



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica

## **USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL**

**Kevin Francisco Hernández Ramírez**

Asesorado por el Phd. Juan Carlos Córdova Zeceña

Guatemala, junio de 2011



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE  
AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**KEVIN FRANCISCO HERNÁNDEZ RAMÍREZ**

ASESORADO POR EL Phd. JUAN CARLOS CÓRDOVA ZECEÑA

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRÓNICO**

GUATEMALA, JUNIO DE 2011



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Romeo Neftalí López
EXAMINADOR	Ing. José Anibal Silva
EXAMINADORA	Inga. María Magdalena Puente
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas



## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha noviembre de 2009.

---

**Kevin Francisco Hernández Ramírez**

Guatemala, 17 de enero de 2011

Ing. Carlos Guzmán  
Coordinador del Área de Electrónica  
Escuela de Ingeniería Mecánica-Eléctrica  
Facultad de Ingeniería  
USAC

Estimado Ing. Guzmán.

El 10 de noviembre de 2009, suscribiendo la carta de aceptación correspondiente, accedí a asesorar el trabajo de graduación del estudiante *Kevin Francisco Hernández Ramírez*, carné No. 2005-11774, mismo que se titula *Uso de ondeletes en el indexado de segmentos de audio con aplicaciones en recomposición musical*.

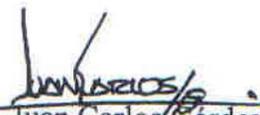
Dicho trabajo de graduación surgió del deseo del Sr. Hernández de realizar su tesis sobre el tema general de ondeletes (en inglés, wavelets) y de una idea de mi concepción sobre la utilidad que estas estructuras matemáticas podrían tener en el campo de las comunicaciones y, en particular, señales de audio y sus representaciones alternas. Específicamente estaba interesado en el símil acústico de mosaicos de imágenes, i.e., la recreación o recomposición de piezas musicales con fragmentos seleccionados de diversos orígenes.

La idea le resultó interesante al Sr. Hernández y, a decir verdad, debo agradecerle que tuviera la confianza de abordar un tema poco explorado y (en ese momento) de solución incierta, bajo mi tutelaje.

Luego de trabajar en las particularidades y dificultades del tema, y de realizar diversos experimentos relacionados, el Sr. Hernández ha producido una implementación apropiada de mi idea de recomposición musical usando como base ondeletes de la familia de Daubechies y ha plasmado el trabajo realizado en forma escrita.

He tenido oportunidad de guiar muy de cerca este proceso y puedo dar fe que el producto final efectivamente demuestra la capacidad del Sr. Hernández de aplicar los conocimientos que adquirió en la carrera de *Ingeniería Electrónica*, de adquirir nuevos conocimientos y habilidades para la resolución de un problema específico y de atender las indicaciones que se le hicieron como parte de la asesoría. De hecho felicito al Sr. Hernández por su trabajo, persistencia y entusiasmo.

En consecuencia a lo anteriormente expuesto informo a la *Escuela de Ingeniería Mecánica-Eléctrica* que doy por aprobado el trabajo de graduación del estudiante *Kevin Francisco Hernández Ramírez* titulado *Uso de ondeletes en el indexado de segmentos de audio con aplicaciones en la recomposición musical* y firmo la presente para que el interesado haga uso de la misma en la manera que corresponde a los trámites de graduación subsecuentes.

  
Dr. Juan Carlos Córdova Zeceña





Ref. EIME 11. 2011

Guatemala, 4 de FEBRERO 2011.

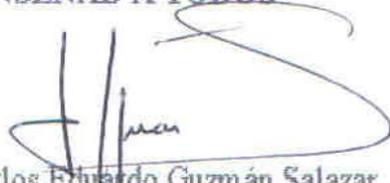
Señor Director  
Ing. Guillermo Antonio Puente Romero  
Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:  
**USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE  
AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL,**  
del estudiante, **Kevin Francisco Hernández Ramírez,** que cumple con  
los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,  
**ID Y ENSEÑAD A TODOS**

  
Ing. Carlos Eduardo Guzmán Salazar  
Coordinador de Electrónica

CEGS/sro





REF. EIME 16. 2011.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; KEVIN FRANCISCO HERNÁNDEZ RAMÍREZ titulado: USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL, procede a la autorización del mismo.

Ing. Guillermo Antonio Puente Romero

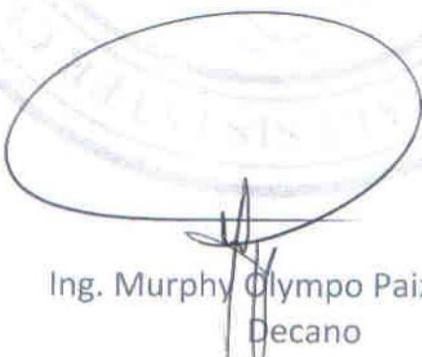


GUATEMALA, 21 DE FEBRERO 2,011.



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Civil, al Trabajo de Graduación titulado: **USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO CON APLICACIONES EN RECOMPOSICIÓN MUSICAL**, presentado por el estudiante universitario **Kevin Francisco Hernández Ramírez**, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympto Paiz Recinos  
Decano

Guatemala, 3 de junio de 2011

/gdech



## **ACTO QUE DEDICO A:**

- Dios** Por ser mi guía, mi fortaleza, la fuente de mi sabiduría durante todo este largo camino que me condujo hasta aquí.
- Mis padres** Francisco Javier Hernández Girón y Elva Patricia Ramírez Cabrera de Hernández quiénes con su apoyo incondicional, su amor y todo lo que me han brindando hicieron que fuera posible este éxito que hoy estoy alcanzando, gracias de ustedes también es este triunfo.
- Mi hermano** Daniel José Hernández Ramírez para que tome un ejemplo de mí y que algún día él también pueda estar alcanzando un éxito como este.
- Mi novia** Mónica Venegas, a quien dedico este triunfo porque con ella conviví mucho tiempo en la universidad y también en parte la sacrifiqué para lograr esto, gracias por tu amor y tu comprensión. Te amo.
- Mis abuelos** Francisco Hernández Cabrera y María Luisa Girón Golón, quiénes me han llenado de amor y de cariño durante toda mi vida.

**Familia Velázquez Hernández** A quiénes apreció mucho y esta alegría también la quiero compartir con ustedes, los quiero mucho.

**Todos mis tíos y tías** Porque de alguna u otra manera han contribuido para que el día de hoy esté aquí.

**Mis primos** Por sus consejos, sus ejemplos y su cariño.

## **AGRADECIMIENTOS A:**

**A todos mis amigos**

Por haber compartido junto a mí buenos y malos momentos, en especial a: Carlos Archila, Carlos López, Marcos Morales, Daniel Hernández, Miguel Marroquín, Alvaro García, Wellington Vázquez, Pedro Orellana,

**Al Doctor Juan Carlos  
Córdova**

Por todo su conocimiento trasladado a mi persona y por su apoyo moral y académico.



# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO	1
1.1. Objetivo	1
1.2. Indexación de segmentos de audio	2
1.2.1. Señales de audio	2
1.2.2. Segmento de audio	4
1.2.3. Indexación de segmentos de audio	5
1.2.4. Uso de Ondeletes en la indexación de segmentos	5
1.2.5. Utilización del indexado	10
1.3. Propuesta del Sistema de recomposición musical por medio de indexación de segmentos de audio	12
1.3.1. Funcionalidad	13
1.3.1.1. Búsqueda y reconocimiento	14
1.3.1.2. Sonidos originales de la base de datos	14
1.3.1.3. Base de datos	15
1.3.1.4. Canción original	15
1.3.1.5. Composición	16

1.3.2.	Interfaz gráfica para creación de base de datos	16
1.3.3.	Interfaz gráfica para composición de canciones	17
2.	CONCEPTOS GENERALES	21
2.1.	Objetivo	21
2.2.	Transformada discreta de Fourier	21
2.3.	Ondeletes	24
2.4.	Banco de filtros de reconstrucción perfecta	26
2.4.1.	Banco de filtros de Q constante	30
2.4.2.	Filtros Daubechies	32
2.5.	Método de filtros para la obtención de la transformada de Ondelete	35
2.6.	Transformada ondelete discreta	38
2.7.	Transformada Ondelete discreta inversa	43
2.8.	DWT como vector	45
2.9.	Normalización	47
2.10.	Algoritmo búsqueda de emparejamiento	48
2.11.	Matlab	51
3.	FUNCIONALIDAD	53
3.1.	Objetivos	53
3.2.	Sonidos originales de la base de datos	53
3.3.	Base de datos	60
3.3.1.	Construcción de los Ondeletes Daubechies	61
3.3.2.	Ejercicio para gráficas de energía contra componentes notables	63
3.3.2.1.	Señal original	64
3.3.2.2.	DWT	65
3.3.2.3.	Componentes notables	67
3.3.2.4.	Almacenamiento en archivo CSV	69

3.3.2.5.	Interfaz GUI para mostrar las gráficas	69
3.3.3.	Ejercicio para gráficas de energía contra ondelete	71
3.3.4.	Ejercicio para número de componentes notables	72
3.3.4.1.	Canción	73
3.3.4.2.	DWT	73
3.3.4.3.	Componentes notables	74
3.3.4.4.	IDWT	74
3.3.4.5.	Creación de archivo de audio	75
3.3.5.	Creación de la base de datos	76
3.4.	Canción original	80
3.4.1.	Canción original	81
3.4.2.	Segmentación	81
3.4.3.	DWT	81
3.4.4.	Componentes notables	82
3.5.	Búsqueda y reconocimiento	83
3.6.	Composición	85
3.7.	Interfaz gráfica	88
3.7.1.	Funcionamiento	88
CONCLUSIONES		91
RECOMENDACIONES		93
BIBLIOGRAFÍA		95
APÉNDICES		97
ANEXOS		187



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1.	Octava en un piano	4
2.	Función Haar	6
3.	Gráfica DB3	8
4.	Escalograma	9
5.	Diagrama de bloques composición	13
6.	Interfaz gráfica base de datos	16
7.	Interfaz gráfica composición	18
8.	Análisis multiresolución	25
9.	Banco de filtros de reconstrucción perfecta	26
10.	Funciones de sub-muestreo y sobre-muestreo	27
11.	Banco de filtros de Q constante	31
12.	Árbol ondelete	32
13.	Transformada ondelete inversa	44
14.	Vectores	49
15.	Histograma de base de datos	55
16.	Acercamiento de histograma	55
17.	Interfaz gráfica para la elección del tamaño del segmento	56
18.	Distribución de almacenamiento de segmentos	60
19.	Diagrama de flujo para gráficas energía contra porcentaje de componentes notables	64
20.	Interfaz para gráficas de energía contra componentes notables	69
21.	Gráfica de energía vrs. componentes notables	70
22.	Gráfica de componentes notables vrs. familia de ondeletes	72

23. Diagrama de flujo de ejercicio de número de componentes notables	73
24. Interfaz gráfica de la creación de la base de datos	77
25. Diagrama de bloques de canción original	80
26. Interfaz gráfica de recomposición	88

## **TABLAS**

I. Tipos de tonos	3
II. Ejemplo de codificación	46
III. Lista de canciones que se utilizaron para elegir un tamaño de segmento	58
IV. Ejemplo de matriz de bloque	68
V. Componentes notables por tamaño	76

## LISTA DE SÍMBOLOS

Símbolo	Significado
<b>BW</b>	Ancho de banda de un filtro
$a_j$	Aproximación de nivel $j$
<b>CN</b>	Componentes notables
$d_j$	Detalle de nivel $j$
<b>E</b>	Energía de un segmento
<b>Au</b>	Extensión de archivo de audio
<b>Wav</b>	Extensión de archivo de audio
<b>Q</b>	Factor Q de un filtro que muestra la relación entre el ancho de banda BW y la frecuencia central $f_0$
<b>DB</b>	Familia Daubechies
$f_0$	Frecuencia central de un filtro
$\varphi$	Función de escala
$\psi$	Función de ondelete
$\uparrow 2$	Función de sobre muestreo
$\downarrow 2$	Función de sub muestreo
$x[n]$	Función de tiempo discreto (secuencia)
<b>Hz</b>	Hertz, medida de frecuencia en ciclos por segundo
<b>KHz</b>	Kilohertz, $10^3$ Hertz
<b>Matlab</b>	Laboratorio matriz ( <i>matrix laboratory</i> )
$\beta$	Producto punto del algoritmo búsqueda de emparejamiento

#	Símbolo que representa un semitono, generalmente está acompañado de una letra que representa una nota y es conocido como sostenido
B	Símbolo que representa un semitono, generalmente está acompañado de una letra que representa una nota y es conocido como b-mol
F(w)	Transformada de Fourier de la función f(t)
DWT	Transformada de ondelete discreta
DFT	Transformada discreta de Fourier
IDWT	Transformada inversa de ondelete discreta
FFT	Transformada rápida de Fourier
F(z)	Transformada z de la función f(t)
J	Valor de escala de la transformada de ondelete
K	Valor de traslación de la transformada de ondelete
$\pi$	Valor Pi, igual 3.1416

## GLOSARIO

<b>Algoritmo</b>	Conjunto de reglas secuenciales con el fin de resolver un problema.
<b>Algoritmo búsqueda por emparejamiento</b>	Algoritmo diseñado por Mallat y Zhang, utilizado para representar una señal con un subconjunto de segmentos que pertenecen a un diccionario D por medio de una sumatoria.
<b>Análisis multiresolución</b>	Herramienta vectorial utilizada por los ondeletes, para representar una señal en un nivel de resolución con la suma de señales obtenidas, en niveles inferiores.
<b>Aproximación</b>	Coefficientes, de la transformada de ondelete, obtenidos de un filtro paso bajo.
<b>Árbol ondelete</b>	Método ondelete utilizado, para encontrar la transformada de ondelete por medio de bancos de filtros.
<b>Armónico</b>	Múltiplo de la frecuencia central a la que está oscilando un sonido.

<b>Au</b>	Formato para almacenar archivos de audio, el cuál fue uno de los primeros que permitió almacenar y reproducir audio con calidad aceptable. Fue desarrollado por la empresa <i>Next Machine Sound System</i> .
<b>Banco de filtros</b>	Conjunto de filtros que separa una señal en distintas frecuencias.
<b>Componentes notables</b>	Componentes de una transformada ondelete, los cuales se escogen en base a un porcentaje de energía, y pueden representar a la señal de una manera inteligible.
<b>Convolución</b>	Operación matemática que se usa para representar la salida de un sistema lineal.
<b>Daubechies</b>	Filtros creados por Ingrid Daubechies, con características muy buenas para representar señales de audio.
<b>Detalle</b>	Coeficientes, de la transformada de ondelete, obtenidos de un filtro paso alto.
<b>DFT</b>	Representación discreta de la transformada de Fourier.
<b>Diagrama de bloques</b>	Conjunto de cuadros que muestran la secuencia de un <i>software</i> .

<b>Escalograma</b>	En inglés “ <i>Scalogram</i> ”, representación gráfica de la energía de los ondeletes.
<b>FFT</b>	Algoritmo conocido como: transformada rápida de Fourier, muy eficientes en cuanto al tiempo se refiere para encontrar la transformada discreta de Fourier.
<b>Filtro paso alto</b>	Componente electrónico, capaz de dejar pasar frecuencias altas y atenuar las bajas.
<b>Filtro paso bajo</b>	Componente electrónico, capaz de dejar pasar frecuencias bajas y atenuar las altas.
<b>Frecuencia promedio ponderada</b>	Herramienta estadística aplicada a la electrónica para saber cuál es la frecuencia promedio de un segmento.
<b>Función de escala</b>	Función que se utiliza para determinar los coeficientes de aproximación en una transformada de ondelete.
<b>Función de ondelete</b>	Función utilizada para encontrar los componentes de detalle en una transformada de ondelete.
<b>Haar</b>	Función descubierta por un matemático del mismo nombre, muy útil para la explicación de los ondeletes.
<b>Histograma</b>	Representación gráfica que se utiliza para saber con qué frecuencia aparece alguna cosa en un experimento.

<b>Indexar</b>	Caracterizar segmentos de audio con base en un índice dado.
<b>Interfaz gráfica</b>	Forma en que se presenta un <i>software</i> a una persona para que lo pueda utilizar fácilmente.
<b>Interpolación</b>	Término utilizado para comparar señales y escoger cuál es la más parecida a un pivote, en un gran diccionario de éstas.
<b>Matlab</b>	<i>Software</i> matemático, utilizado en el trabajo por su potencial manejo de vectores.
<b>MP3</b>	Archivo de audio creado en formato MPEG, cuya principal ventaja es que se reduce el peso de los archivos generados.
<b>Nivel de significancia</b>	Valor especificado de probabilidad, usado para establecer el límite de aceptación o rechazo de una hipótesis en el análisis estadístico.
<b>Nota musical</b>	Sonido determinado por una vibración cuya frecuencia fundamental es constante.
<b>Octava</b>	En escalas musicales, intervalo entre un tono musical y otro con la mitad o el doble de su frecuencia.
<b>Ondelete</b>	Neologismo derivado del francés " <i>Ondelette</i> " que en inglés se traduce " <i>Wavelet</i> ".

