemplo, en un disco duro, CD-ROM o tarjeta de memoria.

Transmisión: útil para implementar redes de videoconferencia y Telefonía IP.

Hoy en día el mundo de los CÓDECs es muy amplio, lo que exige clasificarlos de algún modo. Una primera clasificación bastante general permite distinguir entre:

CÓDECs con pérdida: se trata de CÓDECs que provocan una pérdida de información que no podrá ser recuperada en favor de un menor tamaño (más compresión) del archivo. El método habitual de compresión suele basarse en la eliminación frecuencias inaudibles para el ser humano, de tal forma que la calidad se reduzca lo menos posible a los oídos del usuario.

CÓDECs sin pérdida: se trata de CÓDECs, que en el proceso de decodificación son capaces de recuperar toda la información tal y como era originalmente.

Se exponen a continuación los códec de audio más usados en la actualidad.

Capítulo 1 : Fundamentación Teórica

MPEG-1 Layer 3 Códec.

MPEG-1 Layer 3 Códec fue creado por Fraunhofer Corporation para el sistema operativo Windows y tiene licencia *Freeware*. Fraunhofer-Gesellschaft (FhG) es una empresa alemana que ayudó a la tecnología pionera de MP3. El instituto es un centro de investigación en curso de codificación de audio en bitrates bajos. Básicamente se está tratando de lograr la mejor calidad de sonido con archivos de menor tamaño. Desde 1998, el instituto Fraunhofer ha desarrollado una serie de codificadores que han combinado sus avances tecnológicos. (16)

MP3: acrónimo de MPEG-1 Audio Layer 3, es, sin duda, el CÓDEC de audio por excelencia. Aunque el MPEG comenzó a desarrollarlo a mediados de los 80 su uso no se extendió hasta mediado-final de los 90. El Mp3 ofrece gran calidad de audio pero con un tamaño hasta 15 veces menor (depende del grado de compresión) que el archivo original sin compresión. Fue precisamente esta característica la que convirtió al Mp3 en el formato musical más demandado a través de Internet desde finales de los 90 y, actualmente, uno de los formatos más usado en lo referente a la fabricación de dispositivos musicales (discman capaces de reproducir Mp3, aparatos de Mp3 que permiten reproducir archivos extraídos mediante un cable USB de cualquier computador personal, etc.). Por desgracia, hoy en día recibe más críticas que elogios, ya que su éxito en Internet trajo consigo un aumento considerable de la piratería.

LAME MP3 Códec

LAME MP3 Códec es un codificador de alta calidad para MPEG-1 Layer 3 licenciado bajo LGPL (Licencia Pública General. (17) Las siglas LAME significan: LAME no es un codificador MP3 (**LAME Ain't an Mp3 Encoder**). Hoy en día, LAME es considerado el mejor codificador MP3 a bitrates medio-alto y en VBR, en su mayoría gracias a la dedicada labor de sus desarrolladores de código abierto y el modelo de licencia que permite el proyecto para aprovechar los recursos de ingeniería de todo el mundo. Tanto la calidad del fichero obtenido como la velocidad de compresión se siguen mejorando, probablemente es el único codificador MP3 que sigue siendo desarrollado activamente. (35) LAME compila en los sistemas operativos Windows, DOS, GNU / Linux, MacOS X, BSD, Solaris, HP-UX, Tru64 Unix, AIX, Irix, NeXTstep, SCO Unix, UnixWare, Ultrix, Plan 9, OpenVMS, Classic MacOS, BeOS, QNX, Riscos, AmigaOS, OS / 2, SkyOS, FreeMiNT (Atari), y algunos otros, brindando un audio de alta calidad.

Capítulo 1 : Fundamentación Teórica

Las operaciones con LAME pueden ser controladas desde una ventana de consola a través de comandos o su código puede ser obtenido y ser recompilado para su utilización integrado a una aplicación.

K-Lite Códec Pack

K-Lite Códec Pack es un paquete de códec para audio y video creado por KL Software y liberado bajo licencia *Freeware*. Integra más de 20 códec que permiten reproducir casi cualquier tipo de archivo de audio y video, hasta formatos DVD. (18) Fue creado para ser usado en Windows.

PCM

PCM (Pulse Code Modulation. Modulación por codificación de pulsos). PCM es el método estándar utilizado para codificar audio digitalmente y es el formato de datos descomprimidos básico utilizado en formatos de archivo como WAV y AIFF. (19)

Proceso de edición de audio. Uso de formatos y códecs.

A continuación se muestra un esquema donde se grafica el proceso de edición de audio. Se muestran además los formatos y códecs que serán utilizados en dicho proceso.

Capítulo 1 : Fundamentación Teórica

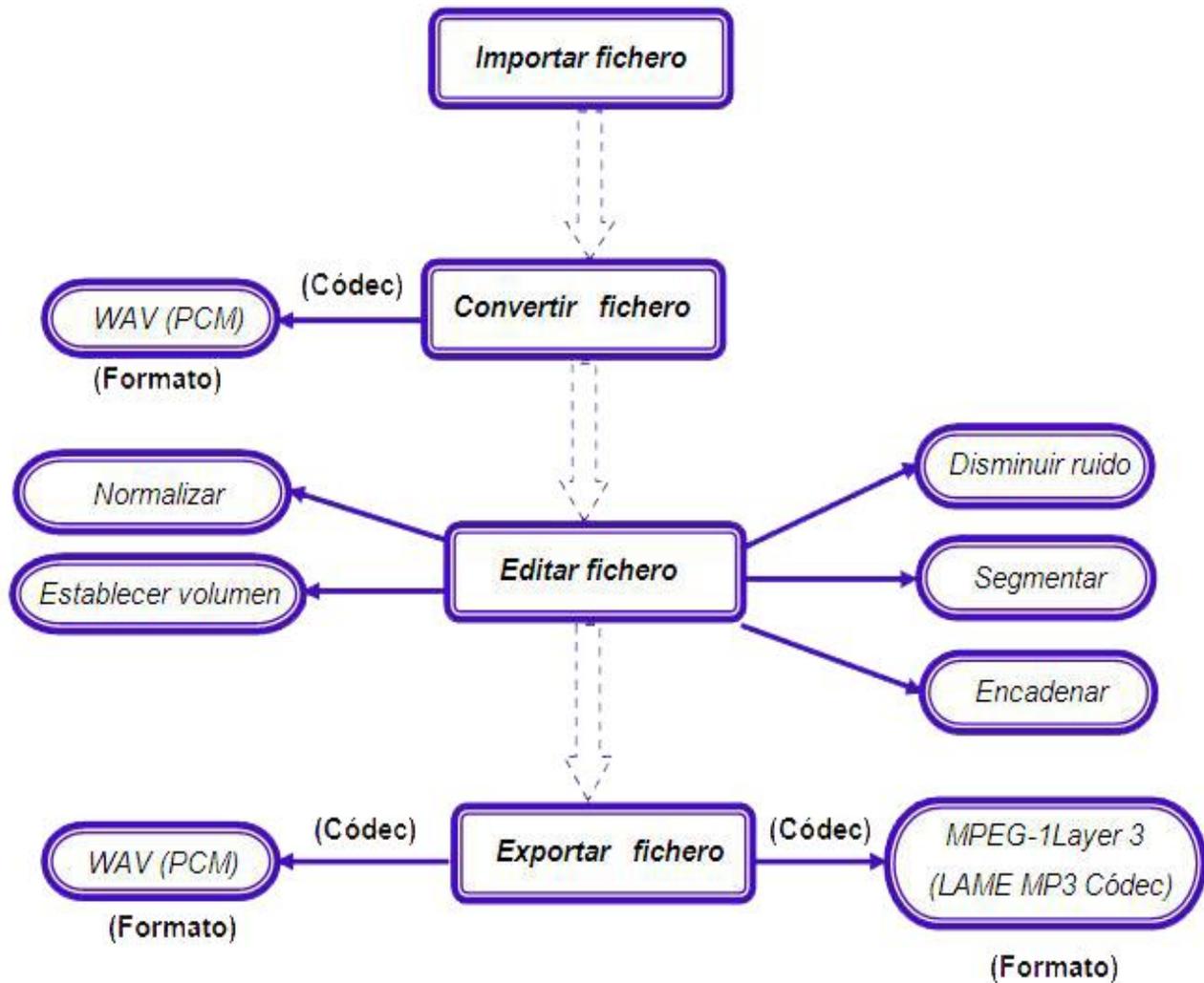


Figura 3: Proceso de edición de audio.

Para el proceso de edición es necesario importar un fichero de audio en la aplicación. Teniendo en cuenta que el proyecto CCM utiliza el formato mp3 para el almacenamiento de los archivos de audio se decide que el sistema convierta los ficheros al formato wav con el objetivo de mantener toda la información y calidad del fichero, esto facilitará su posterior manipulación y edición.

Capítulo 1 : Fundamentación Teórica

Una vez convertido es posible realizar procesos como normalizar, encadenar, segmentar, disminuir ruido y establecer volumen para obtener el audio de la manera deseada. Como paso final se debe exportar el archivo al formato MPEG-1 Layer 3 apoyado en la extensión de audio .mp3, utilizando el LAME MP3 Códec si el audio está listo para el almacenamiento. Si se desea editar en otra ocasión el audio puede ser exportado en formato wav garantizando que no se pierda información.

La mayoría de los codificadores de audio están elaborados para plataformas propietarias y muchos de estos no se encuentran en constante desarrollo de las técnicas de compresión, motivo por el cual son superados por otros compresores de sonidos. Algunos de estos códec están bajo licencias de uso que limitan la posibilidad de usarlos en productos comercializables. Por no presentar ninguno de los problemas expuestos anteriormente y por ser un excelente codificador en cuanto a velocidad de compresión y la alta calidad de los ficheros obtenidos con él para el desarrollo del sistema se propone utilizar LAME como códec al formato MPEG-1 Layer 3. Para la edición del audio se propone el formato WAV pues presenta una compresión sin pérdida, de esta manera el proceso de edición se hace más cómodo y factible, ya que el audio presenta una alta calidad.

1.7 LA EDICIÓN DE AUDIO.

En el mundo actual el término ***digital*** se ha vuelto parte del vocabulario común cotidiano debido a la forma tan impresionante en que las técnicas digitales se han difundido en casi todas las ramas de la vida como: la ciencia de la computación, la automatización, la ciencia y la tecnología médica, la transportación, el entretenimiento, la exploración espacial, etc.

Las conferencias magistrales en las universidades, las vistas judiciales y demás procesos en los tribunales, sesiones parlamentarias, reuniones de gobierno, encuentros de convenciones, comunicaciones oficiales y otras actividades pueden ser grabadas como audio, que al igual que las imágenes y el video se clasifican como contenido multimedia. (20)

Mediante la edición de audio es posible mejorar la calidad de estas grabaciones, eliminar aspectos no primordiales, insertar elementos que sirvan de complemento a la grabación, así como almacenar la información en el formato deseado.

Antiguamente el proceso de edición de audio era lento, tedioso, y a veces inestable. Si en el proceso de edición ocurría algún error inesperado había que deshacer todo de nuevo. Sin embargo, hoy en día ocurre

Capítulo 1 : Fundamentación Teórica

todo lo contrario, el proceso de editar un archivo informático no resulta muy complicado, mediante este proceso se mantiene la estética de la grabación, y por lo tanto aumenta la atención del oyente.

La edición de audio, imagen y vídeo digital son tareas muy gratificantes que permiten dar rienda suelta a la imaginación para expresar ideas o mensajes de un modo ameno y divertido, la misma es utilizada con diversos fines y en beneficio de personas con distintos intereses, por ejemplo, es usada en el área musical en la creación y arreglos de temas musicales, además de ser aplicada en los software para expresión musical, pudiendo hacer uso del teclado de la computadora o de un controlador MIDI con el fin de complementar y mejorar el análisis, la audición y la composición, además de poder interpretar partituras, hacer grabaciones en tiempo real, visualizando y escuchando luego la interpretación y analizando los errores producidos.

La creación, modificación y conversión de ficheros de audio son acciones muy usadas en los medios audiovisuales, todas estas acciones son insertadas en la edición de audio con el fin de obtener el sonido de la manera deseada y con la calidad requerida.

En el marco de la edición de audio el hombre ha creado software con procesos de edición asociados para lograr un alto funcionamiento en la aplicación. A continuación se exponen algunos de los principales procesos que contiene un sistema de edición de audio.

- Grabación de audio: grabar audio desde un micrófono o cualquier dispositivo que pueda conectar al puerto de entrada de línea de una tarjeta de sonido.
- Reproducción de audio: permite la reproducción lineal del audio de un punto temporal a otro. Se puede reproducir la parte visible de un archivo, el área desde el indicador de tiempo de inicio al final del archivo o el archivo entero.
- Importación de archivos de audio: abrir archivos en una amplia gama de formatos de audio.
- Copiar una selección de audio: esta acción permite extraer una porción de audio desde la pista, duplicarla dentro de la misma pista o crear una pista nueva sólo con la selección realizada.
- Cortar y Pegar una selección de audio: mover un fragmento de sonido en alguna posición que se selecciona dentro de la misma pista de sonido o moverla a una nueva sesión.
- Aplicar filtros: efectos que se le aplica a una onda para transformar algunas de sus propiedades.
- Aplicar efectos: cambios que se le adhieren a un audio determinado.

