

Tipo de artículo:

Temática:

Recibido: dd/mm/aa | Aceptado: dd/mm/aa | Publicado: dd/mm/aa

# Diseño del subsistema de edición de audio para el proyecto captura y catalogación de medias

## *Subsystem design for audio editing for the capture and cataloging media project*

Annies Cedeño López<sup>1</sup>, Yuneisy Barrios Pérez<sup>1</sup>

<sup>1</sup> Departamento de Aplicaciones Informáticas. Facultad Regional “Mártires de Artemisa”. Universidad de las Ciencias Informáticas, Batey Lincoln, Consejo Popular Abraham Lincoln, Artemisa. Cuba

---

### 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 requieren 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. El subsistema de edición de audio usa tecnologías libres siguiendo las actuales tendencias en el mundo.

**Palabras clave:** audio, edición, 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. The subsystem of audio editing uses the free technologies following the world current tendencies.*

**Keywords:** audio, edition, files.

## Introducción

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 Medias (CCM) perteneciente al Departamento de Señales Digitales (DSD) de la Facultad 9, cuyo objetivo es almacenar ficheros de audio y video para catalogarlos y posteriormente recuperarlos. La obtención de los ficheros de audio se puede realizar de diferentes maneras, ya sean grabaciones realizadas por el propio sistema o grabaciones procedentes de otro origen. 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 del archivo. Actualmente el proyecto CCM 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. Un sistema de edición de audio resolvería la problemática descrita anteriormente y constituiría un producto de valor agregado al proyecto de Captura y Catalogación de Medias. Por lo expuesto 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.

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 digital se puede definir como el resultado de la codificación digital (representada por ceros y unos) de una onda sonora o de una señal analógica. Actualmente el hombre necesita modificar el sonido para diversos fines, como pueden ser, 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. Que 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. ¿Qué es un editor de sonido? Es aquella aplicación informática usada para manipular audio digital.

### 1. 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 conversor 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 conversor llamado digital-análogo (DAC).

#### 1.1 Calidad del sonido digital

Con el objetivo de reducir el tamaño de los archivos de audio y lograr 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.

## 2. 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.

## 3. 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.

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.

## 4. Proceso de edición de audio. Uso de formatos y códec

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. 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.

## 5. Herramientas utilizadas

Debido a que el subsistema de edición de audio constituye un producto de valor agregado al proyecto de CCM se proponen las herramientas y la metodología usada en dicho proyecto con el objetivo de que ambos productos se complementen y de ese modo puedan ser integrados correctamente.

Se decide utilizar Java como lenguaje de programación, Netbeans como entorno de desarrollo integrado, Visual Paradigm como herramienta Case y RUP como metodología de desarrollo de software. Fue realizado un estudio de las herramientas que en la actualidad resuelven los procesos de edición de audio, entre las que se encuentran: Java Sound API, Java Media Framework (JMF), JSyn (Java síntesis), Ffmpeg.

Luego de analizar las características de las herramientas antes mencionadas se decide utilizar Java Sound API (Interfaz de programación de aplicaciones de sonido en Java) y Ffmpeg como herramientas de apoyo, 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 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. Se muestran a continuación las principales características de estas dos herramientas.

**5.1 FMPEG:** 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.

**5.2 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. Esta API se compone de 4 paquetes: 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.

## 6. Análisis del sistema

En este apartado se podrá contar con una breve 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 presenta el diagrama de casos de uso del sistema.

### 6.1 Modelo de dominio

El subsistema de edición de audio está concebido como un producto que puede ser personalizado o comercializado a cualquier cliente o entidad interesada, es decir, que no presenta clientes definidos ni procesos específicos a desarrollar; por lo tanto se decide realizar un modelo de dominio, el cual contiene los conceptos fundamentales del entorno en que trabaja el sistema.

### 6.2 Conceptos principales del modelo de dominio

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

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

*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.

### 6.3 Requerimientos funcionales

**RF 1. Editar Audio.** El sistema debe 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.

**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.

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.

### 6.4 Requerimientos no funcionales

*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.

**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. Además 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.

### 6.5 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 (7). Debido a las características particulares del sistema solamente se tiene un actor, el editor. Los casos de usos que han definidos se relacionan exclusivamente con este actor.

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

### 6.6 Diagrama de casos de uso del sistema

A partir de los requerimientos funcionales que fueron identificados se conformó el Diagrama de Casos de Uso del Sistema (DCUS), el cual muestra el comportamiento del sistema mediante la relación entre el actor y los CU.