<b>Ortogonal</b>	Este término se refiere a 2 vectores cuyo producto punto es igual a 0, es decir son perpendiculares.
<b>Ortonormal</b>	Este término se refiere a un conjunto de vectores, cuyo producto punto es igual a 0 (ortogonales) y la norma de cada uno de sus vectores es igual a 1.
<b>Pieza musical</b>	Conjunto de notas que se mezclan para tener un sonido característico.
<b>Producto punto</b>	También llamado producto escalar, se da entre dos vectores multiplicando solamente los valores que tengan las mismas bases o índices, dando por resultado un escalar.
<b>Recomposición musical</b>	Arreglo de una melodía, conformado de un grupo de sonidos que no pertenecen al conjunto de los originales.
<b>Segmento de audio</b>	Trozo de una señal de audio.
<b>Semitono</b>	Intervalo musical más pequeño comúnmente usando en la música.
<b>Señal</b>	Cantidad física, donde se miden variaciones de amplitud con respecto a un tiempo.
<b>Señal continua</b>	Señal que se mide todo el tiempo.

<b>Señal discreta</b>	Señal que se mide cada cierto intervalo de tiempo.
<b>Sistema lineal</b>	Sistema que obedece las propiedades de escalado y superposición.
<b>Sobre muestreo</b>	Aumentar el número de muestras de una señal discreta agregando ceros entre muestra y muestra.
<b>Software</b>	Conjunto de componentes lógicos, destinados a realizar una tarea específica.
<b>Sub muestreo</b>	Disminuir el número de muestras de una señal discreta tomando una muestra cada cierto número de éstas.
<b>Tono</b>	Propiedad de los sonidos, que los clasifica como agudos o graves dependiendo su frecuencia. Frecuencia percibida de un sonido.
<b>Transformada de Fourier</b>	Herramienta matemática que utiliza como función base los senos y cosenos.
<b>Transformada de ondelete</b>	Herramienta matemática que utiliza filtros para su construcción y es muy útil para el procesamiento de señales.
<b>Vector comprimido</b>	Vector que representa a uno de mayor dimensión, utilizando sólo los componentes importantes de éste.

**Wav**

Formato para almacenar archivos de audio que fue desarrollado por *Microsoft* e IBM. Su principal desventaja, es el peso muy elevado que tienen los archivos.



## RESUMEN

El trabajo de graduación *Uso de ondeletes en el indexado de segmentos de audio con aplicaciones en recomposición musical*, surge de una idea del Dr. Juan Carlos Córdova (de aquí en adelante el asesor), se utiliza para recomponer piezas musicales con sonidos exógenos.

Los Ondeletes, por sus características de construcción, son muy buenos para representar señales en tiempo y frecuencia, lo que los hace más poderosos que la transformada de Fourier, por lo que el asesor pensaba que utilizarlos iba a reducir la complejidad computacional.

En el capítulo 1, se explica qué es una indexación de segmentos de audio, para qué sirve, cómo se piensa hacer el *software* de la recomposición musical entre otras cosas.

Para poder realizar la recomposición, se debe tener muchos conceptos, por ejemplo la transformada de Ondelete, banco de filtros de reconstrucción perfecta, etc., los cuales son desarrollados en el capítulo 2.

En el capítulo 3, se muestra cómo fue que se realizó el *software* de recomposición musical, explicando a detalle, cada una de las partes que lo conforman. Además, este capítulo se complementa con algunos ejercicios, entre los que se pueden mencionar: elección del tamaño adecuado, gráficas de energía vrs. componentes notables, gráficas de energía vrs. ondeletes y número de componentes notables. Estos ejercicios fueron hechos para reducir la complejidad del trabajo de graduación, mostrándose sus tablas en los apéndices del 1 al 3.

Al final se presentan las conclusiones obtenidas de este trabajo y algunas recomendaciones que se hacen para mejorar lo obtenido aquí, además de los programas en el apéndice 4.

## OBJETIVOS

### General

Realizar un *software*, que basado en la teoría de comunicaciones electrónicas, utilice la indexación de segmentos de audio, por medio de Ondeletes para recomponer una pieza musical con sonidos exógenos.

### Específicos

1. Hacer un estudio de los Ondeletes dirigido a señales de audio.
2. Indexar un conjunto de sonidos por medio de los Ondeletes.
3. Implementar algoritmos creados, no copiados de Matlab, que sirvan para utilizar la transformada de Ondelete en la recomposición musical.
4. Aprender a diseñar filtros que sirvan para la transformada de Ondelete.

}

## INTRODUCCIÓN

Al finalizar la carrera de ingeniería, se quería aprender acerca de la transformada Ondelete. Con esto, se buscó al asesor para pedirle su opinión y si podía asesorar un trabajo de esta índole.

Él explicó que tenía en mente un punto que era Uso de Ondeletes en el indexado de segmentos de Audio con aplicaciones en recomposición musical, pero necesitaba estar seguro si se podía lograr concretar, por lo que sugirió leer 3 documentos que se relacionaban con la transformada de Ondelete. Después de leerlos, se hizo notorio que si se podía trabajar en el tema, por lo que se inició con el trabajo de graduación.

Con el punto definido, se empezó a trabajar en la realización del protocolo del tema, para presentarlo a la escuela, por lo que el asesor sugirió investigar más acerca de los Ondeletes. Después de las revisiones respectivas y los cambios sugeridos para el protocolo de parte del asesor, se procedió a pedir la aprobación del punto, respuesta que fue devuelta positivamente el día 09-11-2009.

Al inicio se decidió en conjunto con el asesor, que se iba a utilizar el lenguaje de programación Matlab, por la familiaridad que se tenía con el mismo. Tras esto, se empezó a investigar a detalle Matlab, y a finales del 2009 se presentó un programa para leer, guardar y reproducir archivos de audio de extensión "wav".

A inicios del 2010, el asesor sugirió que se trabajara en un programa para poder definir el tamaño de segmentos mínimo, para reconocer una canción, y que el mismo se pasara a distintos compañeros para poder obtener un resultado. El programa fue muy útil ya que mostró el inicio del camino, cabe recalcar que para este programa no se tuvo mayores complicaciones y los resultados fueron muy buenos para el desarrollo del trabajo.

Al finalizar estas pruebas, el asesor sugirió otro programa, del cuál se tenía que obtener una gráfica de %energía vs. %componentes notables. Este programa fue el verdadero punto de partida para la realización del *software*, ya que para esto se necesitaba manejar temas como los filtros de los Ondeletes, segmentación, normalización, transformada de Ondelete, convolución, entre otros. Estos temas serán descritos en el capítulo 2, por su importancia en este trabajo. Para estas gráficas se utilizaron únicamente Ondeletes Daubechies.

Ya con las gráficas obtenidas, el asesor sugirió realizar otras, que estaban basadas en las anteriores, en donde pudiera definir que familia de Daubechies utilizar para la transformada. Estas gráficas consistían en representar la cantidad de componentes notables vs. la familia Daubechies, y obtener la derivada de esta gráfica. Tras obtener los resultados de este ejercicio, se enviaron al asesor para saber que estuviera de acuerdo con la base Ondelete a utilizar, y la respuesta de su parte fue positiva.

Después de los resultados anteriores, el paso siguiente era almacenar los datos de alguna forma, por lo que el asesor sugirió distintas formas y dejó que se investigara cómo poder hacerlas. Tras una exhaustiva investigación se logró almacenar datos. Ya sabiendo la forma de cómo almacenar datos en Matlab, el asesor proporcionó un conjunto de discos que eran de sonidos de la naturaleza para que se utilizaran en la recomposición.

Luego de almacenar datos, el asesor propuso que se pensara en una forma de hacer el emparejamiento entre segmentos, punto en el que se encontraron los mayores problemas.

Se pensaron distintas formas de realizarlo pero no se obtenían los resultados deseados, por lo que el asesor trabajó en eso y se ideó un algoritmo (búsqueda de emparejamiento), que lecturas futuras mostraron que ya existía.

Él explicó el algoritmo que había pensado y algunos ejercicios extras que se tenían que hacer, por ejemplo variar los tamaños de los segmentos, y la cantidad de componentes notables, que se necesitaban para entender con cierto nivel de legitimidad una canción. Con todo esto, se realizaron los programas necesarios y al cabo de un tiempo se conseguieron los resultados esperados, siendo específicos a finales de septiembre de 2010.

Ya con el *software* realizado, se procedió a redactar el trabajo final. En esta etapa el asesor revisó e hizo las correcciones necesarias para que el trabajo que se cumpliera con ciertas metas que se habían establecido al principio del trabajo.

Cabe recalcar que por interés propio del asesor y para poder asesorar mejor, él desarrolló su propia implementación en Matlab, de las ideas de la recomposición musical, mismas que sirvieron para mostrar que las ideas tenían solución.

Por ser una idea del asesor, éste es el primero de varios trabajos relacionados, por lo que se espera que el material desarrollado aquí sea de utilidad para las futuras investigaciones.



# 1. USO DE ONDELETES EN EL INDEXADO DE SEGMENTOS DE AUDIO

## 1.1. Objetivo

Los objetivos de este capítulo son:

- Explicar qué es la indexación de segmentos de audio.
- Demostrar para qué se van a utilizar los índices de audio.
- Conocer qué es lo que se piensa hacer con la indexación.
- Analizar cómo se puede realizar lo que se piensa hacer.
- Y finalmente mostrar cómo se piensa presentar en una interfaz gráfica al lector.

En el presente trabajo, se busca por medio de herramientas de la teoría de las comunicaciones electrónicas y los ondeletes (que se explicarán en la sección 1.2.4), crear un *software* capaz de recomponer una pieza musical con sonidos ajenos a ésta, por ejemplo: sonidos de la naturaleza, ruidos del mercado, etc. Este *software* no sólo deberá componer las canciones sino también ser eficiente en cuanto al tiempo se refiere. Para lograrlo se investigará acerca de la transformada de ondelete (herramienta central del trabajo), la segmentación, la indexación de segmentos, la normalización, entre otros.

Quizás surgen dudas acerca de ciertos conceptos, pero no hay por qué preocuparse, aquí se explicará cada uno de los mencionados y algunos otros que serán de utilidad para el trabajo.

## **1.2. Indexación de segmentos de audio**

Para explicar, en qué consiste la indexación de segmentos de audio, se necesita antes, conocer acerca de ciertos conceptos básicos, los que se explicarán a continuación, dándole una dirección hacia lo que estamos tratando.

### **1.2.1. Señales de audio**

Una señal, es la representación de algún fenómeno en la naturaleza, por ejemplo: el sonido de un pájaro, la imagen de una cámara, etc. Existen señales análogas (que son las que normalmente se encuentran en la naturaleza), señales de tiempo discreto (que son un número de muestras que representan a una señal análoga) y señales digitales (que se obtienen al hacer discreto el tiempo y la amplitud de una señal análoga). En la actualidad, con el uso de los procesadores se le ha dado mayor importancia a las señales discretas y digitales, por su simplicidad de manejo y su gran utilidad en el avance tecnológico.

Una señal de audio discreta, es una representación de un sonido por medio de muchas muestras igualmente espaciadas en tiempo, que cumplen con el criterio de Nyquist<sup>1</sup>. Este tipo de señal se puede caracterizar por su frecuencia fundamental, sus armónicos, su ancho de banda, su potencia, su relación señal a ruido, su valor pico, entre otros. Generalmente, éstas se encuentran en un rango de 20Hz a 20KHz, el cuál recibe el nombre de rango audible. Este espectro audible se puede dividir según su tipo de tono como se puede ver en la tabla I.

---

<sup>1</sup> TAUB, Herbert; SCHILLING, Donald. *Principles of Communication Systems*. Segunda Edición. Nueva York: Editorial McGraw Hill, 1986. P. 185.

Tabla I. Tipos de tonos

Tipo de tono	Frecuencias	Octavas
Grave	16Hz a 256Hz	Octava 1 a octava 4
Medio	256Hz a 2KHz	Octava 5 a octava 7
Agudo	2KHz a 16KHz	Octava 8 a octava 10

Fuente: elaboración propia

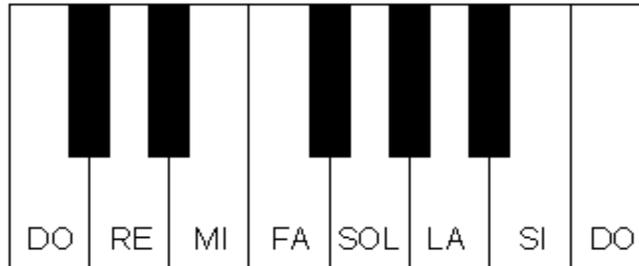
Esta tabla muestra que las bajas frecuencias pertenecen a los tonos graves, las frecuencias medias a los medios y las altas frecuencias a los agudos.

Una octava se refiere a escalas musicales, y es el intervalo entre un tono musical y otro con la mitad o el doble de su frecuencia. Están compuestas por 7 tonos (frecuencia percibida de un sonido) y 5 semitonos (intervalo musical más pequeño comúnmente usado en la música), o 12 notas. En una escala bien ajustada, la relación de frecuencias entre una nota y la siguiente es una constante igual a  $\sqrt[12]{2}$ .

Para un piano por ejemplo, una octava está conformada por las notas: DO, DO#, RE, MI b, MI, FA, FA#, SOL, SOL#, LA, SI b, y SI. Su representación se muestra en la figura 1. Las teclas blancas representan los tonos, y las negras los semitonos.

Figura 1. **Octava en un piano**

## OCTAVA



Fuente: elaboración propia

En la música clásica, muchas de las canciones están compuestas por pianos. Un piano clásico completo tiene 88 teclas, y cubre un poco más de 7 octavas. Para un piano bien ajustado, la primer tecla tiene una frecuencia central de 27,5 Hz, y la relación de frecuencias centrales entre teclas, como se mencionó arriba, de  $\sqrt[12]{2}$ . La última tecla tiene una frecuencia central de 4 186,01 Hz. Esta nos indica que una pieza musical clásica, si se utilizan las frecuencias centrales de las teclas, se encuentra en un rango de 27,5Hz a 4 186,01 Hz.

### 1.2.2. **Segmento de audio**

Un segmento de audio, es un trozo de un sonido. Éstos pueden ser obtenidos por un tiempo específico de duración o por cierto número de muestras de una señal. Generalmente el segmentar una señal sirve para hacer más sencillo su análisis, porque si es demasiado grande, al analizarla se vuelve una tarea muy difícil, tanto por el tiempo de procesamiento como por la cantidad de características que hay que tomar en cuenta.

### **1.2.3. Indexación de segmentos de audio**

La indexación de segmentos de audio trata de ordenar todos los segmentos de una forma adecuada, para hacer su búsqueda más sencilla y eficiente.

Una forma de indexar trozos de audio, es por medio de transformadas (formas de caracterizar señales por medio de la aplicación de un operador<sup>2</sup>); como bien se sabe existen distintos tipos de transformadas, por ejemplo: la transformada de Fourier, la transformada Z, la transformada de Ondelete, entre otras. Por esto y por muchas razones más, existen varias formas de ordenar los segmentos por medio de un índice derivado de una transformación.

### **1.2.4. Uso de Ondeletes en la indexación de segmentos**

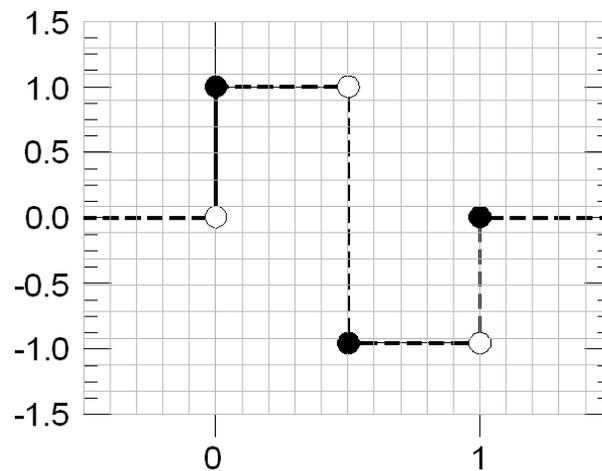
Ondelete, neologismo derivado del francés “*Ondelette*” que en inglés se traduce “*Wavelet*”; es una herramienta matemática que se utiliza en el presente trabajo, como herramienta de caracterización de señales, porque es capaz de representar eficientemente a éstas en el tiempo y la frecuencia, cómo se verá en la figura 4. Su realización se basa en un método de filtros, que se explicará en la sección 2.5.

---

<sup>2</sup> MIX, Dwight; OLEJNICZAK, Kraig. *Elements of Wavelets for Engineers and Scientists*. Estados Unidos: Willey Interscience, 2003. 236 p. ISBN: 0-471-46617-4.

Históricamente los ondeletes fueron descubiertos a principios de 1910, por un matemático alemán de nombre Alfred Haar<sup>3</sup>, quién desarrolló la función que lleva su nombre. Su gráfica se muestra en la figura 2. Ésta es una función madre, de la que se pueden derivar muchas hijas, a esto se le conoce como secuencia de funciones, sin embargo en ese tiempo Haar utilizó estas funciones únicamente para dar un ejemplo de ortonormalidad. Como se observa, es una función muy sencilla pero de trascendental utilidad para la explicación de los Ondeletes.