Capítulo 1 : Fundamentación Teórica

- Hacer un mix: proceso de mezclar todas las pistas en una sola.
- Disminución de ruido: reducir el ruido en los archivos de sonido.
- Segmentación de ficheros de audio: fraccionar el fichero en pequeños segmentos que puedan ser tratados de forma independiente.
- Encadenar ficheros de audio: unir varios segmentos de audio para manipularlo como un único segmento.
- Aumentar o disminuir el sonido: aumentar o disminuir el volumen del sonido.
- Normalizar el audio: ajustar el volumen de la pista y los canales de audio a un valor en porcentaje o decibeles determinado, para evitar por ejemplo saturaciones en una pista con demasiada amplificación.
- Conversión de formatos de sonido: transformar el formato de un archivo de audio a otro formato.
- Masterizar: proceso de optimizar una grabación musical para su venta comercial. Tiene el objeto de compactar los distintos instrumentos o componentes de la obra y someterlos a un procesado conjunto. Entre las soluciones que permite la masterización las más comunes son:
 - Homogeneizar un trabajo de muchas pistas para que pueda ser entendido en conjunto.
 - Dar volumen a la grabación.
 - Destacar aspectos del sonido que resulten atractivos mediante procesados acústicos de complejidad diversa.

1.8 LOS PROCESOS DE EDICIÓN DE AUDIO EN EL PROYECTO CCM.

En la UCI se han creado proyectos para el desarrollo de los medios audiovisuales dada la necesidad de informatizar los procesos de gestión y procesamiento de medias que se maneja en la misma, además para lograr que las medias de audio y video cuenten con una mejor organización, puedan ser localizadas fácilmente y se encuentren en el formato requerido.

A partir de las diferentes temáticas de desarrollo existentes en la Universidad se han creado centros productivos dedicados a aplicar la informática a ramas específicas del conocimiento, uno de estos centros es el de Geoinformática y Señales Digitales, en el que se encuentra el departamento de Señales Digitales, que labora en el desarrollo de productos y servicios informáticos en la área de la televisión, gestión de contenido audiovisual, procesamiento digital de video y sonido; todo sobre la base de la informatización de

Capítulo 1 : Fundamentación Teórica

nuestra sociedad como prioridad y luego la comercialización como parte de la experiencia adquirida en el campo del video y sonido digital.

El proyecto Captura y catalogación de medias perteneciente al Departamento de Señales Digitales del Centro Geoinformática y Señales Digitales permite catalogar y organizar los distintos materiales de medias que son obtenidos y gestionados a partir de diferentes fuentes, funcionalidades que son necesarias en diferentes entidades sociales, lo que lo hacen un producto de alta demanda en el mercado actual. Los materiales que se gestionan y catalogan, deben contar con una alta calidad para el correcto desenvolvimiento de los procesos que automatiza el producto, específicamente en el caso del audio: su catalogación y transcripción.

La edición de audio siempre ha sido importante en aras de ganar calidad en el sonido. Para lograr un mejor estado en los ficheros de audio que son capturados o gestionados se hace necesario contar con una aplicación que permita mejorar las posibles imperfecciones que puedan tener las mismas a la hora de su grabación, ya sea desde la propia aplicación desarrollada o las grabaciones externas que se hayan realizado, que pueden contar con señales ruidosas. Un editor de audio da solución a estos problemas y además sería un valor agregado para el producto de Captura y catalogación de medias, el mismo deberá ser capaz de realizar los siguientes procesos, que se han identificado según las necesidades del equipo de desarrollo y de los futuros clientes.

1.9 SOLUCIONES DE EDICIÓN DE AUDIO EXISTENTES.

Existen diversos software para la edición de audio, a continuación se describen el análisis y las características operacionales de los más significativos.

1.9.1 Adobe Audition.

Aplicación profesional para la captura y el tratamiento de audio, creada por Adobe. Puede configurar varios parámetros de captura así como aplicar gran cantidad de filtros de audio.

Permite combinar varios sonidos de audio, editarlos y exportarlos con infinidad de formatos de audio. (21) Es una aplicación totalmente propietaria que solo funciona sobre Windows XP y necesita altos recursos de hardware. Presenta una gran cantidad de procesos para la edición de audio, los cuales son mencionados a continuación:

Capítulo 1 : *Fundamentación Teórica*

Permite abrir archivos en una amplia gama de formatos de audio, crear un nuevo archivo de audio en blanco, abrir archivos de audio existentes, así como agregar uno o más archivos de audio a otro, además de restaurar el sonido, eliminar chasquidos del micrófono, zumbidos y otros defectos de audio; arreglar segmentos de sonido y reparar grabaciones con los supresores de chasquidos/detonaciones, reducción del ruido. Posibilita el uso de aplicaciones de vídeo, así como la importación del mismo. Ofrece controles de transporte para reproducir, grabar, detener, pausar, avanzar y rebobinar. Presenta un método de grabación de sonido que permite registrar múltiples fuentes sonoras por separado para luego unirlos y formar un todo, esto se conoce como Grabación Multipista, la misma consiste en mezclar varios temas musicales y crear un solo tema, aplicando transiciones, efectos, música de fondo, sonidos, pausas etc. Acepta un máximo de 128 pistas. Admite extraer el audio de algún video MPG, AVI o WMV y amplificar el sonido, hay ocasiones que las pistas de audio están grabadas con una amplificación demasiado baja que dificulta su audición, para esto se puede amplificar el audio especificando el nivel de decibelios (db).

1.9.2 Audacity.

Es un programa libre y de código abierto para grabar y editar sonidos, lo que permite su constante innovación y mejora por parte de los usuarios. El programa está disponible para los principales sistemas operativos como Mac OS X, Microsoft Windows, GNU/Linux y otros sistemas operativos. (22)

Es el editor de audio de software libre por excelencia, dado que posee una Licencia Publica General de GNU (GNU GPL). Puede trabajar con diferentes formatos de audio (WAV, AIFF, AU, Ogg Vorbis y diferentes versiones de MPEG). En cuanto a la calidad del sonido graba y edita muestras de 16-bit, 24-bit y 32-bit, hasta una frecuencia de muestreo máxima de 96 KHz. Las frecuencias de muestreo y formatos son convertidos mediante un proceso de alta calidad. Permite mezclar pistas con diferentes frecuencias de muestreo o formatos. (23)

Presenta las clásicas opciones de copiar, cortar, pegar. Permite la edición de pistas múltiples, grabación de audio en tiempo real, mezcla de pistas con diferentes frecuencias de muestreo o formatos. Ofrece la posibilidad de cambiar la velocidad o el tono de una grabación, utiliza ilimitados niveles de deshacer (y rehacer) para volver a cualquier estado anterior. Posibilita la rápida edición de archivos grandes. Carga complementos VST para Windows y Mac.

Capítulo 1 : Fundamentación Teórica

Permite insertar efectos tales como:

- Eliminar ruidos estáticos, silbidos, tarareos u otros ruidos de fondo constantes.
- Alterar las frecuencias con la ecualización, filtros FFT y amplificar los bajos.
- Ajustar los volúmenes con el compresor, amplificar y normalizar los efectos. (24)

Desventajas: No trabaja con los formatos WMA, AAC, o el resto de los formatos de archivos propietarios o restringidos.

1.9.3 Wavosaur.

Wavosaur es un fuerte editor de audio que funciona bajo Windows 98, Windows XP y Windows Vista, ideal para acometer la edición de clips de audio, diseño musical, masterización, producción de sonido, grabaciones de audio digital, etc. Presenta una interfaz de múltiples documentos y todas las características necesarias para editar audio, como son (copiar, pegar, cortar, mezclar, eliminar, deshacer, etc.). Permite producir loop de música, grabar, normalizar el audio, así como el procesamiento en lotes, lo cual consiste en aplicar procesos, efectos, o plugins VST a una carpeta completa. Además presenta la posibilidad de síntesis, en esta opción se pueden generar ondas, impulsos y barrido de frecuencias. (24). Admite generar informes con estadísticas de uso. Soporta drivers ASIO (Audio Stream Input/Output), salida de audio de hasta 192 KHz, 24 bits. Soporta archivos MP3, wav multicanal, y procesamiento de efectos en tiempo real. (25). Mediante Wavosaur se puede automatizar envolventes de volumen y análisis espectral de sonidos así como crear nuevos ficheros de audio. Presenta varias opciones de procesamiento como son: conversión de canal de estéreo a mono y viceversa, insertar silencios, es decir, poner en mudo un fragmento de la pista en el momento deseado, eliminar silencio en la zona de la pista deseada, así como elimina la voz, pudiéndose escuchar solamente el background. (26). Presenta la opción de crear volumen con lo cual se puede insertar una línea con 4 puntos predefinidos, logrando así modificar los niveles de volumen en cada uno de los canales, ajustar el volumen de un audio, normalizar niveles y aplicar efectos tales como: Fade In / Out, Pitch shift, este último consiste en cambiar el tono de una grabación sin alterar su duración, y efecto invertir, en este es posible invertir las muestras, es decir, los desplazamientos de frecuencia positivos son negativos y todos los desplazamientos negativos son positivos. En las ondas estéreo se invierten ambos canales. (27)

Capítulo 1 : Fundamentación Teórica

1.9.4 Power Sound Editor.

Power Sound Editor es un programa de edición de audio que presenta las herramientas necesarias para grabar, editar y mezclar sonidos, soportando muchas de las más avanzadas y poderosas operaciones con los datos de audio. Este editor se puede ejecutar en sistemas operativos tales como: Windows 98, Windows 98 Second Edition, Windows Millennium Edition, Windows 2000, Windows XP y Windows Vista. La aplicación permite copiar y pegar fragmentos, añadir efectos de sonido, ecualizar el resultado y además guardarlo como un archivo nuevo. Presenta un entorno muy visual. Power Sound Editor admite trabajar con proyectos con una calidad de 8kHz a 48kHz tanto en mono de 8 bits como en estéreo de 16 bits. Incluye una serie de herramientas adicionales que potencian sus posibilidades. Por ejemplo, incluye un sintetizador de voz, una herramienta para ripear las pistas de un CD, varios visores de espectro y tres algoritmos para la reducción del ruido. Mediante este editor se puede abrir, crear y grabar archivos de sonido en cualquiera de los formatos soportados (MPEG (MP3, MP2), WAV, Windows Media Audio, Ogg Vorbis, y pista de audio. Facilita ripear un CD a cualquiera de los formatos soportados. (7). Ofrece la posibilidad de grabar audio desde un micrófono o cualquier entrada disponible, además de ver la forma de la onda musical y realizar zoom completo, zoom in, zoom out y zoom vertical. Mediante este programa se pueden editar los ficheros de audio visualmente (borrar la selección, borrar el silencio, pegar, pegar desde archivo, hacer un mix y hacer un mix desde un fichero), así como aplicar diferentes efectos a cualquier porción de los ficheros de audio, estos efectos pueden ser: Amplificador, Compresor, Retraso, Ecualizador, Fade In y Fade Out, Flanger, Invertir, Normalizar, Reverberación, Revertir, Silencio, Shrink, Stretch, Vibrato, etc.).

1.10 CONCLUSIONES PARCIALES.

En el presente capítulo se ha recogido información referente al estudio de la situación actual del dominio del problema, se pudo verificar la utilidad y necesidad de un sistema capaz de editar grabaciones de audio. Se abordó un estudio de las principales soluciones de edición de audio que existen en el mercado de software actual. Además, han sido estudiados los formatos y códecs llegando a la conclusión que serán utilizados los formatos WAV y MPEG-1 Layer 3 y los códecs PCM y LAME MP3 Códec respectivamente.

Capítulo 2: Tendencias y Tecnologías Actuales

Capítulo 2. Tendencias y Tecnologías Actuales.

2.1 INTRODUCCIÓN.

Las herramientas y tecnologías para la construcción de software constituyen hoy día instrumentos fundamentales que garantizan la buena calidad de los procesos de desarrollo. En el presente capítulo se presenta un informe del estudio de las tecnologías y tendencias actuales en el desarrollo de software con el objetivo de proponer la herramienta, lenguaje de programación a utilizar, así como el lenguaje de modelado, metodología y herramientas de apoyo a emplear.

2.2 LENGUAJE DE PROGRAMACIÓN.

En la comunicación hombre-máquina existe una dificultad real: las máquinas operan en bits (ceros y unos) y los hombres se entienden por medio de idiomas, aunque al principio del nacimiento de la informática el hombre trató de dominar el lenguaje de las mismas nunca obtuvo resultados reales, por tanto, hizo falta un vehículo entre ambos, algo que el hombre llegara a dominar y que las máquinas llegaran a entender, de ahí surgen los lenguajes de programación que no son más que *“una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen un programa informático.”* (28)

Un lenguaje de programación permite a un programador especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones deben tomar bajo una variada gama de circunstancias. (29)

Existen más de 2500 lenguajes de programación incluyendo las diferentes versiones de estos. (30) Según el nivel de trabajo de los lenguajes de programación pueden ser clasificados en tres grupos:

- Lenguajes de maquina: Es el único que entiende directamente la computadora, utiliza el alfabeto binario que consta de los dos únicos símbolos 0 y 1, denominados bits (abreviatura inglesa de dígitos binarios). Fue el primer lenguaje utilizado en la programación de computadoras, pero dejó

Capítulo 2 : Tendencias y Tecnologías Actuales

de utilizarse por su dificultad y complicación, siendo sustituido por otros lenguajes más fáciles de aprender y utilizar, que además reducen la posibilidad de cometer errores. (29)

- Lenguajes de bajo nivel: Proporciona poca o ninguna abstracción del microprocesador de un ordenador. Consecuentemente es fácilmente trasladado a lenguaje de máquina. En general se utiliza este tipo de lenguaje para programar controladores (drivers). (29)
- Lenguajes de alto nivel: Logran la independencia del tipo de máquina y se aproximan al lenguaje natural.