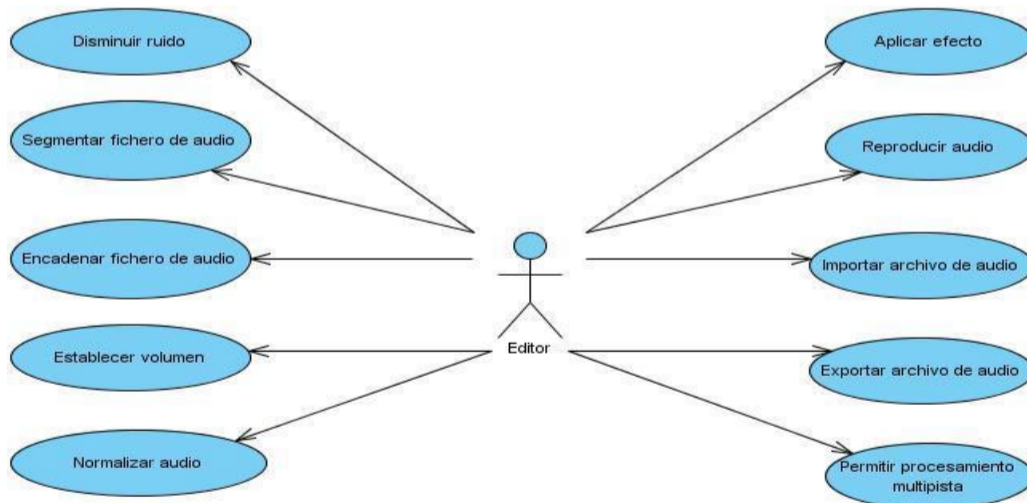


Figura 1. Diagrama de Casos de Uso del Sistema.

A continuación se muestra el prototipo de interfaz de usuario que ha sido diseñado para el editor de audio propuesto.



Figura 2. Prototipo de interfaz de usuario del editor de audio.

## 7. Construcción de la solución propuesta

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. Ejemplo, en el CU Reproducir audio, donde la clase Audio es la que presenta toda la información necesaria para que las demás métodos puedan funcionar de manera correcta. 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. Específicamente la clase procesar Audio crea los objetos de tipo Audio, pues es la que contiene la información necesaria para poder instanciarlos. El patrón de arquitectura usado en el subsistema de edición de audio ha sido el patrón en capas, el cual establece la capa de presentación, la capa de lógica del negocio y la capa de acceso a datos.



Figura 3. Patrón en Capas.

En la investigación fueron descritos los diagramas de interacción pudiendo observar su importancia y uso en la modelación de un flujo particular de un caso de uso. Fueron modelados específicamente los diagramas de colaboración.

## Conclusiones

Con la realización del presente trabajo se ha dado cumplimiento a los objetivos trazados en la actual investigación tales como, la caracterizaron los procesos de edición de audio así como los procesos concebidos en el proyecto Captura y Catalogación de Medias relacionadas con la edición de audio y se seleccionaron los formatos y códec de audio apropiados para la solución propuesta. Además de establecieron las bases teóricas y tecnológicas de la solución propuesta. Por último se realizó el análisis y diseño del subsistema de edición de audio. Todos estos resultados 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.

## Referencias

- THOMSON images & beyond. MP3 Playing Everywhere. The History of mp3. [en línea] Disponible en: [\[http://www.mp3licensing.com/mp3/history.html\]](http://www.mp3licensing.com/mp3/history.html).
- Xiph open source community. vorbis.com. vorbis.com. [en línea] [Consultado el: 19 de enero de 2010]. Disponible en: [\[http://www.vorbis.com/faq/#wha\]](http://www.vorbis.com/faq/#wha).
- Apple Inc. Apple. Apple. [En línea] [Citado el: 19 de enero de 2010]. Disponible en: [\[http://www.apple.com/quicktime/technologies/aac\]](http://www.apple.com/quicktime/technologies/aac).
- Alpha Internet Consulting LLC. Fraunhofer MP3 Encoders. [en línea] 2000. [Consultado el: 20 de enero de 2010]. Disponible en: [\[http://www.mp3-converter.com/encoders/fraunhofer\\_encoders.html\]](http://www.mp3-converter.com/encoders/fraunhofer_encoders.html).
- SOFTONIC. K-Lite Codec Pack. [en línea] [Consultado el: 18 de noviembre de 2009]. Disponible en: [\[http://k-lite-codec-pack.softonic.com\]](http://k-lite-codec-pack.softonic.com).
- Adobe Systems Incorporated. Guía del usuario de Adobe® Audition® 2.0 para Windows®. EE.UU: s.n., 2005.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. El proceso unificado de desarrollo de software. Madrid: Pearson Education S.A, 2000. ISBN: 84-7829-036-2.