Figura 2. **Función Haar**



Fuente: elaboración propia

Con el pasar de los años la tecnología avanzó, y los científicos investigaron nuevas técnicas que los ayudaron a resolver problemas con el procesamiento de señales. Una técnica muy utilizada fue y continúa siendo la transformada de Fourier.

---

<sup>3</sup> RÍOS RIVAS, Manuel Alejandro. *La transformación Wavelet en el procesado digital de audio*. Facultad de ingeniería: ingeniería electrónica. Universidad de San Carlos de Guatemala. Guatemala. 2009. P. 115.

Para señales de audio, ésta tiene limitaciones, porque una señal de este tipo tiene muchos cambios de frecuencia y en ocasiones tiene transientes, características que no pueden ser detectadas adecuadamente por ésta, debido a su forma de construcción, la cual separa a la señal en ventanas de frecuencia del mismo tamaño.

La transformada de ondelete, que recibió este nombre entre los 80's y 90's, superó esta limitante utilizando filtros que varían su ancho de banda, dependiendo la frecuencia que se esté revisando, entonces los ondeletes son capaces de representar eficientemente señales en tiempo y en frecuencia, justo entonces los matemáticos y científicos descubrieron la gran utilidad de la función Haar para poder desarrollar la teoría de los ondeletes.

En la actualidad existen diferentes tipos de ondeletes, entre ellos los de Daubechies que fueron desarrollados por Ingrid Daubechies<sup>4</sup>, ella diseñó filtros continuos en la frecuencia que son máximamente planos en la banda de paso, así como en la de rechazo.

No existe una función ondelete en el tiempo definida para este tipo, como sucede con la función Haar, porque como se observa en la figura 3, gráfica de los filtros Daubechies no.3 (DB3), existen picos que muestran la complejidad de la fórmula de estos ondeletes; sin embargo, sí existe una función, que se explica en la sección 2.4.2, para la obtención de los coeficientes de los filtros.

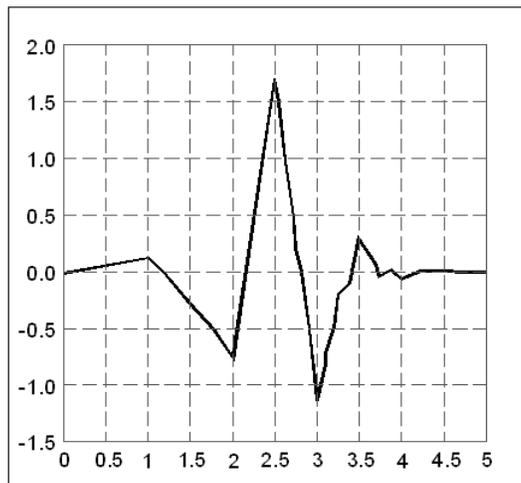
---

<sup>4</sup> RÍOS RIVAS, Manuel Alejandro. *La transformación Wavelet en el procesado digital de audio*. Facultad de ingeniería: ingeniería electrónica. Universidad de San Carlos de Guatemala. Guatemala. 2009. P. 118.

Estos filtros Daubechies son muy importantes en el análisis discreto de las transformadas de ondelete por su forma de construcción.

Se pueden construir Ondeletes de esta familia, de acuerdo a un parámetro  $p$ , el cual afecta que tan planos resultan los filtros asociados. El ondelete Haar también conocido como Daubechies no. 1, por su parámetro  $p$ , no es muy utilizado hoy en día por sus discontinuidades.

Figura 3. **Gráfica DB3**



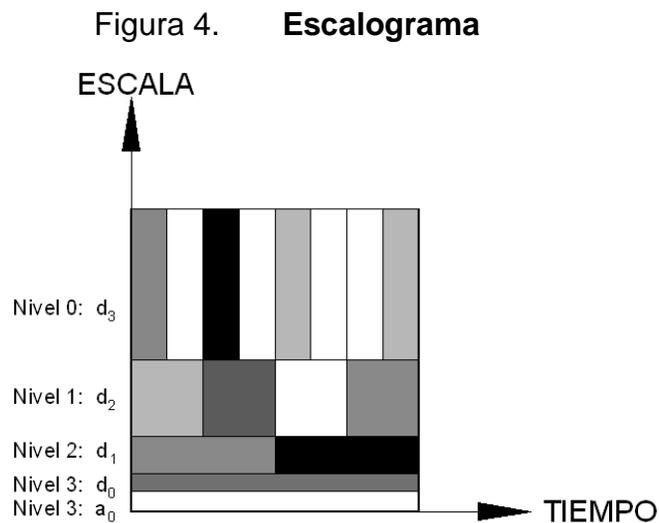
Fuente: Manuel Alejandro Ríos Rivas. Trabajo de graduación la transformación wavelet en el procesado digital de audio. Pág. 119

Los ondeletes están diseñados para capturar contenido espectral, en ventanas de tiempo relativamente cortas. Por esta razón son muy buenos para poder localizar transientes y otros fenómenos de audio, lo que los hace muy atractivo para analizar estas señales. Un ejemplo gráfico de cómo se observa la transformada Ondelete, se muestra en la figura 4, a esta representación se le denomina escalograma.

Escalograma (en inglés “*Scalogram*”), es la representación gráfica de la distribución de la energía, para una transformada de ondelete. Se forma al pasar una señal por un tipo de filtro con ventanas de tiempo, que varían dependiendo la escala, esto se explicará más a detalle en el capítulo 2.

Como se puede ver en la figura 4, existen distintos componentes, éstos se representan por un valor de escala  $j$  y un valor de traslación  $k$ . Las escalas  $j$  se representan en el eje de las “Y”, y la traslación  $k$  se representa en el eje de las “X”. Conforme se va subiendo de nivel de escala  $j$ , el número de componentes va aumentando, siendo igual a  $2^j$ .

El valor  $k$  por escala va a ir desde 0 hasta  $2^j - 1$ , los ondeletes en las escalas altas, se caracterizan muy bien en el tiempo, pero no así en la frecuencia, y a escalas bajas su caracterización en frecuencia es buena, pero en el tiempo es mala.



Fuente: elaboración propia

La transformada de ondelete, de una señal va a tener componentes positivos y/o negativos. La energía se puede obtener elevando cada componente al cuadrado y sumándolos.

En la figura 4, los cuadros que tienen más energía se muestran en negro y los blancos no aportan energía a la señal, de aquí se puede ver que se necesita cierto número de componentes para distinguir una canción. Esto se puede lograr ordenando los componentes según el porcentaje de energía que aporten a la señal de mayor a menor. Este número varía dependiendo del tamaño del segmento y el ondelete utilizado.

Se propone un ejercicio, para saber cuántos componentes se necesitan para distinguir una canción, el cual se explicará en el capítulo 3.

Existen muchas formas para indexar segmentos de audio. La forma que se utiliza en este trabajo se basa en el porcentaje de energía y se explicará a detalle en el capítulo 2.

#### **1.2.5. Utilización del indexado**

Para medir la similitud entre dos señales, se utiliza normalmente la correlación, en este trabajo se investigará cómo medir el parecido entre dos señales, por medio de la indexación de segmentos.

Como se mencionó en la sección anterior, se puede mantener cierto número de componentes de la transformada de ondelete de una señal, según el porcentaje de energía que aporten, los cuales son llamados componentes notables. Todos los componentes de la transformada de ondelete se representan por un valor de escala  $j$  y uno de traslación  $k$ , y estos valores en conjunto representan el índice del componente.

Los componentes notables tendrán un valor y un índice (valor que se podrá codificar, como se mostrará en la tabla II en la sección 2.8) que representarán a un vector mucho mayor, cómo se verá en la sección 2.8.

Como se sabe en la naturaleza existen distintos sonidos, como el canto de un pájaro, el ladrido de un perro, el sonido de la lluvia, etc., éstos de alguna manera se pueden digitalizar si se cuenta con los accesorios adecuados, por ejemplo grabadora de sonidos o una computadora con micrófono.

Ya digitalizados, los sonidos, se pueden segmentar, caracterizar por medio de ondeletes, y almacenar en una base de datos de acuerdo a una indexación específica.

Una pieza musical está compuesta de notas bajas, medias y altas y en los sonidos de la naturaleza también existen éstas. Es muy probable que una pieza de este tipo, pueda ser compuesta no por instrumentos musicales, sino más bien por sonidos de la naturaleza.

Básicamente, el *software* en el que se está pensando debe:

- Descomponer una pieza musical en segmentos
- Caracterizar cada segmento con la transformada de ondelete
- Obtener un número adecuado de componentes notables ordenados
- Buscar en la tabla de datos cuál o cuáles segmentos son los más parecidos por medio de un algoritmo.
- Ya con él o los trozos electos, buscar la forma de cómo se puede hacer la señal, lo más parecida posible a la original.
- Al final, se debe contar con dos piezas musicales, la escogida y la creada, cuyo contenido va a tener otro nivel de significancia, sería como escuchar música clásica, por ejemplo la novena sinfonía de Beethoven, con cantos de pájaros, ladridos de perros, maullidos de gatos, entre otros.

En un principio, parece una tarea muy difícil de cumplir, pero con una buena base, en cuanto a los ondeletes se refiere, un buen conocimiento del *software* a utilizar; que en este caso es Matlab versión R2008a (escogido por la familiaridad que se tiene con esta herramienta), una buena aplicación de los conceptos aprendidos en cursos de comunicaciones, esfuerzo y dedicación es posible realizarlo.

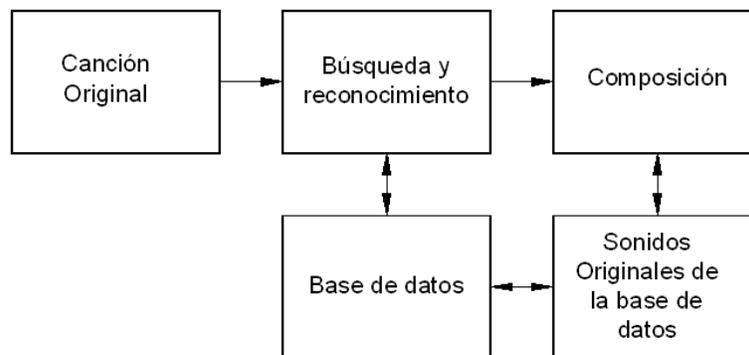
### **1.3. Propuesta del Sistema de recomposición musical por medio de indexación de segmentos de audio**

Para poder explicar cómo puede llegar a funcionar el *software*, se presenta un diagrama de bloques, junto con las interfaces gráficas con las que se piensa se va a poder manejar de una forma sencilla.

### 1.3.1. Funcionalidad

La figura 5 muestra, los bloques que se utilizarán para recomponer una pieza musical, con sonidos de la naturaleza, sustituyendo a los instrumentos musicales.

Figura 5. **Diagrama de bloques composición**



Fuente: elaboración propia

Como se puede ver en la figura 5, existen cinco bloques nombrados:

- Canción original
- Búsqueda y reconocimiento
- Composición
- Base de datos
- Sonidos originales de la base de datos

A continuación se explicará cada uno a detalle.

### 1.3.1.1. Búsqueda y reconocimiento

Es la parte central del *software*, y la que manejará toda la inteligencia, se conectará a 3 de los bloques que son: canción original, composición y base de datos, es la parte más difícil de realizar, por ser el cerebro de todo.

Su función principal se basará en escoger segmentos de una base de datos, quienes van a ser los más parecidos a uno original, todo esto lo logra, como se verá en la sección 3.5, gracias a un algoritmo llamado búsqueda de emparejamiento (en inglés *Matching Pursuit*<sup>5</sup>).

En la explicación de cada bloque se determinará de forma más específica, de qué manera se conectará hacia él y cuál será la funcionalidad de éste.

### 1.3.1.2. Sonidos originales de la base de datos

Es el espacio en memoria, en el que se almacenarán todos los segmentos de audio que serán utilizados para recomponer una canción, éstos serán escogidos de distintas fuentes y tendrán el mismo tamaño.

Para poder realizar esto, se necesitará:

- Escoger los sonidos a utilizar para la recomposición
- Dividir cada uno en pequeños trozos del mismo tamaño
- Ubicarlos en memoria identificándolos por el nombre del sonido al que pertenecen y el número de segmento

---

<sup>5</sup> MALLAT, S. G.; ZHANG, Z. "*Matching Pursuit with Time-Frequency Dictionaries*". IEEE Transacción en procesamiento de señales. 1993, vol.41, num. 12, p. 3397-3415.

Como se puede ver en la figura 5, éste tendrá conexión hacia los bloques llamados: base de datos y composición.

### **1.3.1.3. Base de datos**

En este bloque se caracterizarán todos los segmentos de la base de datos, por medio de la transformada de Ondelete, además se creará una tabla que es precisamente un diccionario de sonidos, en el que se colocarán los componentes ondeletes importantes; los números de segmento, los nombres de las fuentes originales, entre otras cosas. Aquí todo se ordenará en base a un índice.

El bloque búsqueda y reconocimiento utilizará esto para poder consultar las características de los segmentos, y para saber el nombre de los sonidos que van a servir para la composición de la canción.

### **1.3.1.4. Canción original**

En esta parte se dividirá la canción a recomponer, en trozos del mismo tamaño, y se caracterizará cada segmento con la transformada ondelete, cada segmento deberá contar con las siguientes especificaciones:

- El tamaño debe ser el mismo, que el de los trozos del bloque, base de datos
- El Ondelete utilizado para caracterizarlo, debe ser el mismo que el del bloque, características de la base de datos

El bloque búsqueda y reconocimiento, almacenará esto, para conocer qué tipo de características debe ir a buscar a la tabla que pertenecerá a características de la base de datos.

### 1.3.1.5. Composición

Éste será el encargado de componer la pieza musical final, lo hará por medio de dos bloques: búsqueda y reconocimiento y sonidos originales de la base de datos, el primero, brindará los nombres de las canciones, y los números de segmentos para que vaya a buscarlos al segundo y extraerlos para la recomposición.

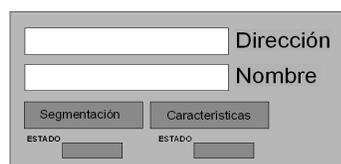
La funcionalidad de éste será recibir la información, extraerla, procesarla, ubicarla y escribir la pieza en un formato de audio digital para ser escuchada por cualquier persona que lo desee.

En el capítulo 3 se explica en detalle, cómo fue que se realizó cada uno de los bloques anteriores.

### 1.3.2. Interfaz gráfica para creación de base de datos

La figura 6, muestra la interfaz que será presentada, para que de una forma sencilla, se cree una base de datos de un sonido específico.

Figura 6. **Interfaz gráfica base de datos**



The image shows a graphical user interface with a light gray background. At the top, there are two white rectangular input fields. The first field is labeled 'Dirección' and the second is labeled 'Nombre'. Below these fields are two gray buttons with white text: 'Segmentación' on the left and 'Características' on the right. At the bottom of the interface, there are two checkboxes, each labeled 'ESTADO'.

Fuente: elaboración propia

La interfaz podrá contar con:

- Dos campos editables nombrados “Nombre” y “Dirección”
- Dos botones nombrados “Segmentación” y “Características”
- Dos cuadros de estado

Es importante recalcar que la extensión de archivos, que va a poder caracterizar, serán “au” o “wav”, por las limitaciones de Matlab.

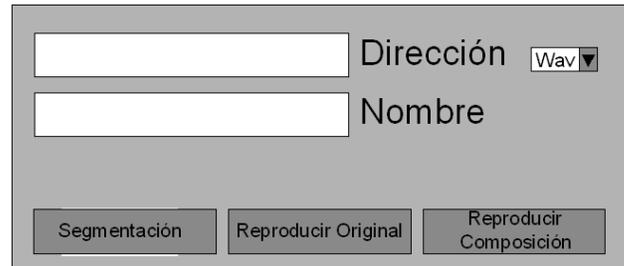
Los 2 campos editables nombrados “Dirección” y “Nombre”, servirán para escribir la dirección de donde se encuentra ubicado el sonido a ser puesto en la base de datos, y su respectivo nombre. Luego de llenar estos dos campos se deberá darle clic al botón de nombre; “Dividir en Segmentos”, debajo de este botón aparecerá un cuadro de estado, aquí aparecerá en blanco, o con las frases “En Proceso” o “Realizado”.

Cuando aparezca la palabra “Realizado” se podrá proceder al siguiente y último paso, finalmente se deberá proceder a darle clic al botón “Caracterización” que, como su nombre lo indica, caracterizará cada segmento con la transformada de ondelete, obtendrá los componentes importantes y los ubicará en el diccionario de una manera adecuada.

### **1.3.3. Interfaz gráfica para composición de canciones**

La figura 7, muestra la interfaz que utilizará el usuario para componer canciones. Debe tener la capacidad de componer una canción en unos cuantos minutos, sin necesidad de que el usuario tenga un mayor conocimiento de cómo lo hace.