El proyecto Captura y Catalogación de Medias emplea Java como lenguaje de programación. Teniendo en cuenta que la solución propuesta constituirá un módulo integrado a dicho proyecto se utilizará este lenguaje en el desarrollo del editor de audio. En el siguiente epígrafe serán abordadas las principales características del lenguaje Java.

2.2.1 Lenguaje Java.

Es un lenguaje de programación desarrollado por Sun Microsystems a principios de los años 90, completamente orientado a objetos. El lenguaje en sí mismo toma mucha de su sintaxis de C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, como la manipulación directa de punteros o memoria. Las aplicaciones Java están típicamente compiladas en un código de bytes, que normalmente es interpretado en tiempo de ejecución y llevado a código de la máquina nativa, aunque la compilación en código máquina nativo también es posible. (31)

Principales Ventajas

El lenguaje Java está completamente orientado a objetos, por lo que soporta las tres características propias de este paradigma: encapsulación, herencia y polimorfismo (32). Su notación es muy similar al C++ precisamente para facilitar su uso y aprendizaje. Java permite la captura, reproducción y la manipulación de ficheros de audio, ya que presenta librerías nativas que facilitan el trabajo con sonido. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o

Capítulo 2 : Tendencias y Tecnologías Actuales

corrupción de memoria, que era una cuestión importante en el desarrollo de aplicaciones con C++. Una de sus principales ventajas radica en que es completamente multiplataforma ya que genera ficheros con código de bytes de alto nivel independiente de la máquina. Este código (bytecode) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema que corre en tiempo real, que sí es dependiente de la PC que lo está corriendo. Además como el acceso a memoria no se hace directamente usando punteros desde la aplicación, es mucho más seguro.

Principales Desventajas

Java es un lenguaje interpretado, esto trae como consecuencia que a pesar de ser muy portable dado que la ejecución de su código no depende en lo absoluto del sistema en que fue desarrollado, sea mucho más lento en su ejecución. Por otra parte depende para su funcionamiento de una máquina virtual, la cual ocupa memoria RAM en la interpretación del código, que al ser bastante complejo, dada las tareas que deberá asumir, puede disparar el consumo de recursos ralentizando además del sistema completo las tareas para las cuales fue destinado. Otra de sus desventajas es que no permite la sobrecarga de operadores, que limita mucho el uso que se les puede dar a los operadores dentro del programa teniendo que escribir mucho más en ocasiones donde una simple sobrecarga hubiera resuelto el problema. Su código puede ser a veces redundante en comparación con otros lenguajes, debido a las frecuentes declaraciones de tipos y conversiones de tipo manual (casting) (31).

Java a pesar de consumir recursos de hardware en su ejecución, es completamente multiplataforma, por lo que todo lo que se realiza en el sistema sería completamente portable. Posee una curva de aprendizaje muy rápida y permite también que el ejecutable que genera funcione en la mayoría de los sistemas operativos teniendo como base la máquina virtual de java. Una característica fundamental de su selección es que implementa gran cantidad de librerías nativas para el trabajo con audio, aspecto que facilitará el trabajo a los futuros desarrolladores de la solución propuesta.

2.3 HERRAMIENTAS DE APOYO.

Para facilitar el trabajo con el audio son utilizados diferentes componentes como: Interfaz de programación de aplicaciones (API por sus siglas en inglés), librerías y framework. Con el objetivo de seleccionar

Capítulo 2 : Tendencias y Tecnologías Actuales

herramientas de este tipo para el desarrollo del subsistema propuesto, se detallan a continuación algunas de sus principales características.

2.3.1 Java Sound API.

El API de Java dedicada al sonido es la llamada Java Sound API, el cual proporciona soporte para las operaciones de audio, tales como la reproducción de audio y la captura (grabación), la mezcla, secuenciación MIDI, y la síntesis de MIDI en un marco extensible y flexible.

Mediante Java Sound se puede reproducir de una forma sencilla la gran mayoría de los formatos de audio, como WAV, MIDI, AU, AIFF. Permite tratar audio de 8 o 16 bits, en mono o estéreo, muestreado entre las frecuencias de 8kHz y 48Khz y audio en forma de stream.

Este API se compone de 4 paquetes:

`javax.sound.sampled.`

`javax.sound.sampled.spi.`

`javax.sound.midi.`

`javax.sound.midi.spi.`

El primero, *javax.sound.sampled*, contiene las clases necesarias para el manejo del sonido muestreado, esto incluye la captura, la mezcla y la reproducción de audio, proporcionando además algún control y efecto sobre el sonido así como interfaces para el almacenamiento.

El paquete *javax.sound.midi* proporciona las interfaces de síntesis, secuenciamiento y transporte MIDI.

Los paquetes *javax.sound.sampled.spi* y *javax.sound.midi.spi* proporcionan una interfaz para los desarrolladores de servicios basados en las interfaces anteriores.

2.3.2 Java Media Framework (JMF).

Java Media Framework es una API utilizada para incluir multimedia en las aplicaciones de Java, permite básicamente el tratamiento de audio y video desde java, proporcionando a los applets y aplicaciones Java la capacidad de reproducir, capturar y transmitir/recibir en tiempo real audio, video y otros contenidos multimedia.

Capítulo 2 : Tendencias y Tecnologías Actuales

Mediante JMF se pueden utilizar los distintos formatos de vídeo, como MPEG, AVI, MOV, capturar streams de vídeo y también se puede reproducir audio, y aunque el tratamiento es algo diferente es igualmente efectivo.

Provee de una serie de codificadores y decodificadores para los formatos multimedia más relevantes siendo capaz además, de realizar transcodificación entre dichos formatos. (33)

Sus principales características son:

- Estabilidad debida a que funciona sobre la máquina virtual de java.
- Sencillez, ya que permite, usando unos pocos comandos, realizar complejas tareas multimedia.
- Potencia, permitiendo la manipulación de elementos multimedia de vídeo y audio locales (procedentes de la misma máquina en la que se ejecuta el programa), así como la retransmisión en tiempo real de vídeo y audio a través de la red mediante el protocolo RTP (Real-time Transport Protocol), este se utiliza en JMF para proporcionar a los usuarios una forma de flujos de medios de transporte a los nodos de la red.

Con la aparición de las API Java Media Framework y Java Sound ha cambiado el panorama multimedia de Java. Gracias a ellas ya es posible desarrollar aplicaciones multimedia que utilicen audio y vídeo de una forma sencilla y práctica, sin perder por ello la compatibilidad que siempre ofrece Java. El conjunto de posibilidades se ha incrementado de manera notable. Gracias a estas dos API, se pueden crear aplicaciones multimedia que hasta ahora se desarrollaban con Visual Basic, Delphi o similares, con la ventaja de la portabilidad, la orientación a objetos y la integración con el lenguaje.

2.3.3 JSyn (Java síntesis).

JSyn es una API gratuita para el desarrollo de aplicaciones interactivas de sonido en Java. JSyn utiliza métodos nativos escritos en 'C' para ofrecer en tiempo real la síntesis de audio para los programadores de Java. Puede ser usado para generar efectos de sonido, los entornos de audio, o la música. (34)

Características

Soporte para entrada de audio para grabación de voz y procesamiento.

Capítulo 2 : Tendencias y Tecnologías Actuales

Soporte para Windows, Macintosh y Linux.

Se puede utilizar desde una aplicación o un applet de Java utilizando el Internet Explorer, Firefox, Safari y complementos del navegador. (34)

En cuanto a la licencia el plug-in JSyn es libre. Los desarrolladores pueden utilizar el JSyn libremente para fines no comerciales. Los desarrolladores con licencia tienen acceso a herramientas de uso extendido y puede ser utilizada JSyn con fines comerciales. Los desarrolladores comerciales pueden adquirir la licencia de JSyn para la redistribución de sus productos, o para el uso en su sitio web comercial.

2.3.4 Ffmpeg.

FFmpeg es una colección de software libre y gratuito dedicado a la codificación, decodificación y filtrado de audio y video, permite grabar, convertir y hace streaming de audio y vídeo. Es un conversor muy rápido e incluye libav codec, una biblioteca de códecs. Ffmpeg está desarrollado en GNU/Linux, aunque puede ser compilado en la mayoría de los sistemas operativos, incluyendo Windows. Presenta una variedad de funcionalidades como extraer el sonido de un vídeo y guardarlo como MP3, la conversión de formatos de audio, obtener información de un fichero, la extracción de un stream, añadir pistas, cortar un segmento, subir el volumen del audio y la captura de stream de Internet.

A través de la clase *Runtime* de Java se pueden ejecutar los comandos del Ffmpeg para realizar las funcionalidades necesarias que debe poseer la aplicación. Esto permite la vinculación del Ffmpeg con el Java facilitando de esta manera la interacción entre ellos.

2.4 SELECCIÓN DE LAS MÁS IDÓNEAS.

Después de un estudio de las herramientas que en la actualidad resuelven los procesos de edición de audio se decide utilizar Java Sound API (*Interfaz de programación de aplicaciones de sonido en Java*) y Ffmpeg, ya que permiten su implementación en software libre y presentan las características necesarias para dar solución a los procesos de edición de audio que han sido identificados.

Java Sound API será utilizado para dar solución a los procesos de establecer volumen, aplicar efectos; además de permitir la utilización de los controles de reproducción, pausa, rebobinado y avance rápido de

Capítulo 2 : Tendencias y Tecnologías Actuales

una pista de audio. Estos serán implementados haciendo uso del paquete `javax.sound.sampled`. Por su parte los procesos de normalización, segmentación y concatenación del audio se realizarán mediante el `Ffmpeg`, ya que brinda tales posibilidades, además se utilizará en la conversión de los formatos de audio.

2.5 ENTORNO DE DESARROLLO INTEGRADO.

Un entorno de desarrollo integrado o IDE (del inglés *Integrated Development Environment*) es un programa compuesto por herramientas que son necesarias al programador a la hora de implementar. Puede ocurrir que el programa se dedique únicamente a un solo lenguaje de programación o a varios.

Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

El equipo de desarrollo del proyecto CCM cuenta con experiencia en el trabajo con NetBeans, ya que es el IDE usado en dicho proyecto. Por lo cual se propone para la construcción del subsistema de edición de audio hacer uso del propio entorno de desarrollo, de esta manera se hace más factible y sencilla su utilización. Además NetBeans está escrito puramente sobre la base de la tecnología Java, de modo que puede ejecutarse en cualquier ambiente que ejecute este lenguaje. A continuación se exponen las principales características de esta herramienta de desarrollo.

2.5.1 NetBeans.

NetBeans soporta gran cantidad de librerías y es un IDE de desarrollo maduro. Es muy sencillo de utilizar y exige muy poco conocimiento por parte del usuario, lo cual agilizará el proceso de construcción de la solución propuesta. Es completamente software libre y tiene versiones para los principales sistemas operativos como: Windows, MacOS y LINUX, permitiendo que el producto pueda ser desarrollado en varias plataformas, ampliando las posibilidades de su uso.

Contiene herramientas para crear aplicaciones profesionales para el escritorio, la empresa, la web y equipos móviles con el lenguaje Java, C/C++, y Ruby. NetBeans IDE es fácil de instalar y de uso instantáneo.

Es un producto de código abierto, con todos los beneficios del software disponible en forma gratuita, el cual ha sido examinado por una comunidad de desarrolladores. Aparte de la filosofía de distribución y

Capítulo 2 : Tendencias y Tecnologías Actuales

desarrollo que respalda a NetBeans, el IDE ofrece a los desarrolladores numerosas ventajas, en la creación de nuevas aplicaciones multiplataforma.

La versión NetBeans IDE 6.1 provee nuevas características y mejoras, como funciones de edición enriquecida de Java Script, soporte para el uso del framework web Spring. Esta versión también provee mejor rendimiento, especialmente un arranque más rápido (más de 40%), menor consumo de memoria y mejor respuesta cuando se trabajan con proyectos grandes.

Algunas de las tecnologías y lenguajes soportadas por NetBeans son: Ajax, C/C++, Databases, Debugger, Desktop, Editor, Groovy, GUI Builder, Java EE, Java FX, Java ME, Java SE, Java Script, Mobile, PHP, Profiler, Python, Refactor, REST, Rich Client Platform, Ruby, SOA, SOAP, UML, Web, WSDL, XML.

2.6 METODOLOGÍA DE DESARROLLO DE SOFTWARE.

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Específicamente "define **quién** está haciendo **qué**, **cuándo** y **cómo** para alcanzar un determinado objetivo. (35). Un proceso de software detallado y completo suele denominarse "Metodología". (36)

Una metodología de desarrollo de software es un conjunto de pasos o procedimientos a seguir para obtener un producto informático de calidad y en el tiempo establecido. Entre las metodologías de desarrollo de software más difundidas hoy en día se pueden mencionar: Rational Unified Process (RUP), Extreme Programming (XP) y Scrum.

Se describen en el siguiente epígrafe las principales características de la metodología RUP, la cual ha sido seleccionada como metodología de desarrollo de software a usar en la solución propuesta.