Figura 7. **Interfaz gráfica composición**



The image shows a graphical user interface for audio composition. It features two text input fields at the top. The first is labeled 'Dirección' and has a dropdown menu next to it showing 'Wav'. The second is labeled 'Nombre'. Below these fields are three buttons: 'Segmentación', 'Reproducir Original', and 'Reproducir Composición'.

Fuente: elaboración propia

Como se puede ver la interfaz contará con:

- Dos campos editables denominadas: “Dirección” y “Nombre”
- Tres botones nombrados: “Crear Composición”, “Reproducir Original” y “Reproducir Composición”
- Un menú desplegable con dos campos “au” y “wav” conocido como “Extensión”
- Y un campo de informes llamado “Estado”

En el primer campo editable nombrado “Dirección”, se deberá escribir la ubicación de la canción que se quiere componer, por ejemplo: “C:\Documents and Settings\Kevin\Mis documentos\”. Como se puede notar al final de la ubicación se escribe una “\”, esto le indica al programa que es un directorio.

En el segundo campo editable llamado “Nombre”, se escribirá el nombre de la canción que se quiere componer con su respectiva extensión, por ejemplo: “a ti.wav”.

El menú desplegable, mostrará los dos formatos que podrá procesar el *software*, que serán “au” y “wav”, por defecto estará electo el “wav”. Con esto se le indicará al programa, que tipo de archivo es el que se va a componer. Como se puede notar, el *software* manejará únicamente dos tipos de archivos de audio, esto es muy importante, porque le indica al lector que un archivo “mp3”, no podrá ser compuesto por éste, a menos que se utilice un programa externo que convirtiera el archivo “.mp3” a “.wav”, o bien a “.au”.

El botón de nombre “Crear Composición”, es el que se presionará cuando se hayan llenado los campos editables y se haya elegido la extensión del archivo. Cuando se pulse este botón se ejecutarán las funciones “Canción Original”, “Búsqueda y Recomposición” y “Composición”, de la figura 4. En “Estado” se irán mostrando en qué función se encuentra el *software*, y cuando finalice aparecerá la palabra “Realizado”.

El botón llamado “Reproducir Original” se encargará de: ir a buscar al directorio designado por “Dirección”, la canción mencionada en “Nombre”, y reproducir ésta, para que la persona que está manipulando el programa pueda escuchar la canción original. Para que este botón reproduzca la canción, los dos campos editables deberán de poseer información real.

El botón denominado “Reproducir Composición”, se encargará de ir a buscar a un directorio específico, en donde se guardan las composiciones finales, elegir el archivo que se escribe en “Nombre” y reproducirlo, para que el usuario pueda escuchar la composición y comparar que tan parecida es la pieza original de la compuesta. Para que este botón reproduzca la composición deberá cumplirse dos cosas; una es que el campo editable “Nombre” tenga información real, y que en el cuadro “Estado” aparezca la palabra “Realizado”, de lo contrario no se reproducirá nada.



## 2. CONCEPTOS GENERALES

### 2.1. Objetivo

En el siguiente capítulo se explica:

- Cómo se usan ondeletes para caracterizar señales
- La conexión de los filtros con éstos
- Cómo se representa una señal por un vector
- Cómo se normaliza
- Y qué es el algoritmo búsqueda de emparejamiento

Con todo lo anterior y con los conceptos de la teoría de comunicaciones, se tendrá la base teórica para poder desarrollar los algoritmos que resuelvan el *software*, los cuáles recompondrán una canción con sonidos exógenos y que serán explicados a detalle en el capítulo 3.

### 2.2. Transformada discreta de Fourier

También nombrada DFT, es un tipo de transformada de Fourier que se utiliza para señales finitas de tiempo discreto, es el resultado de muestrear en frecuencia, la transformada de Fourier de una señal.

Una señal finita de tiempo discreto, puede convertirse en una señal periódica con período  $N$ , donde  $N$  es el número de muestras de la señal original. Para restablecer a la señal original, sólo basta tomar un período y hacer cero el resto de muestras.

Una señal periódica, es representada por una serie de Fourier, que en el dominio de la frecuencia son muestras igualmente espaciadas. Del concepto que una señal finita de tiempo discreto, se puede convertir en una señal periódica, se puede utilizar una especie de serie de Fourier para representar a la señal en la frecuencia, ésta es la transformada discreta de Fourier.

Existe una diferencia en el cálculo de la DFT con las series de Fourier que se explica a continuación; las series de Fourier, normalmente tienen una función continua en el tiempo a su entrada, la DFT tiene una función discreta en el tiempo. Si se elige la forma exponencial de la serie de Fourier una señal continua en el tiempo se representa por:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{j2\pi n t f_0} \quad (\text{Ecuación 2.1})$$

Donde  $f(t)$  es la función continua en el tiempo,  $F_n$  es la muestra en el dominio de la frecuencia, y  $f_0$  la frecuencia fundamental. El  $F_n$  se calcula de la siguiente manera:

$$F_n = f_0 \int_{-t_0/2}^{t_0/2} f(t) e^{-j2\pi n t f_0} dt \quad (\text{Ecuación 2.2})$$

La obtención de las dos ecuaciones anteriores queda fuera del alcance de este trabajo<sup>6</sup>.

De las dos ecuaciones anteriores se puede obtener la DFT, sólo hay que tomar en cuenta que el cálculo es para funciones discretas.

---

<sup>6</sup> NILSSON, James W. *Circuitos electrónicos*. Cuarta Edición. Massachusetts: Editorial Eddison-Wesley. 195, pp. 785-871.

Entonces una función discreta en el tiempo se puede representar por:

$$x[n] = \frac{1}{N} \sum_{k=-\infty}^{\infty} X[k] e^{j2\pi kn/N} \quad (\text{Ecuación 2.3})$$

Sin embargo, como es una señal periódica, la ecuación 2.3 se convierte en:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi kn/N} \quad (\text{Ecuación 2.4})$$

Donde  $x[n]$  es periódica, para obtener la señal finita de tiempo discreto, basta tomar sólo un período de ésta, es decir  $N$  muestras.

Los coeficientes de la DFT se obtienen de forma análoga a la ecuación 2.2 de la siguiente manera:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} \quad (\text{Ecuación 2.5})$$

Donde  $X[k]$  es una señal periódica con período  $N$ , para obtener la DFT de la señal finita de tiempo discreto, se toma un período  $N$  de  $X[k]$ , esto muestra que la DFT posee el mismo número de muestras que su señal de entrada.

El cálculo de la DFT con la ecuación 2.5 es muy ineficiente, por lo que se utiliza la transformada rápida de Fourier (FFT<sup>7</sup>) para su cálculo.

---

<sup>7</sup> OPPENHEIM, Allan V.; SCHAFER, Ronald W.; BUCK, John R. *Tratamiento de señales en tiempo discreto*. Segunda Edición. Massachusetts: Editorial Prentice Hall pp. 543-671.

### 2.3. Ondeletes

Como se describió en la sección 1.2.4, la transformada de ondelete, es una herramienta muy útil, que puede representar una señal en escala y tiempo.

Para explicar una transformada de este tipo, se utilizan dos funciones; la de ondelete (función madre) y la de escala (función padre). Éstas se representan por los símbolos  $\psi(t)$  y  $\varphi(t)$  respectivamente.

Las funciones de escala y ondelete pueden ser trasladadas y escaladas para formar familias de funciones<sup>8</sup>, estas están definidas así:

$$\varphi_{j,k}(t) = 2^{\frac{j}{2}}\varphi(2^j t - k) \quad (\text{Ecuación 2.6})$$

$$\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi(2^j t - k) \quad (\text{Ecuación 2.7})$$

Siendo  $j$  el factor de escala y  $k$  el factor de traslación.

Existe un análisis llamado multi-resolución<sup>9</sup>, que muestra como la suma de un espacio vectorial  $R_0$  más su complemento  $S_0$  representa el siguiente nivel de espacio vectorial  $R_1$ .

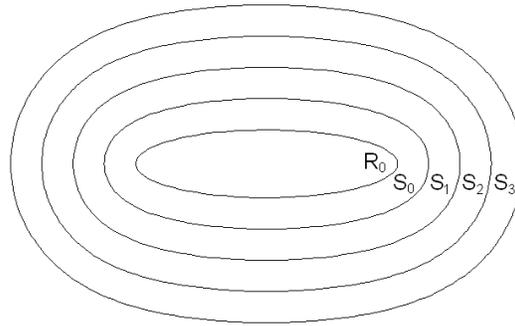
---

<sup>8</sup> MIX, Dwight F.; OLEJNICZAK, Dwight F. *Elements of Wavelets for Engineers and Scientists*. Estados Unidos: Editorial Wiley-Interscience. 2003, pp. 158-160.

<sup>9</sup> RÍOS RIVAS, Manuel Alejandro. *La transformación Wavelet en el procesado digital de audio*. Facultad de ingeniería: ingeniería electrónica. Universidad de San Carlos de Guatemala. Guatemala. 2009, pp. 177-178.

Las funciones de escala representan los espacios vectoriales  $R$  y las Ondeletes el espacio  $S$ , esto quiere decir que son complementarios. La figura 8, muestra un análisis multi-resolución con distintos niveles de escala para obtener  $R_4$ .

Figura 8. **Análisis multiresolución**



Fuente: elaboración propia

$R_4$  se obtiene de la siguiente ecuación:

$$R_4 = R_0 + S_0 + S_1 + S_2 + S_3 \quad (\text{Ecuación 2.8})$$

Utilizando las funciones de escala y ondelete, se puede formar una señal (vista como función) en un espacio vectorial de la siguiente manera:

$$f(t) = c_{j,k} \varphi_{j,k}(t) + \sum_{j=J}^{\infty} \sum_{k=0}^{\infty} d_{j,k} \psi_{j,k}(t) \quad (\text{Ecuación 2.9})$$

Siendo  $c_{j,k}$  el o los componentes obtenidos de la transformada, utilizando la función de escala como base (aproximación), y  $d_{j,k}$  el o los componentes obtenidos de la transformada, utilizando la función de ondelete como base (detalles).

Las ecuaciones 2.10 y 2.11 muestran lo anteriormente dicho.

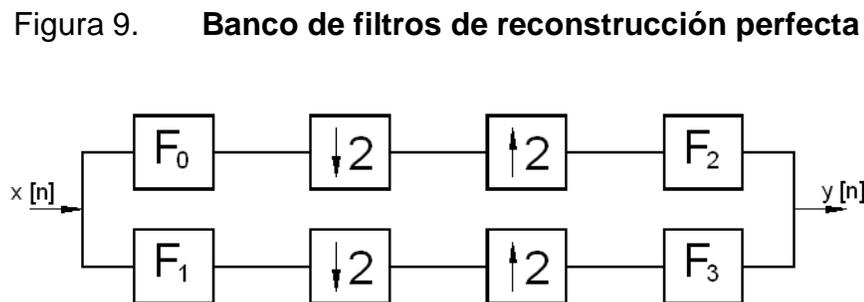
$$c_{j,k} = \int_{-\infty}^{\infty} f(t)\varphi_{j,k}(t) dt \quad (\text{Ecuación 2.10})$$

$$d_{j,k} = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t) dt \quad (\text{Ecuación 2.11})$$

Estos componentes forman la transformada ondelete, sin embargo, la mayoría de funciones de escala y de ondelete no tienen una función establecida, por tanto el cálculo se debe hacer de otra manera. Ésta se explicará a detalle en las secciones siguientes.

## 2.4. Banco de filtros de reconstrucción perfecta

Un banco de reconstrucción perfecta, es un conjunto de filtros, cuya señal de salida de la red es la misma que la señal de entrada, sólo que retrasada<sup>10</sup>. Su representación gráfica se muestra en la figura 9.



Fuente: elaboración propia

<sup>10</sup> MIX, Dwight F.; OLEJNICZAK, Dwight F. *Elements of Wavelets for Engineers and Scientists*. Estados Unidos: Editorial Wiley-Interscience. 2003, p. 186.

En la rama de arriba, los cuadros que se nombran  $F_0$  y  $F_2$ , son filtros paso bajo, y abajo los que se nombran  $F_1$  y  $F_3$ , son los filtros paso alto, entonces para revertir la acción de un filtro paso bajo se necesita otro filtro paso bajo, y para un filtro paso alto se necesita otro filtro paso alto. Tomando en cuenta solamente los filtros, la transformada Z de la salida es igual a:

$$Y(z) = X(z)\{F_0(z)F_2(z) + F_1(z)F_3(z)\} \quad (\text{Ecuación 2.12})$$

Analizando término por término, esto da por resultado:

$$Y(z) = X(z)[(a_0 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}) + (b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_nz^{-n})] \quad (\text{Ecuación 2.13})$$

Donde los valores que se encuentran entre paréntesis son los de la multiplicación entre las transformadas z de los filtros. En cada rama existen las funciones de sub-muestreo y sobre-muestreo que se representan en la figura 10.

Figura 10. **Funciones de sub-muestreo y sobre-muestreo**



Fuente: elaboración propia

Una señal cualquiera que se pasa por una red en serie con estas dos funciones, tiene la siguiente transformada Z:

$$(\downarrow 2)(\uparrow 2)x(n) \leftrightarrow \frac{1}{2}\{X(z) + X(-z)\} \quad (\text{Ecuación 2.14})$$

De la figura 10 y la ecuación 2.14, se obtiene la rama de arriba, antes de pasar la señal por el filtro  $F_2$ , lo siguiente:

$$M(z) = \frac{1}{2}\{X(z)F_0(z) + X(-z)F_0(-z)\} \quad (\text{Ecuación 2.15})$$

Y en la rama de abajo antes de pasar por el filtro  $F_3$ , se tiene:

$$N(z) = \frac{1}{2}\{X(z)F_1(z) + X(-z)F_1(-z)\} \quad (\text{Ecuación 2.16})$$

A la salida de la red se obtiene:

$$Y(z) = \frac{1}{2}\{F_0(z)F_2(z) + F_1(z)F_3(z)\}X(z) + \frac{1}{2}\{F_0(-z)F_2(z) + F_1(-z)F_3(z)\}X(-z) \quad (\text{Ecuación 2.17})$$

Si se separan en dos términos la ecuación 2.17 se obtiene:

$$R_1(z) = \frac{1}{2}X(z)\{F_0(z)F_2(z) + F_1(z)F_3(z)\} \quad (\text{Ecuación 2.18})$$

$$R_2(z) = \frac{1}{2}X(-z)\{F_0(-z)F_2(z) + F_1(-z)F_3(z)\} \quad (\text{Ecuación 2.19})$$

Para que la red represente un banco de filtros de reconstrucción perfecta, deben cumplirse 2 cosas, la primera:

$$\begin{aligned} a_0 + b_0 &= 0 \\ a_1 + b_1 &= 0 \\ &\dots \\ a_l + b_l &= 2 \\ a_{l+1} + b_{l+1} &= 0 \\ &\dots \\ a_n + b_n &= 0 \end{aligned} \quad (\text{Ecuación 2.20})$$

Donde  $l$  es el retraso que existe en la señal y esto indica que la señal de salida va a ser la señal de entrada retrasada  $l$  muestras.

Y la segunda:

$$R_2(z) = 0 \quad (\text{Ecuación 2.21})$$

Esto permite que no exista distorsión en la señal de salida, debido a que  $X(-z)$  no causaría así ninguna modificación a ésta, entonces las condiciones para que un filtro sea de reconstrucción perfecta son:

$$F_0(z)F_2(z) + F_1(z)F_3(z) = 2z^{-l} \quad (\text{Ecuación 2.22})$$

$$F_0(-z)F_2(z) + F_1(-z)F_3(z) = 0 \quad (\text{Ecuación 2.23})$$

Con estas ecuaciones se obtiene  $Y(z)$  igual a:

$$Y(z) = z^{-l}X(z) \quad (\text{Ecuación 2.24})$$

Que demuestra que la transformada  $Z$  de la señal de salida es igual a la de entrada retrasada  $l$  muestras.

Para que se cumplan las condiciones de los bancos de filtros de reconstrucción perfecta,  $F_2(z)$  y  $F_3(z)$  deben ser iguales a:

$$F_2(z) = F_1(-z) \quad (\text{Ecuación 2.25})$$

$$F_3(z) = -F_0(-z) \quad (\text{Ecuación 2.26})$$

Si se define el producto de los dos filtros pasa bajo, entonces se obtiene:

$$T_0(z) = F_0(z)F_2(z) \quad (\text{Ecuación 2.27})$$

Si se sustituyen las ecuaciones 2.25, 2.26 y 2.27 en la ecuación 2.22 se tiene:

$$T_0(z) - T_0(-z) = 2z^{-l} \quad (\text{Ecuación 2.28})$$

Existe una manera para evitar el retraso, y esta se obtiene con la siguiente ecuación:

$$T(z) = z^l T_0(z) \quad (\text{Ecuación 2.29})$$

Si se sustituye la ecuación 2.29 en la 2.28, se obtiene:

$$T(z) + T(-z) = 2 \quad (\text{Ecuación 2.30})$$

Que muestra que la suma de la transformada Z de la multiplicación de los filtros pasa bajo, tomando en cuenta el retraso, más su complemento, en un banco de filtros de reconstrucción perfecta, debe ser igual a  $2^{11}$ .

#### 2.4.1. Banco de filtros de Q constante

En filtros, Q es un factor que se obtiene así:

$$Q = \frac{BW}{f_0} \quad (\text{Ecuación 2.31})$$

---

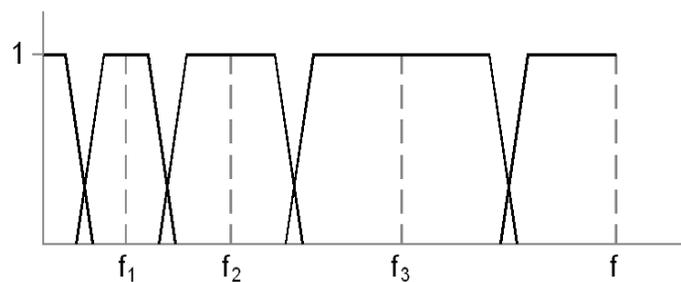
<sup>11</sup> MIX, Dwight F.; OLEJNICZAK, Dwight F. *Elements of Wavelets for Engineers and Scientists*. Estados Unidos: Editorial Wiley-Interscience. 2003, pp. 186-193.

Donde  $BW$  es el ancho de banda y  $f_0$  es la frecuencia central.

Un banco de filtros de  $Q$  constante, cuenta con filtros que su ancho de banda varía, dependiendo la frecuencia central, entre mayor sea  $f_0$  mayor va a ser  $BW$ , y viceversa. El cociente  $Q$  siempre va a ser el mismo.

En la figura 11, se muestra un banco de filtros de  $Q$  constante representado en la frecuencia.

Figura 11. **Banco de filtros de  $Q$  constante**



Fuente: elaboración propia

Como se puede ver, un banco de filtros de  $Q$  constante, cuenta con un filtro pasa bajo, un filtro pasa alto y varios filtros pasa banda.

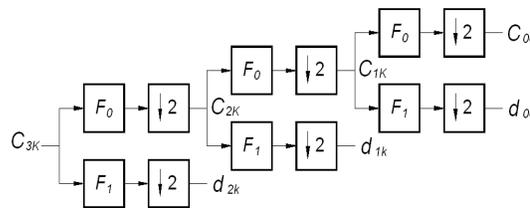
Para obtener el banco de filtros de  $Q$  constante se puede hacer de dos formas, que se explican a continuación.

- La primera, se puede lograr pasando la señal original por un filtro pasa bajo, un filtro pasa alto y varios filtros pasa banda, cada uno con la especificación de su ancho de banda y frecuencia central. Ésta es muy ineficiente porque se necesita crear demasiados filtros.

- La segunda, también nombrada árbol ondelete<sup>12</sup>, se puede lograr pasando la señal por un filtro pasa alto y un pasa bajo con el mismo ancho de banda, cubriendo cada uno la mitad de las frecuencias, se sub-muestra la señal a la salida de los filtros, luego se toma la señal del filtro pasa bajo sub-muestreada y se vuelve a pasar por los mismos filtros pasa bajo y pasa alto y así sucesivamente hasta que se quiera parar o hasta que la señal no se pueda dividir más, es decir; cuando los componentes de detalle y aproximación sean igual a 1.

La segunda forma es la utilizada para obtener la transformada de ondelete, que se explicará más adelante en éste capítulo y se muestra en la figura 12.

Figura 12. **Árbol ondelete**



Fuente: elaboración propia

## 2.4.2. Filtros Daubechies

Para obtener los coeficientes de estos filtros, la parte más importante es la multiplicación de los paso bajo del banco de reconstrucción perfecta  $T_0(z)$ , que se obtiene así:

$$T_0(z) = (1 + z^{-1})^{2p} \frac{1}{2^{2p-1}} \sum_{k=0}^{p-1} \binom{p+k-1}{k} (-1)^k z^{-(p-1)+k} \left(\frac{1-z^{-1}}{2}\right)^{2k} \quad (\text{Ecuación 2.32})$$

<sup>12</sup> RÍOS RIVAS, Manuel Alejandro. *La transformación Wavelet en el procesado digital de audio*. Facultad de ingeniería: ingeniería electrónica. Universidad de San Carlos de Guatemala. Guatemala. 2009, p. 136.

Para obtener esta ecuación se utiliza una representación de  $T_0$  con una variable  $x$  que se encuentra en un rango de 0 a 1 de la siguiente manera.

$$T(x) = 2(1-x)^p \sum_{k=0}^{p-1} \binom{p+k-1}{k} x^k \quad (\text{Ecuación 2.33})$$

Esta ecuación se obtiene de la restricción de los filtros Daubechies, son máximamente planos. Para explicar que es máximamente plano hay que tomar un ejemplo.

Existe un filtro paso bajo representado en la frecuencia que es máximamente plano, es decir su cambio con respecto a la frecuencia cerca de 0 es 0 (primera derivada con respecto a la frecuencia=0, lo que hace que el resto de derivadas sean iguales a 0), y su cambio con respecto a la frecuencia cerca de  $\pi$  es 0 (primera derivada con respecto a la frecuencia=0, lo que hace que el resto de derivadas sean iguales a 0).

En el caso de la ecuación 2.33, en lugar de la frecuencia, existe un parámetro  $x$  que va a variar de 0 a 1, entonces la primera derivada con respecto a  $x$  en 0 debe ser 0 y la primera derivada con respecto a  $x$  en 1 debe ser 0.

Para probar esto, se obtiene la primera derivada de 2.33 que es:

$$T'(x) = -2p(1-x)^{p-1} \sum_{k=0}^{p-1} k \binom{p+k-1}{k} x^{k-1} \quad (\text{Ecuación 2.34})$$

Si se evalúa la ecuación 2.34 en 0 se obtiene:

$$T'(0) = -2p(1) \sum_{k=0}^{p-1} k \binom{p+k-1}{k} 0^{k-1} = 0$$

Y si se obtiene la segunda derivada, es igual a 0 y así sucesivamente hasta la derivada  $p-1$ . Si se evalúa la ecuación 2.34 en 1 se obtiene:

$$T'(1) = -2p(1-1)^{p-1} \sum_{k=0}^{p-1} k \binom{p+k-1}{k} 1^{k-1} = 0$$

Al igual que lo que pasa al evaluar la derivada en 0, el resto de derivadas hasta  $p-1$  son iguales a 0.

Para obtener más información de cómo se obtuvo las ecuaciones 2.32 y 2.33 Gilbert Strang<sup>13</sup> publicó un documento.

Los ondeletes de Daubechies, además de ser máximamente planos deben ser ortogonales. La ortogonalidad se logra introduciendo dos nuevos conceptos: fase mínima y fase máxima.

Un filtro de fase mínima, es el que tiene la característica que todos los ceros de su transformada  $z$  se encuentran dentro del círculo unitario<sup>14</sup> al contrario un filtro de fase máxima es el que tiene la característica que todos los ceros de su transformada  $Z$ , se encuentran afuera del círculo unitario<sup>15</sup>.

---

<sup>13</sup> STRANG, Gilbert. Maxflat filters: Daubechies and Meyer formulas. Spectral Factorization [en línea]. MIT Open course ware, Course 18.327 Wavelets and Filter Banks. Disponible en Web: <<http://ocw.mit.edu/courses/mathematics/18-327-wavelets-filter-banks-and-applications-spring-2003/lecture-notes/Slides8.pdf>>.

<sup>14</sup> MIX, Dwight F.; OLEJNICZAK, Dwight F. Elements of Wavelets for Engineers and Scientists. Estados Unidos: Editorial Wiley-Interscience. 2003, p. 180.

<sup>15</sup> MIX, Dwight F.; OLEJNICZAK, Dwight F. Elements of Wavelets for Engineers and Scientists. Estados Unidos: Editorial Wiley-Interscience. 2003, p. 180.

Para la multiplicación de los filtros paso bajo  $T_0(z)$ , los filtros se dividen como de fase mínima y de fase máxima, con el mismo número de coeficientes. El primer filtro ( $F_0$ ) es de fase mínima y el segundo filtro ( $F_2$ ) es de fase máxima.

La forma en como se construyen los filtros, se muestra en la sección 3.3.1.

## 2.5. Método de filtros para la obtención de la transformada de Ondelete

Como se ha mencionado antes, la mayoría de funciones de ondelete y de escala, no pueden ser descritas fácilmente, y por esto surge el método de filtros, el cual se explica a continuación.

Este método trata de representar los coeficientes de aproximación y detalle de cierto nivel  $j$ , por medio de coeficientes de aproximación de un nivel  $j$  más alto. La parte central de esto, es la ecuación de dilatación que permite representar a  $\varphi(t)$  en términos de su versión escalada y trasladada  $k/2$ . La ecuación 2.35 muestra esta relación.

$$\varphi(t) = \sqrt{2} \sum_{k=0}^N f_0(k) \varphi(2t - k) \quad (\text{Ecuación 2.35})$$

Los valores de  $f_0(k)$  representan a los coeficientes que hacen posible que  $\varphi(t)$  pueda ser representada por su versión escalada y trasladada  $k/2$ . Éstos representan a un filtro paso bajo.

También una función ondelete  $\psi(t)$  puede ser representada por la versión escalada y trasladada  $k/2$  de la función de escala.

Esto se logra porque en el análisis multi-resolución, se muestra cómo un espacio vectorial (escala) puede ser representado por el espacio vectorial que

se encuentra en un nivel anterior (escala) más su complemento (Ondelete). La ecuación 2.36 muestra la relación entre la función ondelete y la función escala.

$$\psi(t) = \sqrt{2} \sum_{k=0}^N f_1(k) \varphi(2t - k) \quad (\text{Ecuación 2.36})$$

Como se puede ver, los coeficientes  $f_0(k)$  en esta ecuación cambian por  $f_1(k)$  que representan un filtro paso alto.

Si se sustituye  $t$  en las ecuaciones 2.35 y 2.36 por  $2^{j-1}t - k$ , que representa a los coeficientes de la escala anterior se tiene:

$$\begin{aligned} \varphi(2^{j-1}t - k) &= \sqrt{2} \sum_{p=0}^N f_0(p) \varphi(2(2^{j-1}t - k) - p) \\ &= \sqrt{2} \sum_{p=0}^N f_0(p) \varphi((2^j t - 2k) - p) \end{aligned}$$

$$\begin{aligned} \psi(2^{j-1}t - k) &= \sqrt{2} \sum_{p=0}^N f_1(p) \varphi(2(2^{j-1}t - k) - p) \\ &= \sqrt{2} \sum_{p=0}^N f_1(p) \varphi((2^j t - 2k) - p) \end{aligned}$$

Haciendo un cambio de variable  $m = 2k - p$  se tiene:

$$\varphi(2^{j-1}t - k) = \sqrt{2} \sum_{m-2k=0}^N f_0(m - 2k) \varphi(2^j t - m) \quad (\text{Ecuación 2.37})$$

$$\psi(2^{j-1}t - k) = \sqrt{2} \sum_{m-2k=0}^N f_1(m - 2k) \varphi(2^j t - m) \quad (\text{Ecuación 2.38})$$

Si se sustituye la ecuación 2.6 en la 2.10 y la 2.7 en la 2.11 se obtiene:

$$c_{j,k} = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} f(t) \varphi(2^j t - k) dt$$

$$d_{j,k} = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} f(t) \psi(2^j t - k) dt$$

Si se quiere obtener la aproximación y detalles de un nivel anterior al  $j$  se obtiene:

$$c_{j-1,k} = \int_{-\infty}^{\infty} 2^{\frac{j-1}{2}} f(t) \varphi(2^{j-1} t - k) dt \quad (\text{Ecuación 2.39})$$

$$d_{j-1,k} = \int_{-\infty}^{\infty} 2^{\frac{j-1}{2}} f(t) \psi(2^{j-1} t - k) dt \quad (\text{Ecuación 2.40})$$

Como se ve en las dos ecuaciones anteriores, se encuentran los términos  $\varphi(2^{j-1} t - k)$  y  $\psi(2^{j-1} t - k)$ , que se representan en las ecuaciones. 2.37 y 2.38. Sustituyendo:

$$c_{j-1,k} = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} f(t) \frac{\sqrt{2}}{\sqrt{2}} \sum_{m-2k=0}^N f_0(m-2k) \varphi(2^j t - m) dt$$

$$d_{j-1,k} = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} f(t) \frac{\sqrt{2}}{\sqrt{2}} \sum_{m-2k=0}^N f_1(m-2k) \varphi(2^j t - m) dt$$

Si se intercambia el orden de la integral y la sumatoria se obtiene:

$$c_{j-1,k} = \sum_{m-2k=0}^N f_0(m-2k) \int_{-\infty}^{\infty} f(t) 2^{\frac{j}{2}} \varphi(2^j t - m) dt$$

$$d_{j-1,k} = \sum_{m-2k=0}^N f_1(m-2k) \int_{-\infty}^{\infty} f(t) 2^{\frac{j}{2}} \varphi(2^j t - m) dt$$

Se puede ver que el valor de la integral, es igual a  $c_{j,k}$ , entonces:

$$c_{j-1,k} = \sum_{m-2k=0}^N f_0(m-2k) c_{j,k} \quad (\text{Ecuación 2.41})$$

$$d_{j-1,k} = \sum_{m-2k=0}^N f_1(m-2k) c_{j,k} \quad (\text{Ecuación 2.42})$$

Con las 2 ecuaciones anteriores, se puede ver como los coeficientes de los filtros  $f_0$  y  $f_1$  convolucionados con los coeficientes de aproximación del nivel  $j$  se pueden utilizar para obtener los coeficientes de detalle y aproximación de un nivel anterior.

## 2.6. Transformada ondelete discreta

Existen transformadas ondeletes continuas y discretas (también llamadas DWT), una DWT es la representación de una señal en escala-tiempo con cierto número de muestras igualmente espaciadas y que representan a la transformada ondelete continua. En el presente trabajo sólo se trata con DWT, la explicación de la transformada ondelete continua<sup>16</sup>, está fuera del alcance de este trabajo.

La DWT se obtiene, como se ha mostrado en las secciones anteriores, utilizando el método de filtros. El número de muestras de la DWT debe ser el mismo que el de la señal original.

---

<sup>16</sup> SARKAR, Tapan K.; SU, C. "A tutorial on Wavelets from an Electrical Engineering Perspective, part 2: Continuos Case". IEEE Antenas y Propagación. 1998, vol. 40, num. 6, pp. 36-49.

Para tener un algoritmo más eficiente en el cálculo de la DWT, es deseable que el número de muestras de la señal de entrada, sea una potencia de 2 ( $2^n$ ), porque en el proceso de sub-muestreo, el número de muestras de la señal se divide en 2, y con esto se va a tener la misma cantidad de componentes a la salida del filtro paso bajo como del filtro paso alto (la mitad de los de la señal de entrada), repitiéndose el proceso hasta que a la salida de los bloques de sub-muestreo tenga solamente 1 componente.

Si esto no se cumpliera es necesario agregar más coeficientes iguales a cero a la señal (hasta que el número sea una potencia de 2), para que el análisis de la DWT no se vea afectado en algún punto.

Las ecuaciones 2.52 y 2.53 se utilizan para obtener los detalles (coeficientes obtenidos con los filtros paso alto) y aproximaciones (coeficientes obtenidos con los filtros paso bajo), que son los componentes que conforman la DWT. Según estas ecuaciones pareciera que en un inicio se necesita pasar la señal por un filtro paso bajo, para obtener los valores de  $c_{j,k}$ , sin embargo como se cuenta con una señal de tiempo discreto que ocupa cierta banda de frecuencias, se puede utilizar ésta como punto de partida.

Dependiendo del tipo de filtro que se utilice, va a existir cierta cantidad de coeficientes, después de la convolución que causan distorsión tanto en los primeros componentes como en los últimos. Por esto es necesario hacer ciertas modificaciones a la señal de entrada que se presentan a continuación:

Primeramente, si se toma en cuenta que una señal de tiempo discreto, como las descritas aquí, cuenta con un número finito de muestras, de ésta se puede obtener otra, que sea periódica con período N (número de componentes de la señal original), utilizando los mismos coeficientes.

Esta acción es fácil de revertir porque sólo se escoge un período con las  $N$  muestras y todas las demás se hacen igual a cero, y se tiene nuevamente la señal original.

En un computador es imposible e ineficiente crear una señal periódica infinita, es por esto que sólo se necesita repetir cierto número de veces la señal. Este número se obtiene así:

$$n = \frac{(2 * l + \text{length}(v[n]) + 1 - \text{length}(f_0[n]))}{\text{length}(v[n])} \quad (\text{Ecuación 2.43})$$

Donde  $\text{length}(v[n])$ , es la cantidad de componentes de señal a descomponer,  $\text{length}(f_0[n])$ , es la cantidad de componentes del filtro,  $n$  el número de veces a repetir la señal y  $l$  un retraso que causa el filtro.

El retraso se obtiene debido a que los filtros a utilizar, forman un banco de reconstrucción perfecta que deben cumplir con la condición impuesta en la ecuación 2.22, ésta indica que si se multiplican las transformadas  $Z$  de los filtros paso bajo y paso alto y luego se suman, debe haber solamente un componente que sea distinto de cero e igual a 2. Éste se va a encontrar en cierta posición, que indica el retraso  $l$  que van a imponer los filtros a la señal de salida.