2.6.1 Rational Unified Process (RUP)

Rational Unified Process (RUP) apareció en 1998, creado por James Rumbaugh, Grady Booch e Ivar Jacobson para la *Rational Corporation*. El Proceso Unificado de Rational es un proceso de desarrollo de

Capítulo 2 : Tendencias y Tecnologías Actuales

software. Según sus autores el proceso de desarrollo de software lo conforman el conjunto de actividades necesarias para transformar los requisitos funcionales de un usuario en un sistema de software. Sin embargo el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Lo que sustenta el proceso de desarrollo de software son: el proyecto, las personas, el producto y el proceso, existe una estrecha relación entre ellas.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software.

Los principales rasgos que definen unívocamente al Proceso Unificado de Desarrollo de Software lo constituyen el ser dirigido por casos de uso, centrado en la arquitectura así como iterativo e incremental. (37). A continuación se describen estos tres aspectos:

Dirigido por casos de uso: El proceso de desarrollo está dirigido por casos de uso, sigue un hilo que avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

Centrado en la arquitectura: La arquitectura es el elemento central a partir del cual el sistema se desarrolla y evoluciona, no solo en su desarrollo inicial, sino también a lo largo de las futuras iteraciones y generaciones, por lo que esta debe permitir el correcto desarrollo de todos los casos de uso requeridos tanto en el presente como en el futuro.

Iterativo e incremental: En RUP el trabajo se divide en partes más pequeñas o mini-proyectos llamados iteraciones, con la culminación de cada iteración se logra un incremento o crecimiento del producto.

RUP se divide en 4 fases:

- Inicio El objetivo de esta etapa es determinar la visión del proyecto.
- Elaboración: El objetivo es determinar la arquitectura óptima.
- Construcción: El objetivo es obtener la capacidad operacional inicial.
- Transición: El objetivo es obtener la primera versión del proyecto (release).

Capítulo 2 : Tendencias y Tecnologías Actuales

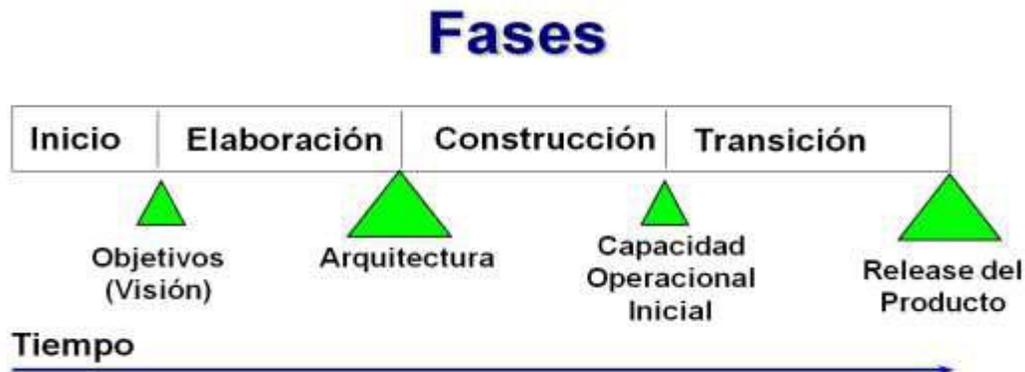


Figura 4: Fases de RUP

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Flujos de Trabajo de Ingeniería:

- Modelamiento del negocio: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.
- Despliegue: Hacer todo lo necesario para la salida del proyecto.

Flujos de Trabajo de Soporte.

- Administración de configuración y cambios: Guardando todas las versiones del proyecto.
- Administración de proyectos: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.

Capítulo 2 : Tendencias y Tecnologías Actuales

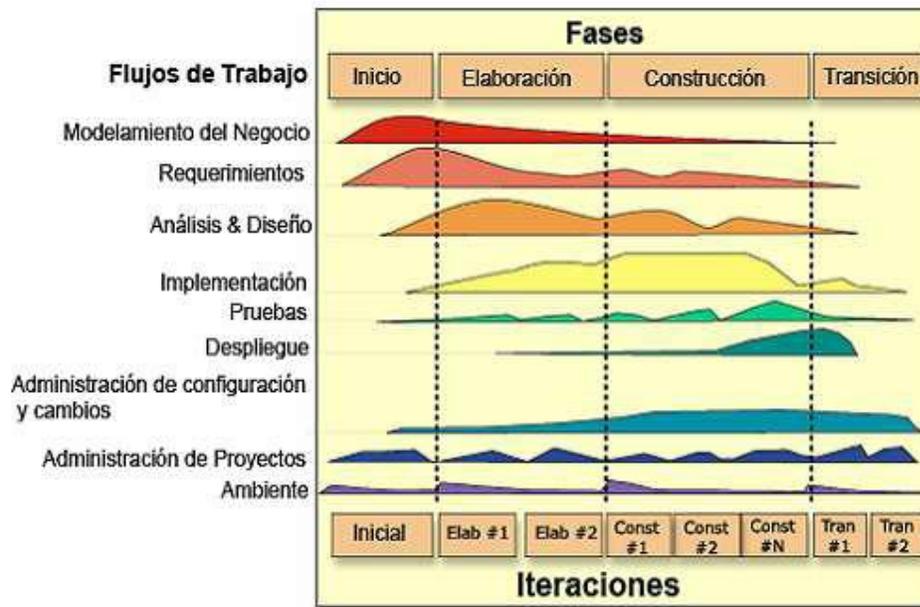


Figura 5: Flujos de trabajo de RUP

RUP es una metodología muy popular y utilizada para la documentación y realización de software orientados a objetos. Brinda grandes potencialidades y genera gran cantidad de documentación, lo cual facilitará el trabajo para la posterior versión del subsistema. RUP unifica los mejores elementos de otras metodologías, lo que garantiza que sea una metodología de primer nivel; una de las razones fundamentales es que está preparada para desarrollar proyectos grandes y complejos.

El analista de sistema en RUP.

El analista de sistemas en RUP es el encargado de dirigir y coordinar la adquisición de requisitos esquematizando la funcionalidad del sistema y delimitándolo. Una persona que actúe en este rol debe ser, por encima de todo, un experto en la identificación y la comprensión de problemas y oportunidades. Esto incluye la capacidad de articular las necesidades que se asocian con el problema clave que se debe solucionar. Además de esto, una persona que desempeñe este rol debe ser un buen moderador y debe tener habilidades de comunicación por encima de la media, así como una gran capacidad de absorber y comprender rápidamente información nueva.

Capítulo 2 : Tendencias y Tecnologías Actuales

Las actividades del analista de sistema durante el ciclo de vida de un proyecto de software se centran en los flujos de trabajo Modelación del negocio y Requerimientos. El trabajo de diploma se concentra específicamente en el flujo de Análisis y Diseño, realizando además los flujos antes mencionados y se lleva a cabo una serie de tareas que tributan a la obtención de una serie de artefactos.

Tareas:

- Buscar actores y casos de uso.
- Crear un vocabulario común
- Definir el contexto del sistema.
- Desarrollar el plan de gestión de requisitos.
- Desarrollar las especificaciones suplementarias.
- Estructurar el modelo de casos de uso.

Artefactos:

- Atributos de requisitos.
- Especificaciones suplementarias.
- Glosario de términos.
- Modelo de casos de uso.
- Paquetes de casos de uso.
- Plan de gestión de requisitos.

El rol de analista de sistemas es de vital importancia para cualquier equipo de desarrollo de software pues el mismo es el encargado de reconocer y comprender los elementos básicos del dominio del problema, obtener los requerimientos funcionales y no funcionales así como definir y describir todas las funcionalidades que tendrá el software. Para el desarrollo del subsistema de edición de audio es necesaria la presencia del analista, ya que los artefactos y la documentación que este genere posibilitaran a los futuros implementadores comprender mejor el software que se pretende desarrollar.

Capítulo 2 : Tendencias y Tecnologías Actuales

2.7 LENGUAJE DE MODELADO: UML

El Proceso Unificado de Desarrollo utiliza Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) para preparar todos los esquemas de un sistema de software. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien está involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. (38).

Características de UML:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con los lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, entre otras.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (39)

2.8 HERRAMIENTAS CASE

Las Herramientas CASE (*Computer-Aided Software Engineering*: Ingeniería de Software Asistida por Ordenador) son básicamente diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y dinero. (40)

Capítulo 2 : Tendencias y Tecnologías Actuales

Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. (41)

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunos de ellos son:

- Verificar el uso de todos los elementos en el sistema diseñado.
- Automatizar el dibujo de diagramas.
- Ayudar en la documentación del sistema.
- Ayudar en la creación de relaciones en la Base de Datos.
- Generar estructuras de código. (41)

Entre las herramientas de tipo CASE más utilizadas en la actualidad se pueden mencionar: Rational Rose, Enterprise Architect y Visual Paradigm.

2.8.1 *Visual Paradigm.*

Se escoge esta herramienta para la modelación del sistema por las facilidades que brinda para el diseño UML del ciclo de vida completo del desarrollo de software: análisis, diseño, construcción, pruebas y despliegue, todo ello orientado a objetos.

El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Posibilita el modelado de base de datos, requerimientos, proceso de negocio, permite realizar todo tipo de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (42)

Está desarrollada con el lenguaje de programación Java, por lo que tiene versiones en los sistemas operativos más conocidos. Visual Paradigm permite realizar, mediante un análisis textual, técnicas para la captura y análisis de requisitos, identificando casos de uso, actores y clases, como elementos de modelado candidatos.

Capítulo 2 : Tendencias y Tecnologías Actuales

La versión a utilizar es Visual Paradigm for UML 6.0 Enterprise Edition, la misma constituye una herramienta multiplataforma y permite modelar todos los artefactos que se obtendrán a partir del análisis del negocio y el sistema, además brinda numerosas funcionalidades como:

- Permite la captura, especificación y análisis de requisitos mediante la creación de Modelos de Análisis Textual.
- A partir de las descripciones contenidas en los análisis textuales se pueden identificar los elementos candidatos a incluir en los diagramas de casos de uso.
- Una vez identificados estos elementos candidatos se pueden utilizar para conformar los diagramas de casos de uso.
- Es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto.
- Permite control de versiones.
- Genera la documentación del proyecto automáticamente en varios formatos como Web o PDF.

Características

- Producto de calidad.
- Soporta aplicaciones web.
- Las imágenes y reportes generados, no son de muy buena calidad.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

Además Visual Paradigm ofrece:

- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Capítulo 2 : Tendencias y Tecnologías Actuales

- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDEs (Integrated Development Environmen).

Visual Paradigm presenta la característica de permitir dibujar todos los tipos de diagramas de clases y mejorar el tiempo de construcción de aplicaciones de calidad. Facilita la ingeniería inversa, la codificación desde diagramas, así como el desarrollo de documentación de una aplicación. Presenta abundante bibliografía, videos instructivos y demostraciones interactivas, que explican el trabajo con la herramienta. Para el desarrollo de la solución propuesta se ha seleccionado Visual Paradigm por las características anteriormente expuestas, además por la posibilidad de trabajar en la plataforma GNU/Linux, sistema operativo sobre el cual se desarrolla el proyecto CCM. No presenta ningún problema con su licencia para la elaboración de software comercial, el cual es un aspecto de suma importancia ya que se desea un software desarrollado completamente con herramientas libres.

2.9 CONCLUSIONES PARCIALES.

El presente capítulo recogió el resultado del estudio y la investigación relacionada a las tendencias y tecnologías actuales a desarrollar. Se llegó a la conclusión que para el editor de audio propuesto se utilizará la metodología RUP; UML como lenguaje de modelado; Visual Paradigm como herramienta case, Java como lenguaje de programación; y como IDE de desarrollo NetBeans por ser una herramienta de software libre. Se hará uso de Ffmpeg y Java Sound API como herramientas de apoyo por su gran aporte en el trabajo con audio. Teniendo en cuenta las tecnologías que se utilizan en el proyecto Captura y Catalogación de Medias se puede afirmar que la selección realizada es óptima para proporcionar una solución eficiente al problema planteado.

Capítulo 3 : *Análisis del Sistema*

Capítulo 3. Análisis del Sistema.

3.1 INTRODUCCIÓN.

En el actual capítulo se podrá contar con la representación del modelo de dominio siguiendo la guía metodológica del RUP. Se lleva a cabo la captura de requisitos funcionales y no funcionales que deberá cumplir la solución propuesta. Se describen los actores y se presentan los diagramas de casos de uso del sistema, así como la descripción detallada de los mismos.

3.2 MODELO DE DOMINIO.

Debido a que el subsistema de edición de audio a desarrollar se incluye en el proyecto Captura y Catalogación de Medias, y el mismo está concebido como un producto, el cual puede ser personalizado y comercializado a cualquier cliente o entidad interesada, resulta difícil encontrar procesos de negocio bien estructurados que permitan realizar un modelado completo de dicho negocio, por lo que se decide realizar un modelo de dominio y un glosario de términos para identificar y describir todos los términos presentes en el modelo.

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. (35)

El modelado de dominio tiene como objetivo describir y comprender las clases y objetos más significativos dentro del contexto del problema, lo cual ayuda a definir los procesos y roles más trascendentales para el sistema a desarrollar. Esto ayuda a los usuarios, desarrolladores y clientes a manejar un vocabulario común, que permita compartir el conocimiento manteniendo un lenguaje único y consistente que evite confusiones y posibilite el entendimiento de todas las partes interesadas.