La ecuación 2.43 se basa en la cantidad de componentes que se obtienen cuando se convolucionan 2 señales y la cantidad de componentes que se necesitan, tomando en cuenta los retrasos que imponen los filtros a la señal de salida. La cantidad de componentes que se obtienen después de convolucionar 2 señales es: la suma del número de componentes de las 2 señales menos 1.

Si se toma en cuenta que la primera señal a convolucionar es la de entrada repetida  $n$  veces y la otra señal es el filtro  $f_0$ , entonces queda una ecuación de la siguiente manera:

$$n * length(v[n]) + length(f_0[n]) - 1 \quad (\text{Ecuación 2.44})$$

El retraso que imponen los filtros después de convolucionar la señal de entrada, se muestra tanto al inicio como al final de la señal de salida, esto se logra ver después de repetir varias veces la señal de entrada y convolucionarla con el filtro, en donde los componentes desde el 1 hasta el de retraso y desde el último-retraso hasta el último, no se repiten nunca en la señal. Esto nos indica que estos componentes son transientes de la señal y que el resto si pertenecen a la DWT.

Como después de convolucionar se obtiene cierto número de componentes que se submuestran, este número debe ser como mínimo igual a la longitud de la señal de entrada para que se obtenga la mitad del número de componentes de la señal de entrada. Entonces el número mínimo de componentes a la salida de la convolución debe ser:

$$2 * retraso + length(v[n]) \quad (\text{Ecuación 2.45})$$

Si se igualan las ecuaciones 2.44 y 2.45 y se despeja  $n$  se obtiene la ecuación 2.43. Para obtener la DWT hay que referirse a la figura 12, que muestra que lo anteriormente descrito se repite varias veces, hasta que se quiera parar, o hasta que los coeficientes de aproximación y detalle sean iguales a 1. Esto se consigue en el nivel al que se eleva el 2, o sea  $n$  ( $2^n$ ), es decir, se obtiene de:

$$n = \log_2 N \quad (\text{Ecuación 2.46})$$

Siendo  $N$  el número de muestras de la señal original.

Normalmente cuando se obtiene la transformada de ondelete, la energía de ésta debe ser igual a la energía de la señal de entrada, para que no exista un desequilibrio en la energía.

Para lograr esto, se tiene que multiplicar alguna de las dos señales por un factor  $\alpha$ . Por ejemplo si se tiene la señal de entrada  $x[n]$ , el filtro  $f_0[n]$  y la señal de salida  $y[n]$  entonces la señal de entrada va a tener una energía igual a:

$$E_{in} = \sum_{k=0}^{N-1} x[k]^2 \quad (\text{Ecuación 2.47})$$

Si se tienen los siguientes coeficientes para el filtro  $f_0[n]$

$$f_0[n] = [a_0 \ a_1 \ \dots \ a_r] \quad (\text{Ecuación 2.48})$$

Donde  $r$  es el número de coeficientes menos 1 del filtro. La señal de salida de un sistema como el descrito, es:

$$y[n] = f_0[n] * x[n] \quad (\text{Ecuación 2.49})$$

Del teorema de Parseval<sup>17</sup>, la potencia de salida es igual a:

$$\sum_{l=1}^{M-1} Y[l]^2 = \sum_{t=1}^{r-1} a[t]^2 \sum_{s=1}^{N-1} X[s]^2 \quad (\text{Ecuación 2.50})$$

---

<sup>17</sup> TAUB, Herbert; SCHILLING, Donald. *Principles of Communication Systems*. Segunda Edición. Nueva York: Editorial McGraw Hill, 1986. pp. 23-24.

Donde  $Y[l]$  son los coeficientes de la transformada Z, de la señal de salida,  $a[t]$  los del filtro,  $X[s]$  los de la señal de entrada,  $M$  el número de coeficientes de la salida,  $r$  el de los filtros y  $N$  el de la señal de entrada.

De esta ecuación, podemos ver que la suma de los cuadrados del filtro debe ser igual a 1, para que la energía de la señal de entrada sea igual a la de la señal de salida. La ecuación 2.51 representa esto.

$$\sum_{k=1}^{r-1} a[k]^2 = 1 \quad (\text{Ecuación 2.51})$$

## 2.7. Transformada Ondelete discreta inversa

Si de una señal, se obtiene la transformada ondelete por medio del método de banco de filtros de reconstrucción perfecta, debe haber una forma que la señal pueda volverse a formar. En la DWT, sólo se utiliza un filtro paso bajo y un filtro paso alto del banco. Los otros dos, que pertenecen a un banco de reconstrucción perfecta, se utilizan para volver a obtener la señal original. A continuación se explica cómo se logra esto.

La figura 13, muestra el proceso que se debe hacer para volver a obtener la señal original. Aquí se ve que la transformada ondelete se encuentra distribuida en varios niveles, lo primero que se debe hacer es convertir la señal en periódica, para que a la hora de convolucionar con los filtros se pueda evitar la distorsión.

Seguidamente se sobre-muestran las señales de aproximación y detalle periódicas, para revertir lo realizado por el sub-muestreo, luego se convolucionan esta señal con los filtro respectivos, se suman éstas y finalmente se escogen los componentes para utilizar como aproximación en el siguiente nivel.

Éstos se escogen de la misma forma que en la sección anterior eligiendo únicamente los coeficientes que se repiten, todo esto es posible gracias al análisis multi-resolución que muestra que la suma de un espacio vectorial (aproximación), más su complemento (detalle), forman un espacio en el siguiente nivel (aproximación). Esto se muestra en la ecuación 2.52.

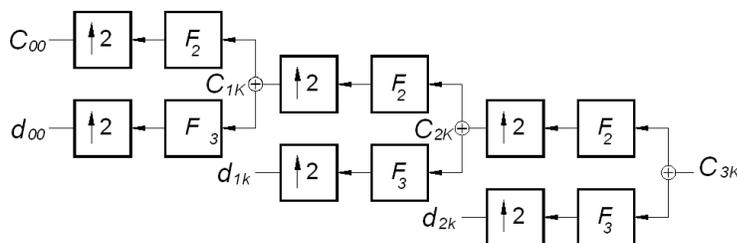
$$a_0 + d_0 = a_1 \quad (\text{Ecuación 2.52})$$

Lo anteriormente descrito se repite hasta regresar al último nivel, en el que se obtiene la señal original.

Según el banco de filtros de reconstrucción perfecta, la señal de salida va a ser una réplica de la señal de entrada, sólo que retrasada en el tiempo. Sin embargo, el análisis por computador, permite tomar la señal de salida, de tal manera que sea justamente la señal de entrada sin el retraso. Este retraso, existe pero se puede evitar, tomando los coeficientes adecuados de la señal de salida.

Esta transformada Ondelete inversa, será utilizada en el capítulo 3, para poder explicar el ejercicio que sirve para determinar cuántos componentes son necesarios para poder distinguir una canción.

Figura 13. **Transformada ondelete inversa**



Fuente: elaboración propia

## 2.8. DWT como vector

Existen muchas formas de ordenar los componentes de la transformada de ondelete, una forma muy conveniente es la que se muestra en la ecuación 2.53, porque se van colocando los términos, desde la escala más baja hasta la más alta. Esto es como si se tuviera el coeficiente del filtro paso bajo al principio, luego el del primer filtro pasa banda, luego los del segundo filtro pasa banda y así sucesivamente hasta llegar a los del filtro paso alto. Los valores de  $k$  se ordenan desde cero hasta  $2^j - 1$ .

$$DWT = [c_{0,0} d_{0,0} d_{1,0} d_{1,1} d_{2,0} d_{2,1} d_{2,2} d_{2,3} \dots] \quad (\text{Ecuación 2.53})$$

Ésta puede ser vista como un vector en distintas dimensiones, porque las bases que se utilizan para cada componente, son ortogonales.

Sabiendo que sólo se necesita cierto número de muestras para poder recomponer una señal, es útil identificar por medio de una codificación a qué base pertenece cada componente. Estas bases también llamadas índices, están representadas por su nivel  $j$  y su traslación en el tiempo  $k$ .

La tabla II muestra un ejemplo de la codificación que se utiliza en el presente trabajo para cada componente.

Tabla II. **Ejemplo de codificación**

j	K	a/d	Val.	j	K	a/d	Val.	j	K	a/d	Val.	J	k	a/d	Val.
0	0	A	0	4	0	D	16	5	0	D	32	5	16	D	48
0	0	D	1	4	1	D	17	5	1	D	33	5	17	D	49
1	0	D	2	4	2	D	18	5	2	D	34	5	18	D	50
1	1	D	3	4	3	D	19	5	3	D	35	5	19	D	51
2	0	D	4	4	4	D	20	5	4	D	36	5	20	D	52
2	1	D	5	4	5	D	21	5	5	D	37	5	21	D	53
2	2	D	6	4	6	D	22	5	6	D	38	5	22	D	54
2	3	D	7	4	7	D	23	5	7	D	39	5	23	D	55
3	0	D	8	4	8	D	24	5	8	D	40	5	24	D	56
3	1	D	9	4	9	D	25	5	9	D	41	5	25	D	57
3	2	D	10	4	10	D	26	5	10	D	42	5	26	D	58
3	3	D	11	4	11	D	27	5	11	D	43	5	27	D	59
3	4	D	12	4	12	D	28	5	12	D	44	5	28	D	60
3	5	D	13	4	13	D	29	5	13	D	45	5	29	D	61
3	6	D	14	4	14	D	30	5	14	D	46	5	30	D	62
3	7	D	15	4	15	D	31	5	15	D	47	5	31	D	63

Fuente: elaboración propia

La columna de  $j$  representa el nivel del componente, la de  $k$  la traslación de la señal, la de  $a/d$  muestra si el coeficiente es de aproximación o de detalle respectivamente, y finalmente Val., el valor que le asigna la codificación a ese componente.

El anterior es solamente un ejemplo de codificación de índices de una DWT, debido a que las señales cuentan con un número mayor de muestras, el tamaño de la tabla va aumentando. Conforme se va pasando de nivel a nivel, solo se va sumando uno al valor de codificación del nivel anterior. Estos valores de codificación, se obtienen por medio de las siguientes ecuaciones y se utilizarán en el capítulo 3 para poder almacenar los segmentos:

$$A = 0 \quad \text{(Ecuación 2.54)}$$

$$D = 2^j + k \quad \text{(Ecuación 2.55)}$$

## 2.9. Normalización

Toda señal (vector), tiene cierta energía, que se calcula como se muestra en la ecuación 2.56.

$$E_v = V_1^2 + V_2^2 + V_3^2 + \dots + V_n^2 \quad (\text{Ecuación 2.56})$$

Donde  $E_v$  representa la energía de la señal,  $V_i$ , el componente de la transformada de ondelete, es decir  $c_{0,0}$  o  $d_{j,k}$  y  $n$  el número de muestras que tiene la señal.

En este trabajo cada señal se ve como que fuera un vector. Normalizar es construir un vector a partir de otro, de manera que tenga una norma unitaria. La norma, representada como norma-p, es la medida de la magnitud de un vector. La norma de un vector puede ser cualquier número positivo. Aquí se utiliza la norma-1 y la norma-2.

Las ecuaciones que representan a éstas, se muestran a continuación.

$$|V_1| + |V_2| + |V_3| + \dots + |V_n| = 1 \quad (\text{Ecuación 2.57})$$

$$\sqrt{V_1^2 + V_2^2 + V_3^2 + \dots + V_n^2} = 1 \quad (\text{Ecuación 2.58})$$

Donde los  $V_i$  representan a cada muestra de la señal y  $n$  el número de éstas.

La norma 2, también puede ser vista como la raíz cuadrada de la energía de la señal igual a 1. Esto se puede lograr sustituyendo la ecuación 2.56 en la 2.58, es decir:

$$\sqrt{E_v} = 1 \quad (\text{Ecuación 2.59})$$

En el presente trabajo, como se explicó en el capítulo 1, se va a tener un grupo de segmentos con distintas energías, por tanto es conveniente, que se normalicen todos con la norma 2, para que tengan la misma magnitud y cuando se multiplique la señal por ella misma sea igual a 1, lo cual va a ser de mucha utilidad para el que el *software* de búsqueda, que se explicará en la sección 2.9, no vuelva a elegir el mismo segmento en la siguiente interacción.

Debido a que las señales manejan distintas energías, usualmente se necesita que la energía de una de las dos se normalice a la energía de la otra. Esto se logra multiplicando por un valor  $\alpha$  alguna de las dos, y se muestra en la ecuación 2.60.

$$\alpha E_i = E_o \quad (\text{Ecuación 2.60})$$

De aquí se puede despejar  $\alpha$  (alfa):

$$\alpha = \frac{E_o}{E_i} \quad (\text{Ecuación 2.61})$$

Esto permite que la energía de la señal escogida, multiplicada por alfa sea igual a la energía de la señal a representar y se conserve la energía en el sistema.

## 2.10. Algoritmo búsqueda de emparejamiento

Es una técnica numérica (en inglés "*Matching Pursuit*"), propuesta en 1993 por Mallat y Zhang<sup>18</sup>, y consiste en obtener funciones y escalares de un diccionario de señales, que al sumarse representen a una señal  $g(t)$ .

---

<sup>18</sup> MALLAT, S. G.; ZHANG, Z. "Matching Pursuit with Time-Frecuency Dictionaries". IEEE Transacción en procesamiento de señales. 1993, vol.41, num. 12. pp. 3397-3415

Esto se representa en la ecuación 2.62.

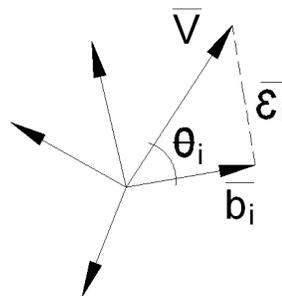
$$g(t) = \sum_{n=0}^{\infty} \beta_n h_n(t) \quad (\text{Ecuación 2.62})$$

Donde  $h_n(t)$  son las funciones del diccionario y  $\beta_n$  los escalares por los que se tienen que multiplicar éstas, para que se represente la señal original de una mejor manera. Este algoritmo lo propusieron para que computacionalmente existiera una búsqueda eficiente de señales en un gran diccionario.

Su funcionamiento, consiste en tener una señal que se quiere representar por la suma de varios vectores que se encuentran en un diccionario. Se inicia obteniendo el producto escalar entre el original y los que pertenecen al diccionario, que indica el valor por el que se tiene que multiplicar el vector elegido para minimizar la distancia entre los vectores.

Esto se desprende de, se tiene la figura 14 y la ecuación 2.63 que representa el producto punto entre el vector original  $\bar{V}$  y los vectores del diccionario  $\bar{b}_i$ .

Figura 14. **Vectores**



Fuente: elaboración propia

$$\bar{V} \cdot \bar{b}_i = |V| |b_i| \cos(\theta_i) \quad (\text{Ecuación 2.63})$$

De la figura 14, se tiene que el vector original se puede representar por:

$$\bar{V} = \bar{\varepsilon}_i + \beta_i \bar{b}_i \quad (\text{Ecuación 2.64})$$

Si se obtiene el producto escalar entre el vector mostrado en la ecuación 2.64 y uno del diccionario  $\bar{b}_i$  se obtiene:

$$\bar{V} \cdot \bar{b}_i = \bar{\varepsilon}_i \cdot \bar{b}_i + \beta_i \bar{b}_i \cdot \bar{b}_i \quad (\text{Ecuación 2.65})$$

El primer término del lado derecho de la ecuación 2.65 es igual a 0, porque son vectores perpendiculares y de la ecuación 2.63 el coseno de 90 grados es igual a 0. Como se explicó en la sección 2.8 los vectores del diccionario están normalizados con la norma 2, entonces el producto punto del vector por el mismo es igual a 1 por tanto se obtiene la ecuación 2.66.

$$\beta_i = \bar{V} \cdot \bar{b}_i \quad (\text{Ecuación 2.66})$$

Entonces si los vectores del diccionario y el original están normalizados y tienen magnitud igual a 1 de la ecuación 2.63, el coseno del ángulo entre los dos vectores va a ser igual a  $\beta_i$  y si el producto punto entre dos vectores normalizados es igual a 1, son el mismo vector, por tanto el vector que se elige del diccionario es el que el valor absoluto de su producto punto con el vector original sea el mayor.

Ya con el vector del diccionario  $b_i$  y el factor multiplicativo  $\beta_i$  se tiene una parte del vector original representada, sin embargo para que la señal se parezca más, se tiene que escoger otro vector, que es la diferencia entre el original y el elegido, esto se obtiene de la ecuación 2.64, entonces el nuevo vector a buscar en la base de datos es:

$$\bar{\varepsilon}_i = \bar{V} - \beta_i \bar{b}_i \quad (\text{Ecuación 2.67})$$

Este nuevo vector se utiliza como el vector original y se vuelve a hacer lo anteriormente descrito. Esto se puede hacer cuantas veces se desee, o hasta que el vector error se casi igual al vector cero.