3.2.1 Conceptos principales del modelo de dominio.

Editor: Es el trabajador encargado de realizar las ediciones de audio.

Archivo de audio: Contenedor multimedia que guarda una grabación de audio.

Capítulo 3 : Análisis del Sistema

Edición de audio: Acción de crear o modificar un audio digital.

Reproducir audio: Reproducir el archivo de audio.

Segmentar fichero de audio: Dividir un fichero de audio o cortar el fichero en su totalidad.

Encadenar fichero de audio: Unir ficheros de audio o una porción del mismo.

3.2.2 Diagrama de Clases del Modelo de Dominio.

Se muestran a continuación cómo interactúan los conceptos que han sido manipulados en el sistema.

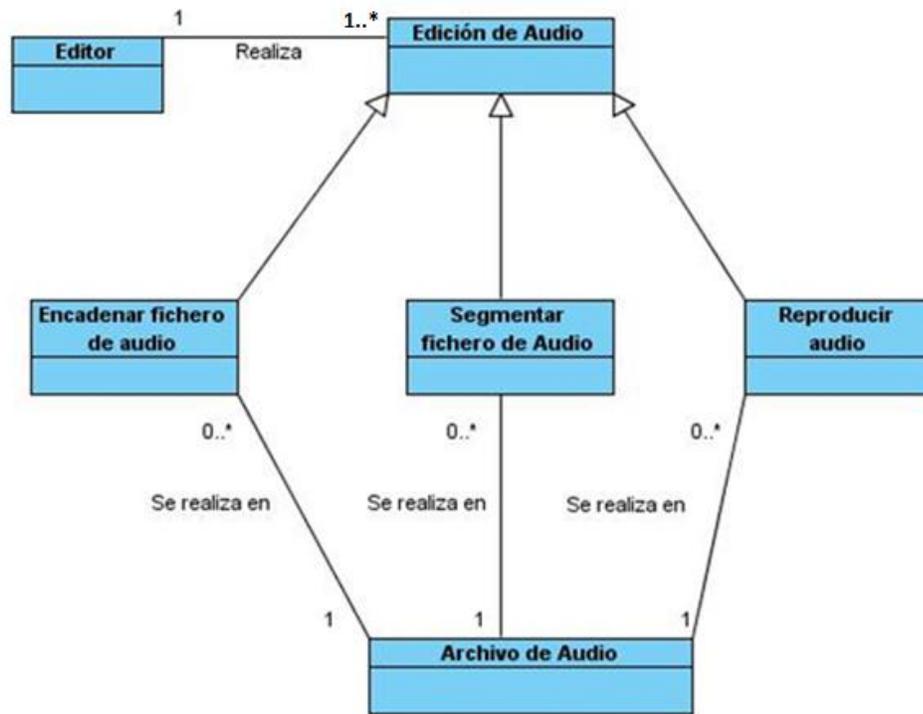


Figura 6: Diagrama de clases del modelo del dominio.

3.2.3 Descripción del Modelo de Dominio.

El editor es la persona encargada de realizar la edición de audio, en este caso puede realizar una o muchas ediciones. Reproducir audio, segmentar y encadenar ficheros de audio son procesos que se

Capítulo 3 : *Análisis del Sistema*

incluyen en la propia edición y que se realizan sobre un archivo de audio, al cual se le pueden aplicar muchos o ningún proceso de los antes mencionados.

3.2.4 Reglas del Negocio.

Para lograr una mejor edición de los archivos de audio, el propio sistema convertirá los ficheros a extensión .wav en el momento que los archivos sean cargados en el editor.

3.3 REQUERIMIENTOS.

En el libro “Ingeniería de Software. Un enfoque práctico”, el autor Roger Pressman plantea: “La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente” (43)

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema (35).

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

- Los requerimientos funcionales: definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.
- Los requerimientos no funcionales: tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de

Capítulo 3 : *Análisis del Sistema*

usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

3.3.1 *Requerimientos Funcionales.*

RF 1. Editar Audio.

El sistema deber permitir que el editor realice las ediciones necesarias al audio, tales como; disminuir ruido, encadenar, segmentar, establecer volumen, normalizar y aplicar efectos a una pista de audio.

RF 1.1 Disminuir ruido.

El sistema debe ser capaz de permitir la disminución de ruido en las grabaciones de audio.

RF 1.2 Segmentar fichero de audio.

El sistema debe brindar la posibilidad de segmentar las grabaciones de audio.

RF 1.3 Encadenar fichero de audio.

El sistema debe brindar la posibilidad de concatenar las grabaciones de audio.

RF 1.4 Establecer volumen.

El sistema debe ser capaz de permitir al editor aumentar o disminuir el volumen a los ficheros de audio.

RF 1.5 Normalizar audio.

El sistema debe permitir la normalización de los archivos de audio.

RF 1.6 Aplicar efecto.

El sistema debe brindar la posibilidad aplicar efectos a las pistas de audio.

RF 2. Reproducir audio.

El sistema debe permitir la reproducción de los archivos de audio.

Capítulo 3 : *Análisis del Sistema*

RF 3. Importar archivo de audio.

El sistema debe ser capaz de importar un archivo de audio en la aplicación.

RF 4. Exportar archivo de audio.

El sistema debe ser capaz de exportar un archivo de audio después de haber agregado las nuevas características al mismo.

RF 5. Permitir procesamiento multipista.

El sistema debe ser capaz de importar tantos archivos de audio en la aplicación como desee el editor.

3.3.2 Requerimientos no funcionales.

Con el objetivo de lograr que el software funciones, pero además, de la mejor manera posible, se han seleccionado un grupo de requisitos no funcionales que ayudarán a cumplir con las expectativas del usuario.

Requerimientos de usabilidad.

- El sistema debe ser sencillo de operar. Todo el proceso debe de ser transparente para el usuario, por lo que este no necesariamente debe poseer grandes conocimientos de la informática para trabajar con el software.

Requerimientos de interfaz gráfica.

- El software debe tener una interfaz atractiva y amigable, afín a los sistemas de edición más comunes.
- El sistema debe mostrar un ambiente profesional, sin información repetida o en exceso.
- El diseño será sugerente y permitirá al usuario navegar con facilidad e intuición por la aplicación.
- La aplicación debe ser rápida y flexible.

Capítulo 3 : *Análisis del Sistema*

Requerimientos de software.

- Tener instalada la máquina virtual de java.

Requerimientos de Hardware.

- Memoria RAM: 512 Mb DDR2(mínimo) – 1Gb DDR2 (recomendado)
- Espacio libre en H.D: 750 Mb (mínimo) – 1 Gb (recomendado).

Restricciones en el diseño e implementación:

- El sistema deberá ser desarrollado como una aplicación de escritorio. Será multiplataforma y libre.
- El sistema estará implementado en lenguaje Java, utilizando como IDE de desarrollo NetBeans. Para la modelación de la arquitectura se utiliza UML haciendo uso de la herramienta Visual Paradigm.

3.4 DESCRIPCIÓN DEL SISTEMA PROPUESTO.

El sistema está destinado a las empresas que trabajan con gran volumen de audio para lo cual es necesario el proceso de edición de audio. El sistema es una aplicación de escritorio e interactúa solamente un usuario con el sistema.

3.5 DESCRIPCIÓN DEL SISTEMA. MODELOS DE CASOS DE USO DEL SISTEMA.

Luego de haber realizado una pequeña descripción del sistema, se da paso a la descripción del modelo de casos de uso del sistema haciendo uso de las ventajas que brinda el lenguaje de modelado UML, se formulan las funcionalidades del sistema y representación mediante un diagrama, para ello es de vital importancia definir los actores y los casos de uso que representarán las responsabilidades del mismo.

3.5.1 Determinación y justificación de los actores del Sistema.

Un actor es un rol que cumple un usuario, puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. (35)

Capítulo 3 : *Análisis del Sistema*

Debido a las características particulares del sistema solamente se tiene un actor, el editor. Los casos de usos definidos se relacionan exclusivamente con este actor.

Actores	Justificación
Editor	Representa a la persona que haga uso del editor de audio.

Tabla 1: Actor del sistema y su funcionalidad.

3.5.2 *Casos de Uso del Sistema.*

Un caso de uso constituye una técnica utilizada para describir el comportamiento del sistema, a través de un documento narrativo que define la secuencia de acciones, que obtienen resultados de valor para un actor que utiliza un sistema para completar un proceso, sin importar los detalles de la implementación.

Casos de Uso del Sistema	Prioridad
Disminuir ruido.	Crítico
Segmentar fichero de audio.	Crítico
Encadenar fichero de audio.	Crítico
Establecer volumen.	Crítico
Normalizar audio.	Crítico
Aplicar efecto.	Crítico
Reproducir audio.	Crítico
Importar archivo de audio.	Crítico
Exportar archivo de audio.	Crítico
Permitir procesamiento multipista.	Crítico

Tabla 2: Prioridad de los Casos de Uso.

3.5.3 *Diagrama de Casos de Uso del Sistema.*

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que

Capítulo 3 : Análisis del Sistema

muestra la relación entre los actores y los casos de uso en un sistema. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

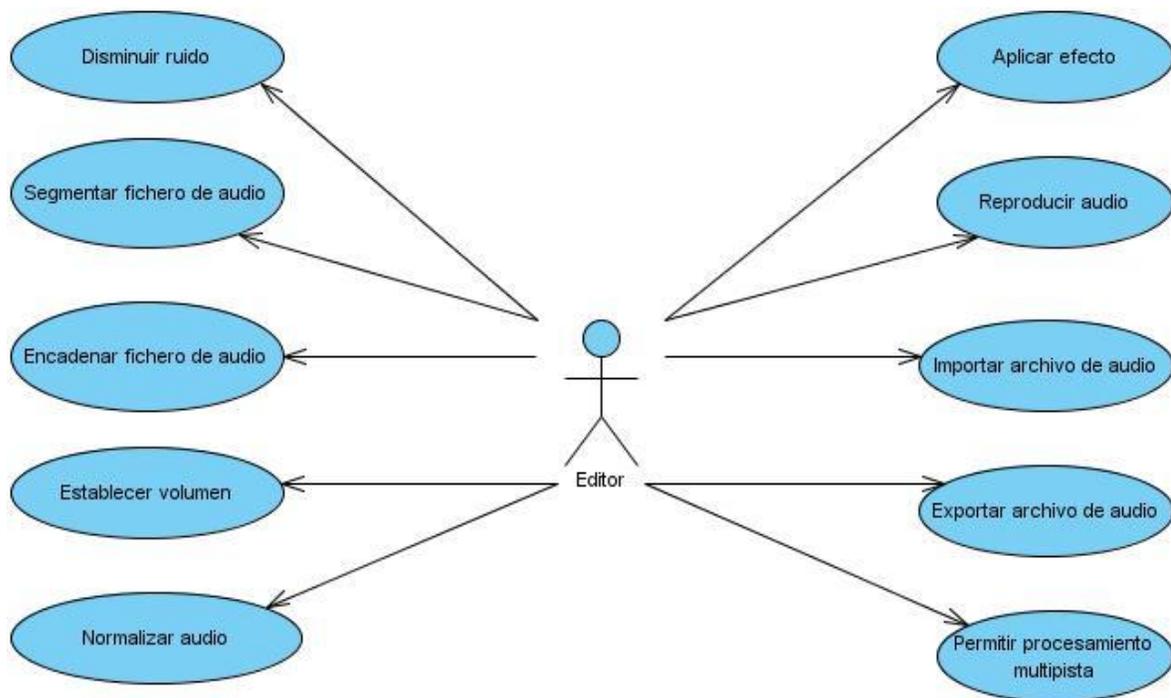


Figura 7: Modelo de Casos de Uso del Sistema.

3.6 DESCRIPCIÓN DE LOS CASOS DE USO.

Descripción del Caso de Uso	
Nombre del Caso de Uso:	Disminuir ruido.
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide disminuir el ruido a un archivo de audio.
Referencias	RF 1

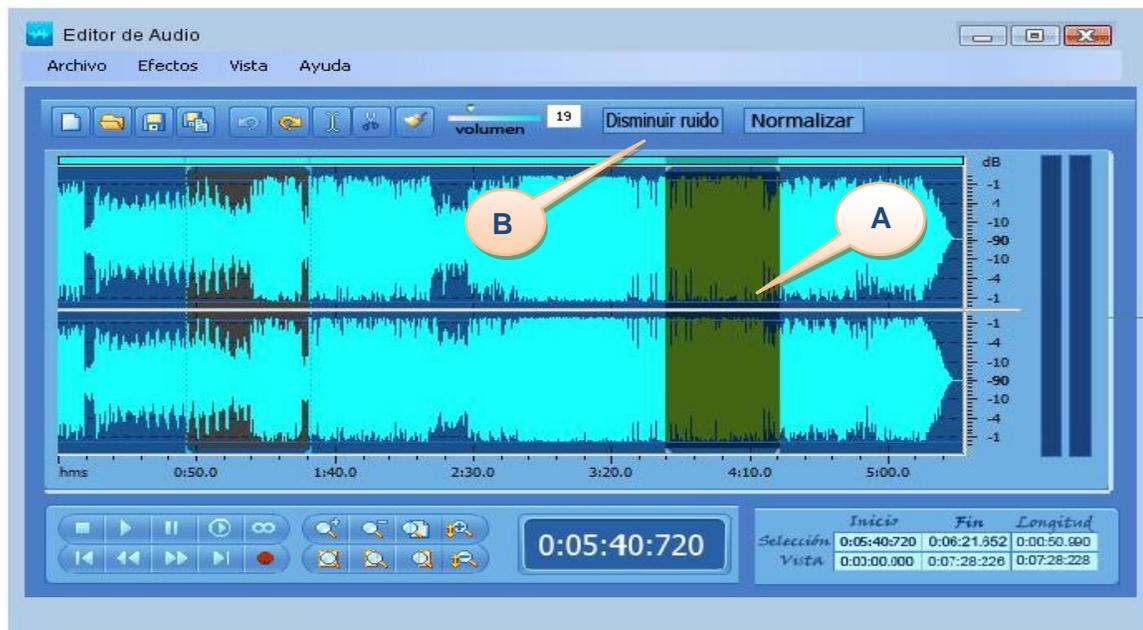
Capítulo 3 : Análisis del Sistema

Precondiciones	El actor debe haber cargado el fichero de audio en el editor.
Pos-condiciones	Se obtiene el audio con el ruido disminuido.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El editor selecciona el fragmento del audio al cual desea disminuirle el ruido (A).	
2. El editor selecciona la opción “Disminuir ruido” (B).	2.1 El sistema reduce el ruido del segmento seleccionado.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
2.1 El editor selecciona la opción “Disminuir ruido” sin haber seleccionado previamente el segmento al cual desea realizar la acción.	2.2 El sistema muestra un mensaje de error indicando que debe seleccionar el segmento al cual desea disminuir el ruido.

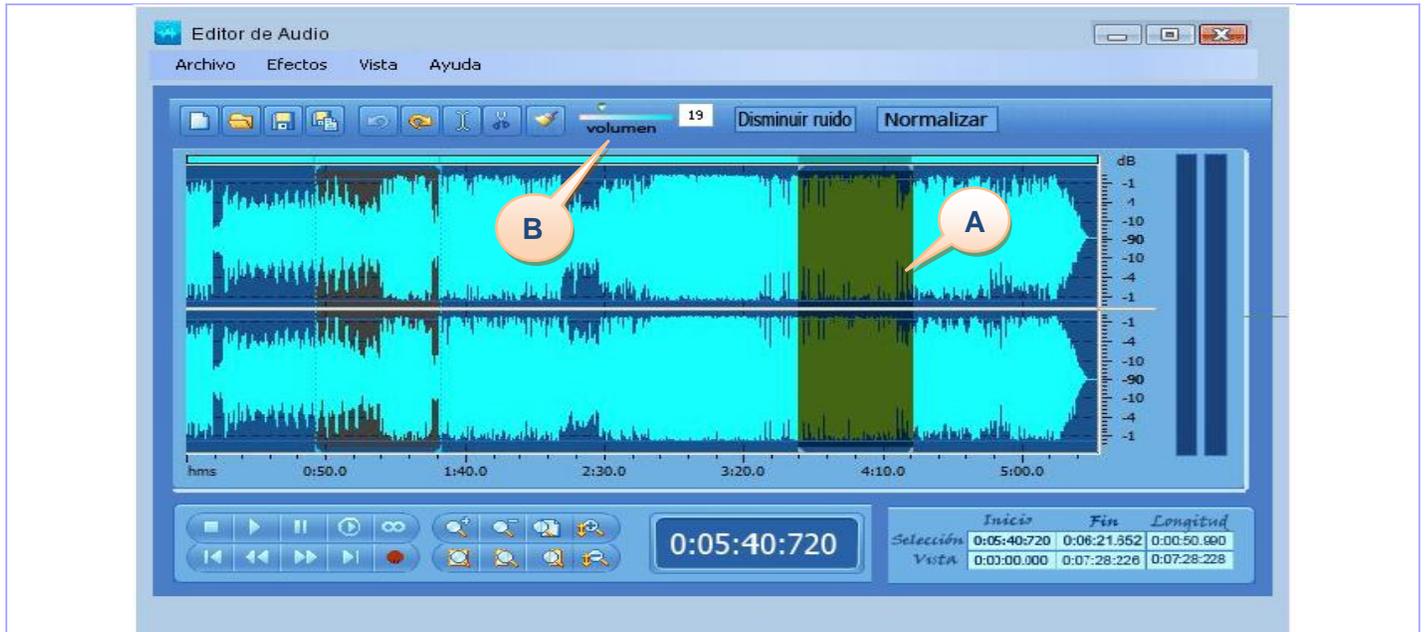
Capítulo 3 : Análisis del Sistema

Flujos Alternos	
<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
2.1 El editor escoge la opción “Cortar” sin haber seleccionado previamente la posición en la cual desea cortar.	2.2 El sistema muestra un mensaje de error indicando que debe seleccionar la posición a cortar.

Tabla 3: Descripción del caso de uso: Segmentar fichero de audio.

Descripción del Caso de Uso	
Nombre del Caso de Uso:	Establecer volumen.
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide aumentar o disminuir el volumen a un archivo de audio.
Referencias	RF1
Precondiciones	El actor debe haber cargado el fichero de audio en el editor.
Pos-condiciones	Se obtiene el audio con el volumen establecido.
Flujo Normal de Eventos	
<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
1. El editor selecciona el fragmento del audio al cual desea establecer el volumen (A).	
2. El editor selecciona la opción “Establecer volumen” (B).	
3. El editor define el nivel de volumen.	3.1 El sistema aplica los cambios de volumen en el audio.
Prototipo de Interfaz	

Capítulo 3 : Análisis del Sistema



Flujos Alternos

<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
2.1 El editor selecciona la opción "Establecer Volumen" sin previamente haber seleccionado el fragmento del audio al cual desea realizar la acción.	2.2 El sistema muestra un mensaje de error indicando que debe seleccionar el fragmento previamente.

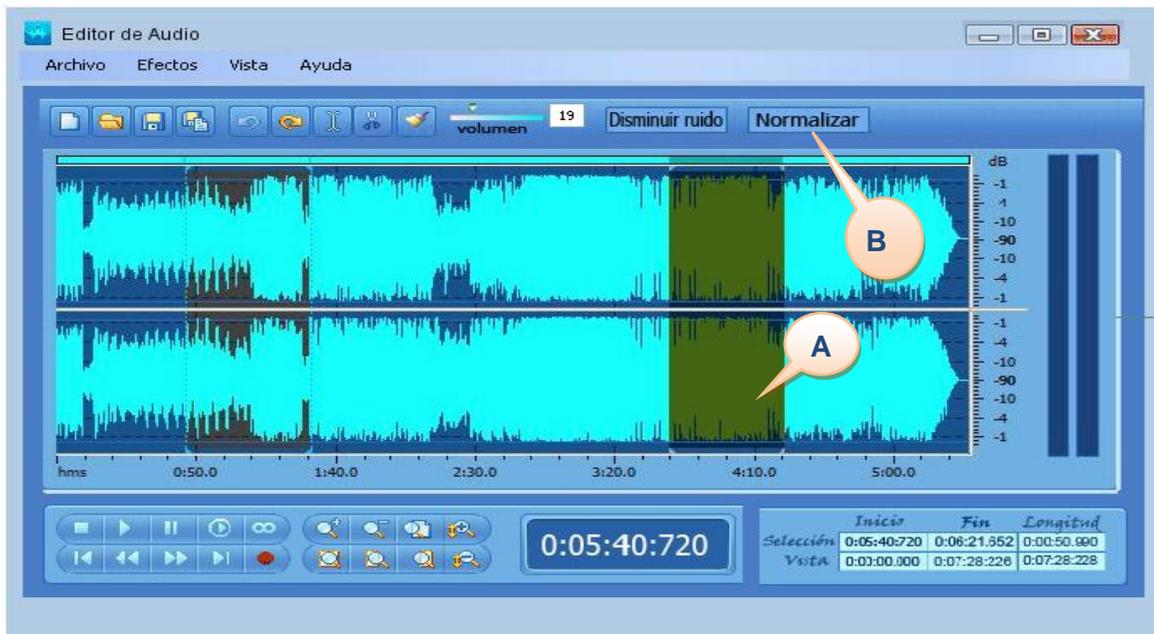
Tabla 4: Descripción del caso de uso: Establecer volumen.

<i>Descripción del Caso de Uso</i>	
Nombre del Caso de Uso:	Normalizar audio.
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide normalizar un archivo de audio.
Referencias	RF1
Precondiciones	El actor debe haber cargado el fichero de audio en el editor.
Pos-condiciones	El audio es normalizado.
Flujo Normal de Eventos	

Capítulo 3 : Análisis del Sistema

Acción del Actor	Respuesta del Sistema
1. El editor selecciona el fragmento del audio que desea normalizar.	
2. El editor selecciona la opción “Normalizar”.	2.1 El sistema aplica los cambios de normalización en el audio.

Prototipo de Interfaz



Flujos Alternos

Acción del Actor	Respuesta del Sistema
2.1 El editor selecciona la opción “Normalizar” sin previamente haber seleccionado el fragmento del audio a Normalizar.	2.2 El sistema muestra un mensaje de error indicando que debe seleccionar el fragmento previamente.

Tabla 5: Descripción del caso de uso: Normalizar audio.

Descripción del Caso de Uso

Nombre del Caso	Aplicar efecto.
------------------------	-----------------

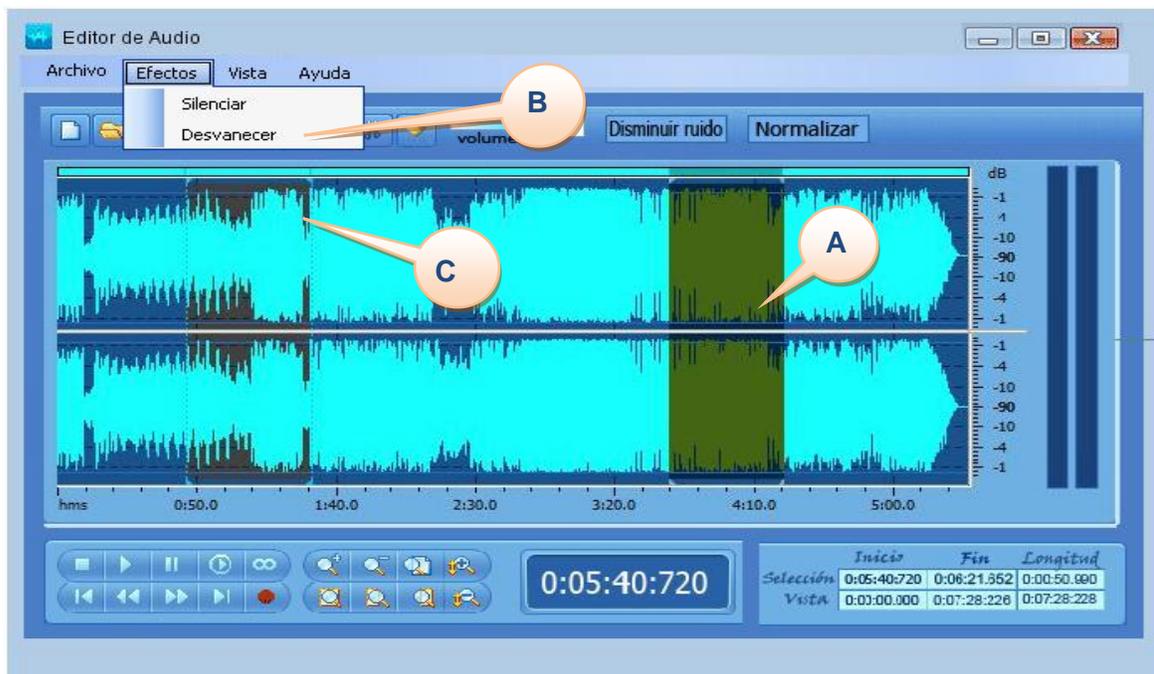
Capítulo 3 : Análisis del Sistema

de Uso:	
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide aplicar un efecto un archivo de audio.
Referencias	RF1
Precondiciones	El actor debe haber cargado el fichero de audio en el editor.
Pos-condiciones	Al audio se le aplican los efectos seleccionados.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El editor selecciona el fragmento del audio al cual desea aplicar el efecto (A).	
2. El editor selecciona la opción "Efectos" (B).	
3. El editor escoge el efecto que desea utilizar (C).	3.1 El sistema aplica el efecto seleccionado.

Prototipo de Interfaz



Capítulo 3 : *Análisis del Sistema*

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3.1 El editor escoge el efecto que desea aplicar sin previamente haber seleccionado el fragmento de audio al cual desea realizar la acción.	3.2 El sistema muestra un mensaje de error indicando que debe seleccionar el fragmento al cual desea aplicar el efecto.

Tabla 6: Descripción del caso de uso: Aplicar efecto.