## **2.11. Matlab**

En el presente trabajo, se utiliza la plataforma de cálculo Matlab versión 7.6.0 (R2008a) 22. El motivo que se tiene para usar este lenguaje en específico, es por la familiaridad que se tiene con las herramientas que posee, por ejemplo la del manejo de archivos de audio, algebra matricial, la versatilidad para crear bases de datos entre otros.

Matlab, es la abreviatura de Laboratorio de Matrices por sus siglas en inglés. Es un lenguaje que trabaja como su nombre lo indica a base de matrices, lo que lo hace muy hábil para el manejo de éstas. En esta plataforma es posible crear funciones y programar en lenguaje m, además de crear interfaces gráficas GUI para hacer más fácil su uso al usuario.

Entre las herramientas que tiene, cuenta con la de poder manejar archivos de audio, específicamente los que tienen extensión “wav” y “au”.

Existen dos funciones para la lectura, las cuales son; “leer wav” (en inglés “wavread”) o bien “leer au” (en inglés “auread”), para cada formato respectivamente. Estas funciones son capaces de devolver un vector (monofónico), o una matriz (estereofónico), donde se ubican las muestras de la canción, además informan a que frecuencia está muestreada la señal y cuántos bits se utilizan para la codificación de las muestras. También se pueden crear archivos de audio, utilizando las funciones “escribir wav” (en inglés “wavwrite”) y “escribir au” (en inglés “auwrite”).

En la “Ayuda de Matlab”<sup>19</sup>, se puede obtener más información del uso de funciones de este *software*.

Como se mencionó, Matlab sólo trabaja con archivos “wav” o “au”, generalmente la mayoría de archivos de música, vienen codificados como “mp3” por el espacio que ocupan. Es por esto que se necesita la ayuda de un *software* extra, que sea capaz de convertir las canciones que tengan algún otro formato a los que maneja Matlab.

Podemos mencionar, por ejemplo el “*Soft Audio Converter*”<sup>20</sup>, un *software* capaz de dividir, convertir de “mp3” a “wav” o a “au” o viceversa entre otras cosas. Pero por supuesto, que existen muchísimos más, se menciona éste por ser el que se utilizó para hacer las conversiones de archivos.

MATLAB, también cuenta con una herramienta para guardar datos. De aquí es posible crear bases de datos con extensión “mat”. Las funciones que se utilizan para este propósito son “guardar” (en inglés “save”) y “cargar” (en inglés “load”). En el capítulo 3 se explica mejor como crear la tabla para la base de datos de la presente.

A base de información, Matlab también cuenta con herramientas para los ondeletes, sin embargo la mayoría de programas de el presente trabajo fueron creados y no copiados de esta herramienta. Para que quede constancia de esto, cada programa se explica a detalle en el capítulo 3 y se ubican en el apéndice 4.

---

<sup>19</sup> THE MATHWORKS. *Image Processing Toolbox™ 7 User’s Guide [en línea]*. Estados Unidos. 1997. Revisión: marzo 2010 [Consulta: junio 2010]. Disponible en Web: <<http://www.mathworks.com/help/toolbox/images/bqj5b2j.html>>.

<sup>20</sup> TANG, HongLiang. *Soft Audio Converter*. Versión 1.67, 2010.

## 3. FUNCIONALIDAD

### 3.1. Objetivos

El objetivo de este capítulo, es explicar cómo se realizó el experimento completo, para que el *software* recomponga una canción utilizando los ondeletes como la base principal del trabajo, se explica a detalle, y por medio de diagramas de flujo se muestra, cómo es que funcionan algunas de las partes que se muestran en la figura 5, además de explicar algunos ejercicios que se realizaron para poder elegir de mejor manera los ondeletes a utilizar, el tamaño de los segmentos, la cantidad de componentes notables a utilizar, entre otros.

### 3.2. Sonidos originales de la base de datos

En la figura 5, se muestra este cuadro en la parte inferior derecha. Básicamente aquí se escogieron los sonidos y se segmentaron para recomponer una canción.

Se escogieron sonidos de la naturaleza, para que recompusieran las canciones, obtenidos de unos discos de nombre “Ecos de la Naturaleza”<sup>21</sup> (en inglés “*Echoes of Nature*”), entre los que se encontraban cantos de pájaros, ruidos de ranas, chillidos de grillos, sonidos de cerdos, el ruido de la lluvia, y generalmente sonidos de distintos animales que se encuentran en la selva.

---

<sup>21</sup> GIBSON, DAN; BEKKER, Hennie. *Echoes of Nature*. [Grabación sonora].

Estos sonidos, fueron escogidos porque los pájaros, dependiendo su especie, pueden cantar a distintas frecuencias en los tonos medios, las ranas pueden hacer los sonidos de los tonos bajos, al igual que los cerdos, los grillos dan los sonidos de los tonos altos, lo que hace a estos ruidos muy convenientes para representar canciones.

Para estar seguro, de que los sonidos escogidos eran adecuados, se hizo un ejercicio. Para esto se segmentaron todas las canciones a un tamaño de 1 segundo y se obtuvo la transformada discreta de Fourier de cada segmento, con la DFT se procedió a calcular cuál era la frecuencia ponderada promedio para cada segmento por medio de la ecuación 3.1. La DFT se obtuvo con la función de Matlab “FFT” que es la que corresponde a la transformada rápida de Fourier. Refiérase a la ayuda de Matlab<sup>22</sup> para más información de la función FFT.

$$f_{Pond.Prom} = \frac{\sum_{n=0}^{N-1} c_n f(2\pi n f_0)}{N} \quad (\text{Ecuación 3.1})$$

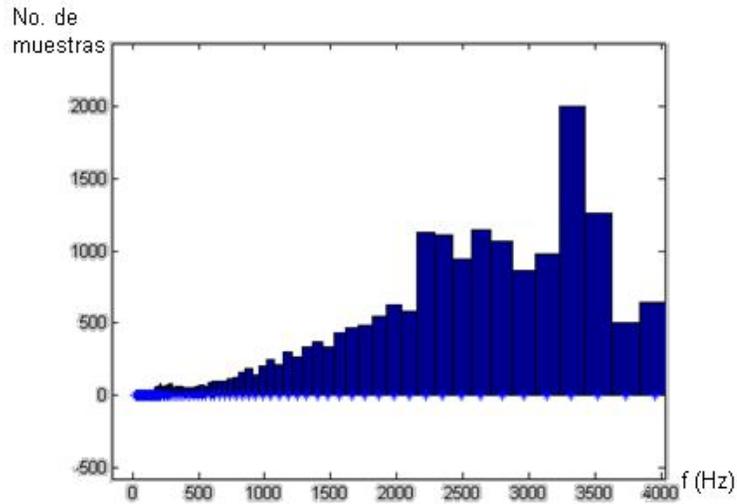
Donde,  $c_n$  es el componente de la DFT que pertenece a la frecuencia  $f(2\pi n f_0)$ , y  $N$  la cantidad de componentes de la señal.

Esta ecuación se utilizó para saber qué frecuencia era la que representaba a cada segmento. Éstas se compararon con las de las teclas de un piano, para saber si los sonidos escogidos podían representar las notas de este instrumento. Se utilizó un histograma en el que se representó la cantidad de veces que aparecía cada nota en la base de datos. La figura 15 muestra esto.

---

<sup>22</sup> THE MATHWORKS. *Image Processing Toolbox™ 7 User's Guide [en línea]*. Estados Unidos. 1997. Revisión: marzo 2010 [Consulta: junio 2010]. Disponible en Web: <<http://www.mathworks.com/help/toolbox/images/bqj5b2j.html>>.

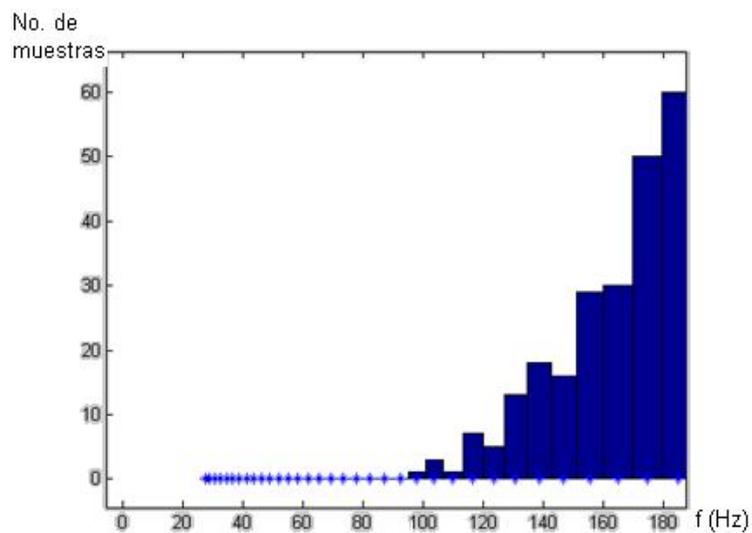
Figura 15. **Histograma de base de datos**



Fuente: elaboración propia

Como se puede ver la mayoría de las notas musicales de un piano están representadas en la base de datos. En la figura 16 se hace un acercamiento, en el que se logra observar las notas que no están representadas.

Figura 16. **Acercamiento de histograma**

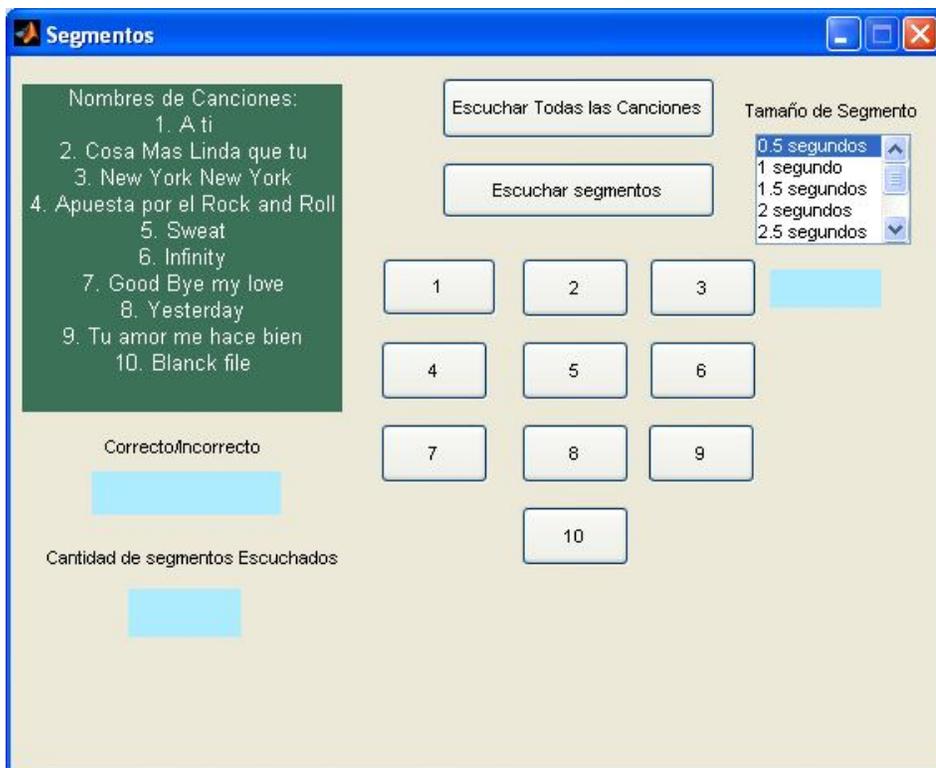


Fuente: elaboración propia

De esta figura se concluyó que, la base de datos tiene segmentos que pueden representar señales que se encuentran en el rango de frecuencias de 97,99 Hz hasta 4 186 Hz, las cuales corresponden a la tecla 23 (SOL en la 2da. octava) y a la 88 (DO en la 8va. Octava) respectivamente, en un piano profesional. Entonces los sonidos cubren una escala musical de alrededor de 5,5 octavas lo que los hace muy buenos para la representación de señales.

Para elegir el tamaño de los segmentos se hicieron varias pruebas. Una de ellas fue proponer un ejercicio que consistía en descomponer 10 canciones conocidas a distintos tamaños, esto en base a tiempo. La interfaz gráfica que se mostró a los usuarios se muestra en la figura 17.

Figura 17. **Interfaz gráfica para la elección del tamaño del segmento**



Fuente: elaboración propia

Como se ve en la figura 17 la interfaz gráfica contaba con los siguientes componentes:

- Una lista de canciones que son las que se utilizaron para ésta prueba
- Un botón llamado “Escuchar Todas las Canciones”, para que la persona que fuera a usar el *software*, pudiera antes de hacer la prueba escuchar las canciones para que se familiarizara con ellas
- Una lista para poder escoger el tamaño de segmento a escuchar, como se puede ver hay de 0,5 segundos equivalente a 22 050 muestras, 1 segundo a 44 100 muestras, 1,5 segundos a 66 150 muestras, 2 segundos a 88 200 muestras y 2,5 segundos a 110 250
- Un cuadro en el que se muestra cuál es el elegido
- Un botón nombrado “Escuchar segmentos” es el que se encargaba de escoger una canción al azar y reproducir sus segmentos desordenadamente
- Diez botones enumerados del 1 al 10, que conforme los segmentos se van reproduciendo, y el usuario reconoce la canción, puede presionar cualquiera dependiendo del número de canción que según él es la que se está reproduciendo
- Para mostrar si el resultado fue correcto o incorrecto se presenta el cuadro con ese nombre
- Y finalmente para saber cuántos segmentos tuvo que escuchar, se presenta el otro cuadro de nombre “Cantidad de segmentos escuchados”

Las canciones que se descompusieron, se enlistan en la tabla III que se muestra a continuación.

Tabla III. **Lista de canciones que se utilizaron para elegir un tamaño de segmento**

Nombre de canción	Autor
A ti	Ricardo Arjona
Cosa más linda que tú	Eros Ramazoti
New York New York	Frank Sinatra
Apuesta por el Rock and Roll	Héroes del Silencio
Sweet	Inner Circle
Infinity	Guru Josh Proyect
Good bye my love	James Blunt
Yesterday	The Beatles
Tu amor me hace bien	Marc Antony
Blanck file	Sonata Ártica

Fuente: elaboración propia

Éstas fueron escogidas así, para que se tomaran en cuenta distintos ritmos y así se tuviera un resultado más general y no sólo de una clase de música. Para tomar una decisión, el *software* almacenó unos archivos de Excel que se muestran en el apéndice 1.

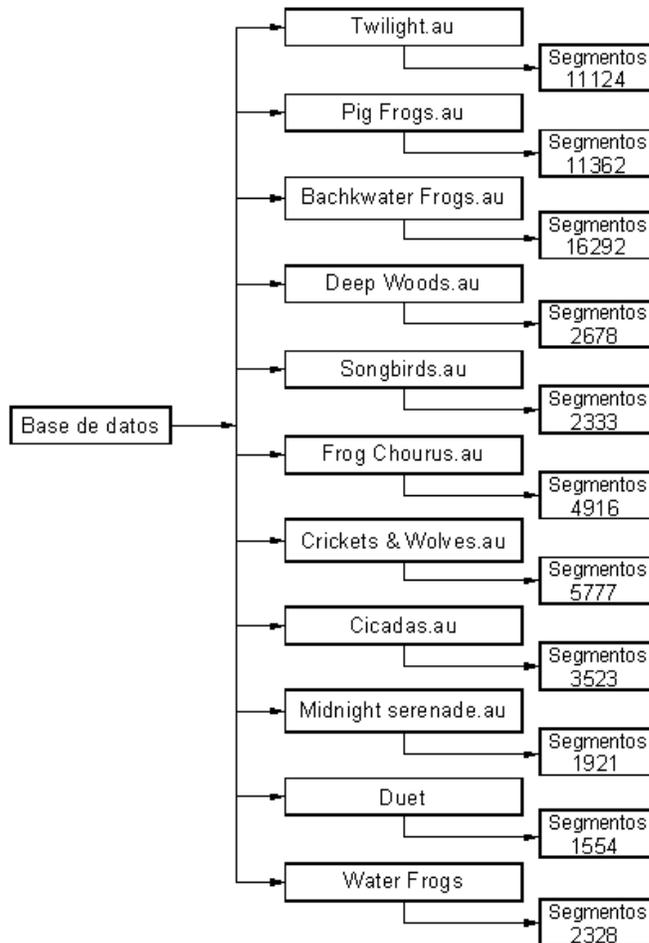
De este ejercicio se concluyó que se necesitaba, para reconocer una canción; segmentos de aproximadamente 1 segundo. Estas canciones están compuestas por voz, es por esto que los segmentos tienen que ser de este tamaño para que se puedan reconocer. Sin embargo los sonidos que se utilizan para la base de datos son de la naturaleza, por tanto se dispuso que si un segmento de menor tamaño, aproximadamente en el orden de 1/8 de tiempo musical (fusa) a 1/16 (semifusa), era escogido, no habría ningún problema en reconocer de dónde provenía, debido a que en la naturaleza se tienen sonidos como el canto de un pájaro, el ruido de una rana, el sonido de la lluvia, etc.

Con estos resultados obtenidos y con la interrogación de cuál era el tamaño correcto, se propuso un segundo ejercicio. Sin embargo, éste requiere la recomposición de una canción, por lo que se explicará durante el recorrido de este capítulo.