Descripción del Caso de Uso	
Nombre del Caso de Uso:	Importar archivo de audio
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide importar un archivo de audio.
Referencias	RF 3
Precondiciones	El editor debe haber accedido al sistema.
Pos-condiciones	El audio se importa en el editor.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El editor selecciona el botón "Archivo" (A).	
2. El editor selecciona el botón "Importar" (B).	
3. El editor selecciona el fichero de audio que desea importar y da clic en el botón abrir del explorador.	3.1 El sistema convierte el fichero a .wav si no se encuentra en ese formato. 3.2 El sistema abre el fichero de audio.
Prototipo de Interfaz	

Capítulo 3 : Análisis del Sistema

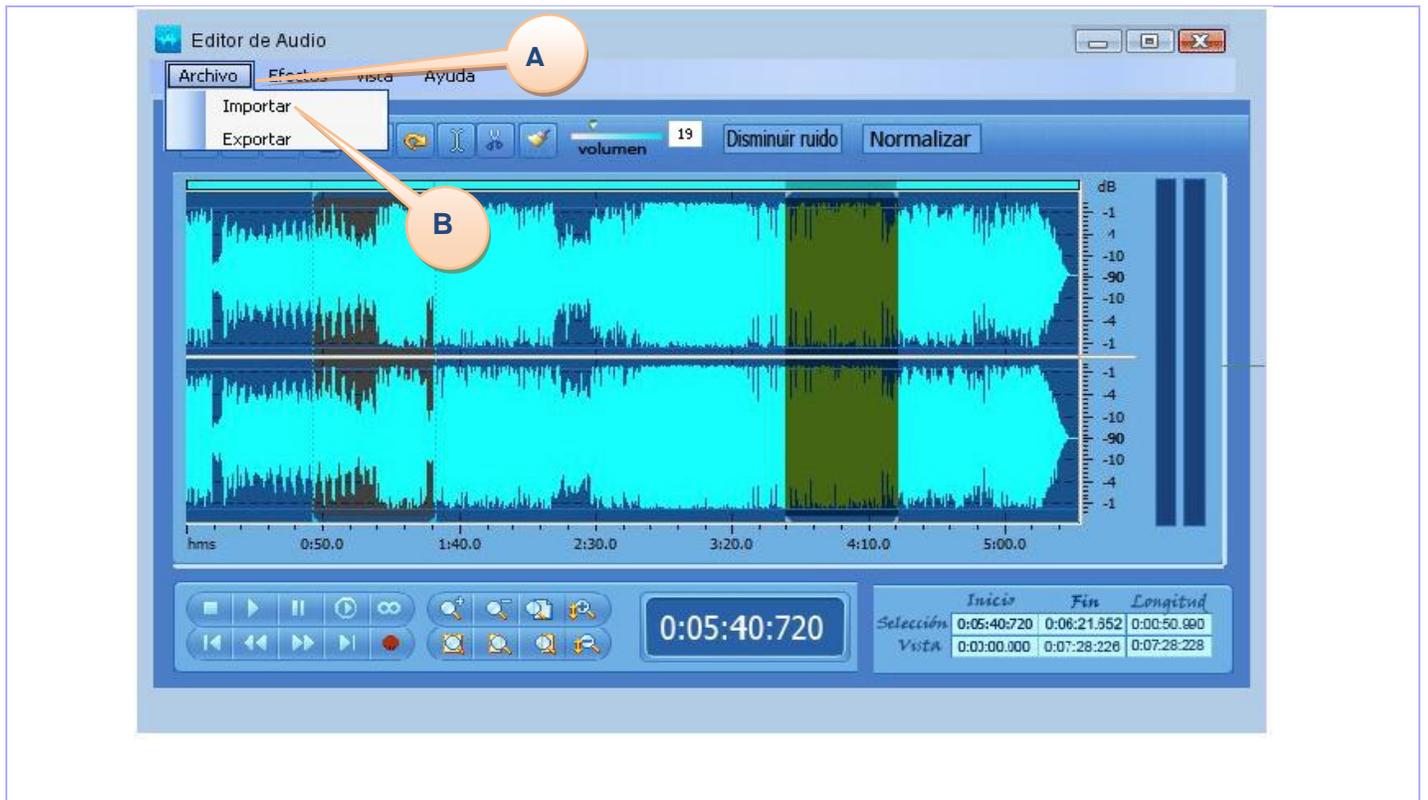


Tabla 7: Descripción textual del caso de uso Importar archivo de audio.

Descripción del Caso de Uso	
Nombre del Caso de Uso:	Permitir procesamiento multipista.
Actor(es):	Editor
Resumen	El caso de uso tiene lugar cuando el actor decide abrir más de un archivo de audio a la vez.
Referencias	RF 5
Precondiciones	El editor debe haber accedido al sistema.
Pos-condiciones	El editor importa todos los archivos que el editor escogió.
Flujo Normal de Eventos	

Capítulo 3 : *Análisis del Sistema*

Acción del Actor	Respuesta del Sistema
1. El editor selecciona la opción “Vista” (A).	
2. El editor selecciona la opción “Procesamiento multipista” (B).	
3. El editor selecciona la opción “Importar”.	
4. El editor selecciona el fichero de audio que desea abrir.	4.1 El sistema convierte el fichero a .wav si no se encuentran en esa extensión. 4.2 El sistema abre el fichero de audio.

Prototipo de Interfaz



Tabla 8: Descripción del caso de uso Permitir procesamiento multipista.

El **anexo 2** muestra las clases del análisis que intervienen en la realización de los casos de uso anteriormente descritos, así como los diagramas de interacción de estas clases para producir un resultado.

3.7 CONCLUSIONES PARCIALES.

El Modelo de Dominio realizado a partir de los procesos identificados permitió conocer todos los términos y conceptos presentes en el entorno, estos fueron descritos y especificados en un glosario de términos y representados en un diagrama de clases, a través del cual se pudo comprender mejor el funcionamiento e interrelación de los mismos.

Mediante la conformación del Diagrama de Casos de Uso del Sistema se pudo traducir los requerimientos funcionales en interacciones de los actores con el sistema, lo que permitió obtener una visión de cómo será la interacción de los roles identificados con las funcionalidades que brindará el sistema. La descripción detallada del flujo de eventos de cada caso de uso en un lenguaje formal y entendible permitió comprender en detalle los procesos incluidos en el cumplimiento de los objetivos trazados para cada caso de uso.

Capítulo 4 : Construcción de la Solución Propuesta

Capítulo 4. Construcción de la Solución Propuesta

4.1 INTRODUCCIÓN.

En este capítulo se abordan temas fundamentales para la construcción del sistema propuesto, como los diagramas de clases del diseño que constituyen el punto de partida para el flujo de trabajo de implementación. Se describen además los patrones de diseño y de arquitectura utilizados en la presente investigación.

En la elaboración del editor de audio se realizó un diseño lo más claro posible para su posterior programación. Además se tuvo muy en cuenta la escalabilidad puesto que esta es una primera versión que estará sujeta a cambios para mejorar su eficiencia y posibilitar mayor funcionalidad al usuario final.

4.2 PATRONES.

A grandes rasgos se puede decir que un patrón es un modelo a seguir para realizar una actividad en específico. Estos surgen con la experiencia de los seres humanos de tratar de lograr ciertos objetivos. Por lo tanto capturan las experiencias existentes y probadas para promover buenas prácticas.

Los patrones:

- Son una abstracción de “problema – solución”.
- Identifican y especifican abstracciones de niveles más altos que componentes o clases individuales.
- Proporcionan vocabulario y entendimiento común.
- Se ocupan de problemas recurrentes.

4.2.1 Patrones de Diseño.

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de

Capítulo 4 : Construcción de la Solución Propuesta

diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Para un mejor diseño de la aplicación se tuvo en cuenta varios patrones de diseño, una vez aplicados estos patrones los diseños se hicieron más flexibles, modulares y reutilizables.

Para la asignación de responsabilidades se tuvieron en cuenta algunos de los Patrones GRASP. Según Booch y Rumbaugh, autores del libro “El proceso unificado de desarrollo de software”, la responsabilidad es un “contrato u obligación de un tipo de clase”. Los patrones de asignación de responsabilidades usados para el desarrollo de la aplicación fueron el patrón Experto, el cual plantea que se debe asignar la responsabilidad al experto en información que en este caso sería la clase que cuenta con la información necesaria para cumplir la responsabilidad. También se hizo uso del patrón Creador, el cual plantea la necesidad de asignarle a una clase la responsabilidad de crear una instancia de otra clase siempre y cuando agregue los objetos de la clase, los contenga, registre las instancias de estos objetos y los utilice específicamente. (44)

4.2.2 Patrones de Arquitectura.

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución.

4.2.3 Arquitectura en Capas.

La arquitectura en capas definida según Garlan y Shaw es una organización jerárquica tal que cada capa proporciona servicios a la capa inmediata superior y se sirve de las prestaciones que le brinda la inmediata inferior. (45) Las ventajas del estilo en capas son muchas, primero que nada, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los programadores la participación de un problema complejo en una secuencia de pasos incrementales. En segundo lugar admite optimizaciones y refinamientos. Proporciona además amplia reutilización, lo que se convierte en uno de los fuertes de esta arquitectura.

Capítulo 4 : Construcción de la Solución Propuesta

El subsistema de edición de audio ha sido diseñado teniendo en cuenta el patrón en capas. Este establece la capa de presentación, la capa de lógica del negocio y la capa de acceso a datos.

4.3 DIAGRAMAS DE CLASES DEL DISEÑO.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación. En el modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota la realización de casos de uso del diseño. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa.

El propósito del diseño constituye transformar los requerimientos en lo que será el sistema y evolucionar una arquitectura robusta para este. (46)

Se presenta a continuación el diagrama de clases del diseño que muestra las especificaciones de las clases del software y de las interfaces del mismo; representando sus métodos, atributos y dependencias. Para un mejor entendimiento se han mostrado las clases por capas y posteriormente la interacción general de las mismas.

Capítulo 4 : Construcción de la Solución Propuesta

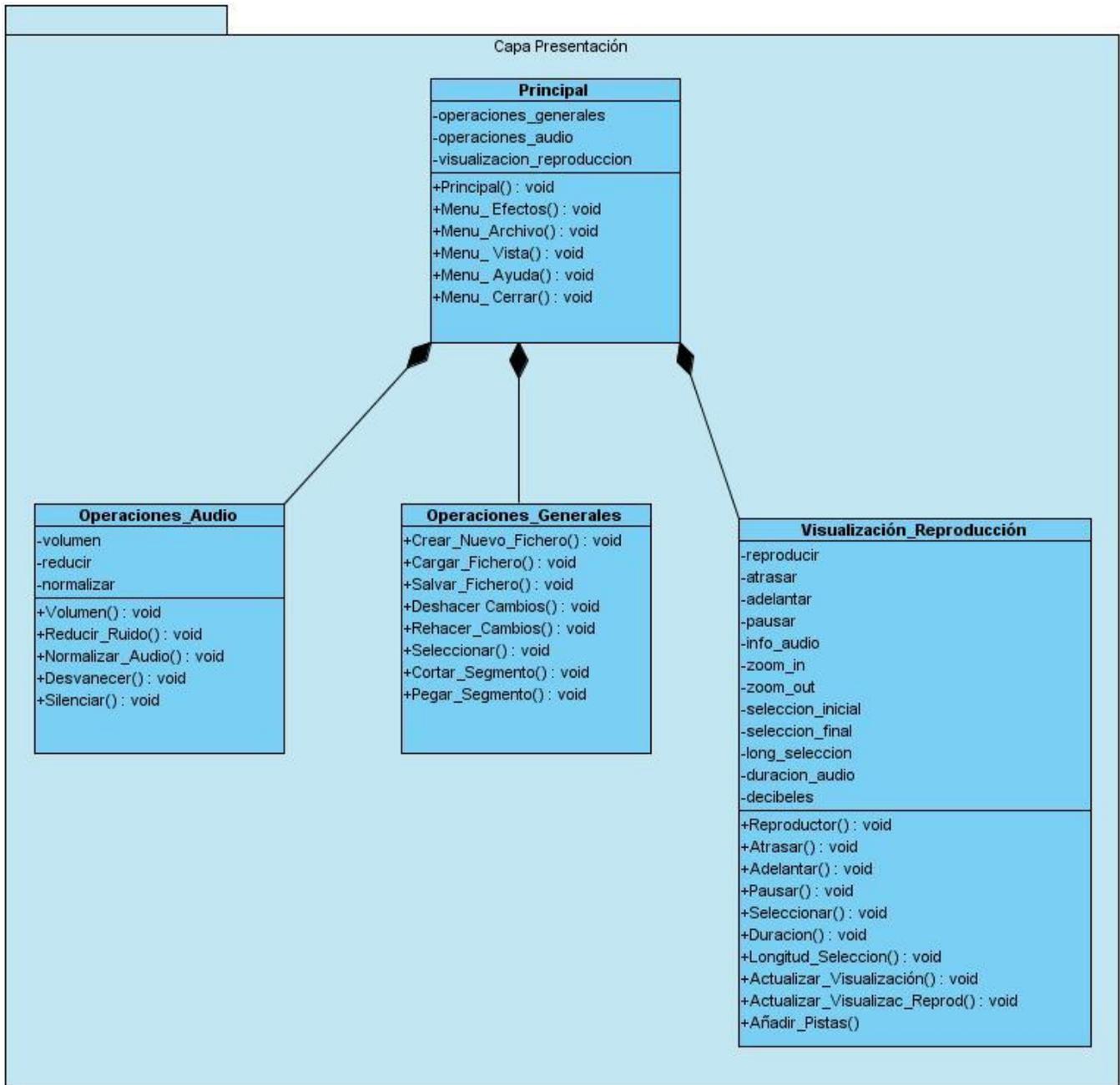


Figura 8: Capa de Presentación, Diseño de Clases

Capítulo 4 : Construcción de la Solución Propuesta



Figura 9: Capa Lógica del Negocio, Diseño de Clases

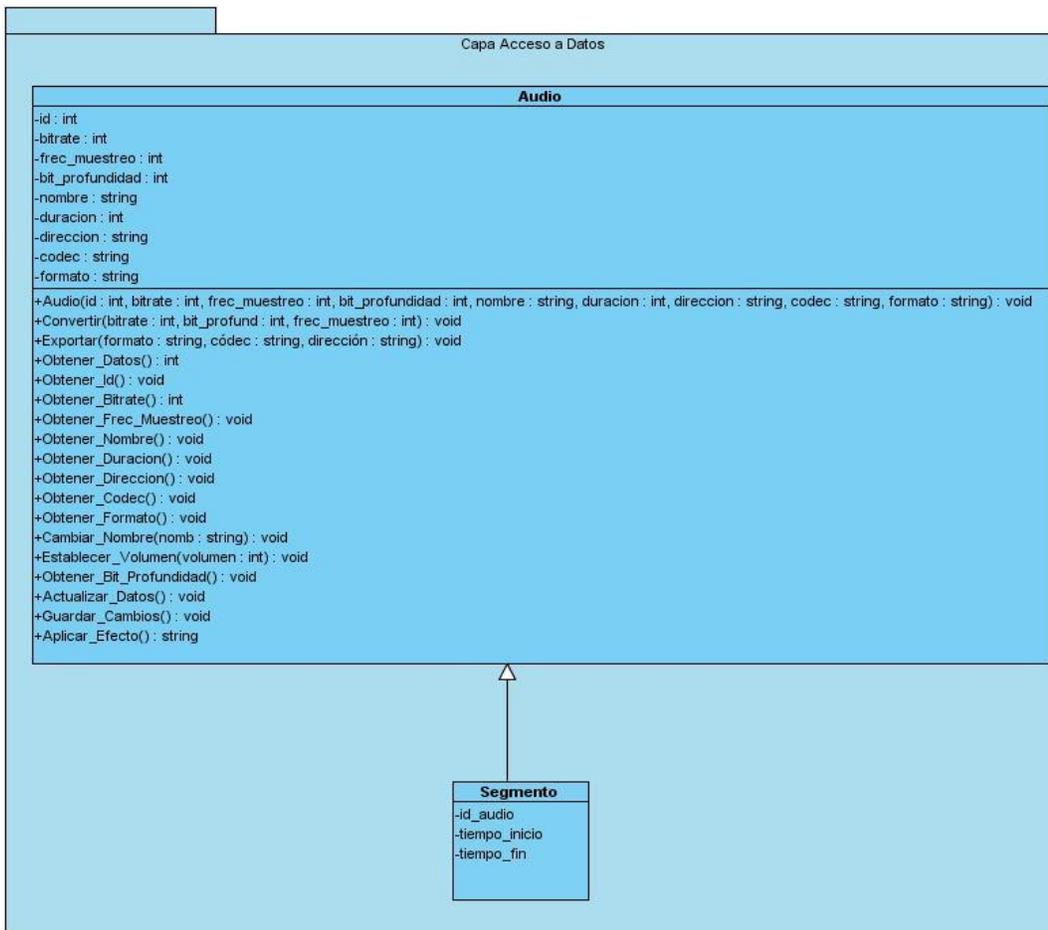


Figura 10: Capa Acceso a Datos, Diseño de Clases

Capítulo 4 : Construcción de la Solución Propuesta

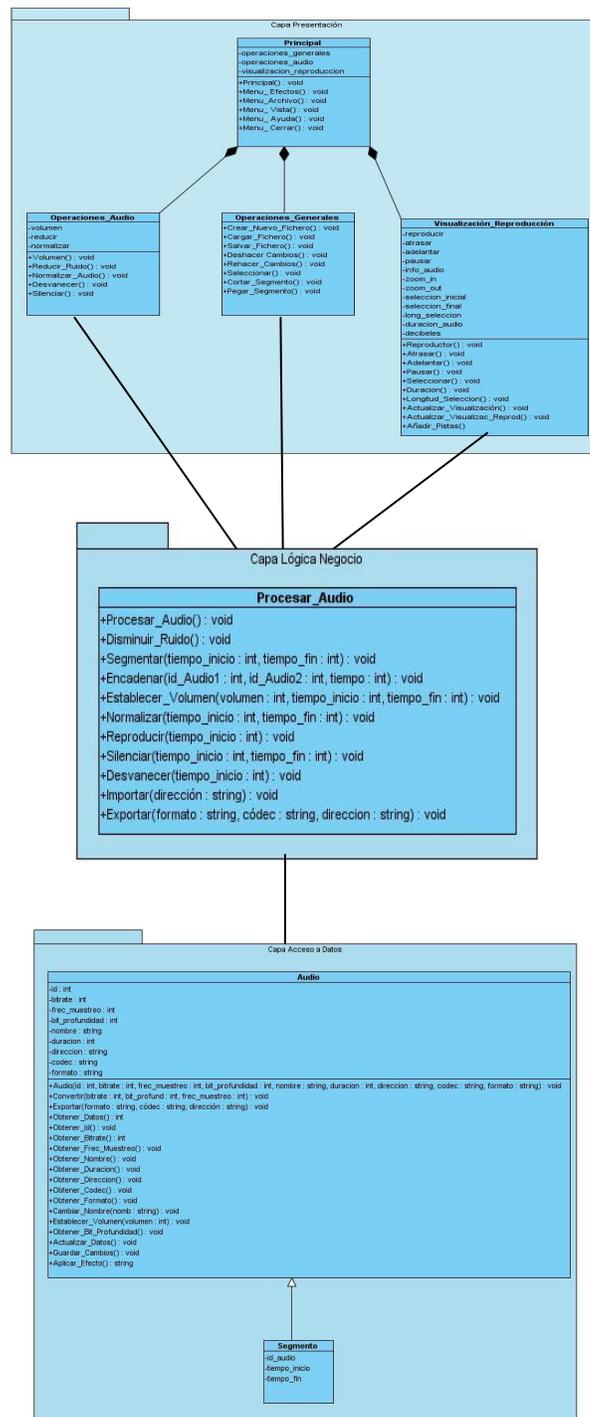


Figura 11: Interacción General de las Clases.

Capítulo 4 : Construcción de la Solución Propuesta

4.4 PRINCIPIOS DE DISEÑO.

4.4.1 Estándares de interfaz de aplicación.

Uno de los procesos más importantes en el desarrollo de software es el diseño de la interfaz de una aplicación, pues esta es la manera de comunicación del sistema con los usuarios finales. Si se desea lograr un alto nivel de aceptación la interfaz de usuario se debe construir teniendo en cuenta los 2 factores más importantes para lograr el éxito:

1. **Visibilidad:** Los componentes esenciales deben ser visibles todo el tiempo, de esta forma el usuario podrá realizar las tareas básicas de forma intuitiva y fácil, por lo que hay que cuidar la navegabilidad.
2. **Comprensión intuitiva:** Todo lo que se muestre en el subsistema debe representar una acción intuitiva, así el usuario podrá utilizar todas las funcionalidades de la aplicación sin necesidad de mucha capacitación, esto ayudará en todos los casos a reducir costos a la entidad en que es implantado el sistema.

Teniendo en cuenta estos elementos la interfaz gráfica de la aplicación fue concebida con un ambiente sencillo, amigable y de navegación fácil e intuitiva para el usuario. Los colores han sido convenientemente utilizados dada la funcionalidad y objetivo del subsistema, siendo claros y sobrios en la mayor parte de la aplicación logrando una vista agradable a los usuarios.

4.5 DIAGRAMAS DE INTERACCIÓN.

Los diagramas de interacción muestran una interacción y valga la redundancia, que consiste de un conjunto de objetos y sus relaciones, incluyendo los mensajes que puedan ser realizados entre ellos. Son importantes para modelar los aspectos dinámicos de un sistema y para construir sistemas ejecutables a través de ingeniería hacia adelante e ingeniería inversa.

Comúnmente contienen:

- Objetos

Capítulo 4 : Construcción de la Solución Propuesta

- Enlaces
- Mensajes

Pueden servir para visualizar, especificar, construir y documentar los aspectos dinámicos de una sociedad particular de objetos.

Los diagramas de interacción están conformados por los diagramas de secuencia y los diagramas de colaboración.

4.5.1 Diagramas de Secuencia.

Los diagramas de secuencia destacan la ordenación temporal de los mensajes. Un diagrama de secuencia se forma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, a lo largo del eje X. Normalmente, se coloca a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la derecha. A continuación, se colocan los mensajes que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo. Esto ofrece al lector una señal visual clara del flujo de control a lo largo del tiempo.

4.5.2 Diagramas de Colaboración.

Los diagramas de colaboración (**Anexo 3**) destacan la organización de los objetos que participan en una interacción. Un diagrama de colaboración se construye colocando en primer lugar los objetos que participan en la colaboración como nodos del grafo. A continuación se representan los enlaces que conectan esos objetos como arcos del grafo. Por último, estos enlaces se adornan con los mensajes que envían y reciben los objetos.

4.6 CONCLUSIONES PARCIALES.

En el actual capítulo se mostró el diseño detallado de la aplicación, así como la arquitectura de la misma basada en patrones de arquitectura y diseño. Con el objetivo de lograr una aplicación cómoda, intuitiva, amigable y de fácil usabilidad se han tenido en cuenta algunos principios de diseño para lograr un elevado nivel de aceptación del software a desarrollar. Se ha realizado una breve descripción de los diagramas de

Capítulo 4 : Construcción de la Solución Propuesta

interacción y se ha podido observar su importancia y uso en la modelación de un flujo particular de un caso de uso.

Conclusiones Generales

Conclusiones Generales

Con la realización del presente trabajo se ha dado cumplimiento a los objetivos trazados en la actual investigación:

- Se caracterizaron los procesos de edición de audio.
- Se caracterizaron los procesos concebidos en el proyecto Captura y Catalogación de Medias relacionados con la edición de audio.
- Se seleccionaron los formatos y códecs de audio apropiados para la solución propuesta.
- Se establecieron las bases teóricas y tecnológicas de la solución propuesta.
- Se realizó el análisis y diseño del subsistema de edición de audio.

Los resultados de la presente investigación serán de gran utilidad para las entidades donde sea necesaria la edición de audio. El Departamento de Señales Digitales contará con un nuevo producto que además de estar incluido en el proyecto Captura y Catalogación de Medias podrá ser integrable a otras soluciones comerciales desarrolladas por el propio DSD. Asimismo el editor podrá comercializarse de forma independiente.

Recomendaciones

Recomendaciones

Al concluir el desarrollo del presente trabajo se recomienda:

- Continuar el desarrollo de la investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la aplicación.
- Implementar el subsistema de edición de audio haciendo uso de la solución propuesta.
- Se recomienda el uso del subsistema en las soluciones comerciales y no comerciales del Departamento de Señales Digitales que lo requieran.

Referencias Bibliográficas

Referencias Bibliográficas

1. *Sociedad de la Información*. **Ruiz Felipe, José Ángel**. 17, 2009. ISSN: 1578-326x.
2. The New York Times. *The New York Times*. [En línea] [Citado el: 7 de 11 de 2009.] <http://www.nytimes.com/2008/03/27/arts/27soun.html>.
3. **Lucci, Héctor**. Todo Tango. *Crónicas*. [En línea] [Citado el: 10 de 11 de 2009.] <http://www.todotango.com/spanish/biblioteca/cronicas/fonovsgra.asp>.
4. Institution of Engineering and Technology. *IET Digital Library*. [En línea] [Citado el: 7 de 11 de 2009.] <http://www.ietdl.org/>.
5. **Arroyo, Rosalía**. Audio Digital, 30 años de historia. *NetMediaEurope Spain S.L.* [En línea] [Citado el: 6 de 11 de 2009.] http://www.itespresso.es/es/vnunet/report/2009/01/26/audio_digital__30_anos_de_historia/4.
6. *Indaganda*. [En línea] [Citado el: 8 de 11 de 2009.] <http://www.indaganda.com/definir/edición>.
7. *Elsevier Science*. USA : Dictionary of Video and Television Technology , 2002. ISBN 1-878707-99-X.
8. digitalfotored. *¿Como se convierte el sonido en digital?* [En línea] 2005. [Citado el: 22 de 11 de 2009.] <http://www.digitalfotored.com/videodigital/audiodigital.htm>.
9. W3C. [En línea] 31 de 7 de 2001. [Citado el: 8 de 11 de 2009.] <http://www.w3.org/WAI/UA/WD-UAAG10-20010731/glossary.html>.
10. **Kirn, Peter**. *Real World Digital Audio*. Berkeley : Peachpit Press : pág. 640, 2005. ISBN-10: 0-321-30460-8.
11. **Jack, Keith y Tsatsulin, Vladimir**. *Dictionary of Video and Television Technology*. Estados Unidos de América : Elsevier Science, 2002. ISBN 1-878707-99-X.
12. Microsoft. *Windows Media Audio Codecs*. [En línea] [Citado el: 16 de 01 de 2010.] <http://www.microsoft.com/windows/windowsmedia/forpros/codecs/audio.aspx>.
13. THOMSON images & beyond. *MP3 Playing Everywhere. The History of mp3*. [En línea] <http://www.mp3licensing.com/mp3/history.html>.
14. Xiph open source community. *vorvis.com*. *vorvis.com*. [En línea] [Citado el: 19 de 01 de 2010.] <http://www.vorbis.com/faq/#wha>.