Sólo adelantaremos que para hacer todas las pruebas, se utilizaron 4 tamaños de segmentos en base a su número de muestras, los que se escogieron fueron 32 768 muestras, 16 384 muestras, 8 192 muestras y 4 096 muestras. Como se puede ver, todos los tamaños escogidos tienen la propiedad de tener un número de muestras que es una potencia de dos, esto para que el algoritmo de la transformada ondelete fuese más eficiente.

Para almacenar los segmentos que pertenecían a cada canción, se utilizó la función de Matlab *wavwrite*. Se pensó guardarlos de forma ordenada para que cuando el bloque “Búsqueda y Reconocimiento” escogiera uno, fuera fácil para el bloque “Composición” ubicarlo. Entonces se propuso identificar a los segmentos por medio de números, empezando desde el 1 hasta la cantidad de segmentos que se obtenían de la canción, y almacenar estos en carpetas que se llamaran como la pieza musical. Finalmente todas las canciones se almacenaron en otra carpeta llamada base de datos, esto se muestra en la figura 18.

Figura 18. **Distribución de almacenamiento de segmentos**



Fuente: elaboración propia

### 3.3. Base de datos

Este cuadro tenía como función principal el caracterizar los segmentos provenientes del bloque “Sonidos Originales de la Base de Datos” por medio de la transformada ondelete. Esta parte es muy importante porque es donde se ve la utilidad que tienen los ondeletes para señales de audio en general.

Para poder armar este bloque se hicieron muchas pruebas, ya que se necesitaba saber cuál era el ondelete a utilizar, la cantidad de componentes notables a almacenar, cómo almacenarlos entre otras cosas.

A continuación se explica cómo fueron obtenidos los coeficientes de los filtros Daubechies, y los distintos ejercicios que se realizaron para este bloque.

### 3.3.1. Construcción de los Ondeletes Daubechies

Los ondeletes Daubechies son unos filtros ortogonales. La construcción de éstos se refería a la obtención de sus coeficientes. Para lograr esto se utilizó la ecuación 2.32, que por conveniencia se vuelve a repetir aquí.

$$T_0(z) = (1 + z^{-1})^{2p} \frac{1}{2^{2p-1}} \sum_{k=0}^{p-1} \binom{p+k-1}{k} (-1)^k z^{-(p-1)+k} \left(\frac{1-z^{-1}}{2}\right)^{2k} \quad (\text{Ecuación 3.2})$$

Esta ecuación, como ya se sabe, representaba la multiplicación de los dos filtros paso bajo  $F_0$  y  $F_2$  que se encontraban en un banco de reconstrucción perfecta. Entonces, esta ecuación había que separarla de alguna manera en dos. Los filtros Daubechies, como se mencionó en la sección 2.4.2, tenían la misma cantidad de ceros en los dos filtros y el filtro  $F_0$  era de fase mínima, y  $F_2$  era de fase máxima.

Para este programa, se necesitaba un valor de entrada que es el parámetro  $p$  de los filtros. Este parámetro definía cuantos componentes iba a tener cada filtro, siendo un total de  $2p$ .

En Matlab los polinomios se definen como vectores. Para multiplicar dos vectores se utiliza la convolución y la suma se hace entre vectores del mismo tamaño.

El primer término del lado izquierdo de la ecuación 3.2, mostraba que para la multiplicación existían  $2p$  ceros en  $z=-1$ , entonces,  $p$  ceros pertenecían a  $F_0$  y los otros  $p$  ceros a  $F_2$ . El polinomio que pertenecía a este primer término se obtenía al convolucionar, en Matlab, el vector [1 1] con el mismo  $p-1$  veces.

Con la sumatoria se obtenía el otro polinomio. Lo primero que se hacía, era obtener el valor que se encontraba entre el paréntesis que representaba una combinación (número de formas en que se pueden extraer subconjuntos a partir de un conjunto dado) por medio de la siguiente ecuación:

$$\binom{p-1}{0} = \frac{(p-1)!}{0!(p-1)!} \quad (\text{Ecuación 3.3})$$

Y se obtenían los demás términos para  $k=0$ . Para todos los coeficientes este primer polinomio era únicamente un retraso de magnitud 1 en  $p-1$ . Luego se obtenía el siguiente polinomio, y antes de sumarlos, se empataban en tamaño. Y así sucesivamente hasta que se llegaba al polinomio  $p-1$ .

Con este polinomio obtenido, se procedía a utilizar la función de Matlab "roots" que daba como resultado las raíces del mismo. Al obtener las raíces se tomaba para el filtro  $F_0$  las que se encontraban dentro del círculo unitario y para  $F_2$  las que estaban afuera. Para que la ecuación quedara en forma de ceros, a cada raíz se le restaba el término  $z^{-1}$ . Ya con esto se procedía a convolucionar éstas con el polinomio obtenido del primer término del lado izquierdo de la ecuación 3.2. En la mayoría de los filtros existían raíces complejas, pero también se encontraba su conjugado, lo que hacía que todos los coeficientes fueran reales.

Existía un término en la ecuación 3.2 representado por  $\frac{1}{2^{2p-1}}$ , el cual no se tomaba en cuenta porque con el polinomio expandido se procedía a normalizar, por tanto se desechaba. La normalización se hacía para que, como se explicó al final de la sección 2.6, no existiera diferencia en las energías de la señal original y de la que se obtuvo de la transformada de ondelete.

Para los coeficientes del filtro  $F_2$ , se utilizara la función de Matlab “*flipr*” que cambiaba el orden de los coeficientes, es decir, el primero lo colocaba de último, el segundo de penúltimo y así sucesivamente hasta llegar al último que lo colocaba de primero.

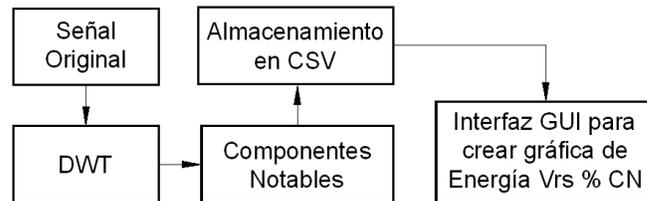
Para los coeficientes de los otros dos filtros es decir  $F_1$  y  $F_3$  se utilizaron las ecuaciones 2.25 y 2.26. El programa se muestra en el apéndice 4 bajo el nombre de “Programa de filtros Daubechies”.

### **3.3.2. Ejercicio para gráficas de energía contra componentes notables**

El primer ejercicio que se hizo, fue crear unas gráficas que además de mostrar el camino hacia qué ondeletes utilizar, también dieron una perspectiva de cuántos componentes notables se podían necesitar para la creación de la base de datos. Estas gráficas eran de dos dimensiones, y consistían en representar la energía de alguna canción, en base a la cantidad de componentes notables utilizados.

El diagrama de flujo de la figura 19 muestra cómo funciona el *software*, que sirve para que el usuario pueda ver las gráficas de energía contra porcentaje de componentes notables.

Figura 19. **Diagrama de flujo para gráficas energía contra porcentaje de componentes notables**



Fuente: elaboración propia

A continuación se explica cada uno de los bloques:

### 3.3.2.1. Señal original

En esta parte, se escoge una canción que se quiera graficar, esta canción, se segmenta en base a un número de muestras, todos los segmentos de la canción deben ser del mismo tamaño, si de la canción original existiera un segmento al que le hagan falta muestras para cumplir con el tamaño, éste no se toma en cuenta, descartándose las muestras que pertenecen a él. Finalmente todos los segmentos se almacenan en una matriz con las siguientes dimensiones: número de muestras por el número de segmentos, entonces cada columna va a pertenecer a cada segmento.

Las gráficas son de una canción, por tanto, se almacenaron todos los segmentos para poder tomar en cuenta toda la canción. Para leer los archivos de audio se utilizó la función de Matlab “*wavread*” o “*auread*” según fuera el caso. Además se utilizaron segmentos de 32 768 muestras que es un poco menos de un segundo, este tamaño fue elegido porque es un aproximado del que se obtuvo con el primer ejercicio explicado en la sección anterior. Ésta parte es muy importante porque de aquí se va a sacar la información necesaria para el resto de bloques que se muestran en la figura 20.

### 3.3.2.2. DWT

En este bloque se utilizó la matriz obtenida de “Señal Original”, para obtener la transformada de ondelete de cada segmento, aquí se obtenía otra matriz de las mismas dimensiones, es decir el número de muestras por número de segmentos, sólo que con los coeficientes de la transformada de ondelete de cada segmento.

Para cada canción se obtuvo 20 matrices, porque se utilizaron 20 ondeletes distintos, los ondeletes que se utilizaron fueron los Daubechies, desde DB2 hasta DB21.

Para saber cuáles son los coeficientes de estos filtros, referirse al Anexo 1, en donde se encuentran unas tablas con los coeficientes de los filtros  $F_0$  y  $F_1$ , los coeficientes de los otros dos filtros se obtienen de las ecuaciones 2.25 y 2.26.

El número de coeficientes de los filtros está relacionado al número que aparece a la par de las iniciales DB. Este número es el parámetro  $p$ , que se mencionó en el capítulo 1, y como se explicó allí, el número de coeficientes es igual a  $2p$ . La suma de los cuadrados de cada filtro es igual a 1, lo que permite que la energía sea igual tanto en el tiempo, como en la representación de ondeletes.

Ya con las matrices ondeletes obtenidas se necesitaba ordenar los componentes de alguna manera, entonces se procedió a elevar cada componente al cuadrado, la suma de éstos representaba la energía de cada segmento, por tanto cada uno iba a aportar cierto porcentaje de energía al segmento.

Éste se calculaba como se muestra en la ecuación 3.2

$$\%Ener(C_n) = \frac{C_n^2}{\sum_{i=0}^{N-1} C_i^2} \quad (\text{Ecuación 3.2})$$

Donde,  $C_n$  es el componente de la transformada de ondelete.

Estos porcentajes forman otra nueva matriz de las mismas dimensiones que las anteriores. Entonces se procedió a ordenar cada columna de ésta, en base a los porcentajes de manera descendente. Es decir, de una columna se escogió el componente de mayor porcentaje y se colocó en la primera fila, luego el del segundo mayor porcentaje se colocó en la segunda fila y así sucesivamente.

Con cada columna ordenada de esta manera, convenía hacer un arreglo para poder saber qué porcentaje de energía había conforme se iban agregando componentes. Para saber esto, se utilizó la matriz anterior, y en cada columna se puso el primer componente, luego a éste se le sumó el segundo componente y se colocó en la segunda fila y así sucesivamente hasta llegar a la última, donde el valor de ésta tenía que ser 100%. Ésta es la matriz que se obtenía de este bloque, la cuál será nombrada matriz de porcentajes acumulados.

### 3.3.2.3. Componentes notables

Esta parte se encarga de crear una nueva matriz de porcentajes, no va a tener las mismas dimensiones que las anteriores y está conformada de porcentajes, tiene una columna en donde se colocan los siguientes porcentajes de energía: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 96, 97, 98, 99, 100 por ciento. Para el resto de columnas se utiliza la matriz del bloque anterior y una columna en donde se colocan los porcentajes de la cantidad de componentes.

La columna de los porcentajes de la cantidad de componentes, se obtuvo fácilmente de la siguiente manera; si por ejemplo se tiene segmentos de 32 768 muestras en total, la primera fila va a tener un porcentaje de  $(1/32\ 768)*100$ , la segunda  $(2/32\ 768)*100$ , la tercera  $(3/32\ 768)*100$ , y así sucesivamente. El tamaño de esta columna va a ser de: número de filas de la matriz de porcentajes acumulados por 1.

El siguiente paso fue tomar la matriz de componentes acumulados y la columna de porcentaje de la cantidad de componentes notables, entonces por columna pasar fila por fila hasta que su valor fuera mayor o igual que el 5% de la energía del segmento, para llegar a este punto se tenía que pasar por cierto número de filas, y se pasaba por el mismo número de filas en la columna de porcentajes de la cantidad de componentes notables, para escoger el valor. Este valor se colocaba en la nueva matriz, dependiendo la columna a la que pertenecía. Se hizo lo mismo para el 10% y así sucesivamente hasta llegar al 100%.

Entonces, la nueva matriz iba a tener las siguientes dimensiones: 24 por número de columnas más 1, este 24 viene de los porcentajes de energía que se iban a representar.

Entonces la matriz se va a llenar de los porcentajes de la cantidad de componentes notables, que se necesitan para llegar a cierto porcentaje de energía. Por ejemplo si se tuvieran 5 segmentos de una canción, la matriz de este bloque queda como se muestra en la tabla IV.

**Tabla IV. Ejemplo de matriz de bloque**

Segmento 1 (%)	Segmento 2 (%)	Segmento 3 (%)	Segmento 4 (%)	Segmento 5 (%)	Energía (%)
2	0,2	0,001	0,5	1	5
5	0,5	0,002	1	2	10
8	1	0,005	1,5	3	15
13	2	0,01	2	4	20
17	3	0,02	2,5	5	25
21	5	0,05	3	6	30
25	7	0,1	3,5	7	35
29	9	0,3	4	10	40
33	11	0,5	5	13	45
37	14	0,9	6	16	50
41	17	1,2	7	19	55
45	21	2	8	22	60
49	24	3	9	25	65
53	28	4	12	30	70
57	32	6	15	35	75
62	36	8	18	40	80
67	40	10	21	45	85
72	45	12	30	50	90
77	50	15	39	60	95
80	53	18	45	65	96
82	56	21	51	70	97
84	59	25	57	75	98
87	65	30	63	80	99
90	75	50	70	90	100

Fuente: elaboración propia

### 3.3.2.4. Almacenamiento en archivo CSV

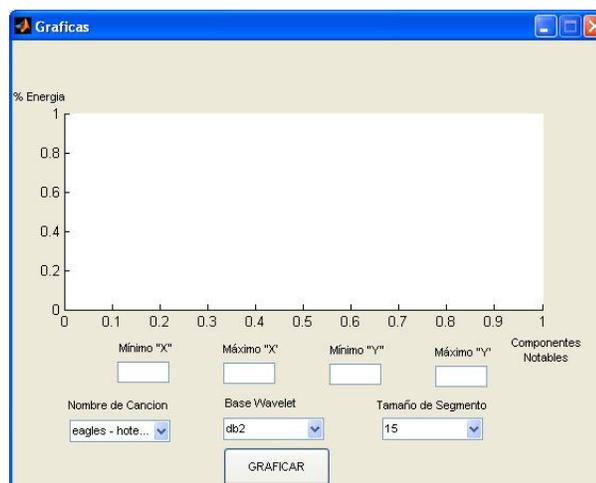
En este bloque se almacena una tabla parecida a la mostrada en la Tabla IV, en un archivo con extensión CSV, con la media y la desviación estándar de los todos segmentos por fila.

Cada archivo es nombrado con el título de la canción, el tamaño del segmento y el nombre del Ondelete utilizado, para que a la hora de graficar se puedan distinguir.

### 3.3.2.5. Interfaz GUI para mostrar las gráficas

En la figura 20, se muestra, como quedó la interfaz para que el usuario pueda ver fácilmente las gráficas.

Figura 20. **Interfaz para gráficas de energía contra componentes notables**

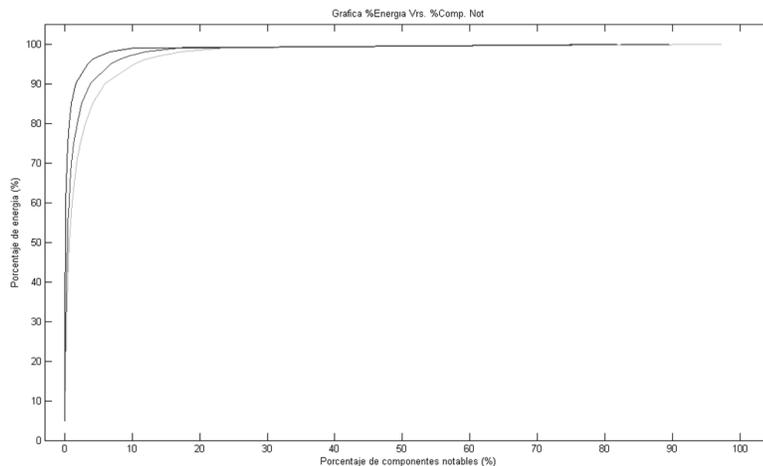


Fuente: elaboración propia

Como se puede ver, cuenta con un cuadro donde se grafica la energía contra los componentes notables, 4 cuadros editables que sirven para acercar o alejar el dibujo, 3 menús desplegables, 1 en el que se escoge la canción, otro en el que se escoge el ondelete, y otro en donde se escoge el tamaño del segmento. Finalmente, cuenta con un botón que se presiona, cuando se escogió todo, para que se grafique la energía contra los componentes notables.

En la figura 21, se muestra una gráfica que pertenece al valor promedio de todas las canciones, en la que se observan tres curvas que pertenecen a: la media, la media más la desviación estándar y la media menos la desviación estándar. En el apéndice 2 se muestran las tablas que se obtuvieron de distintas canciones.

Figura 21. **Gráfica de energía vrs. componentes notables**



Fuente: elaboración propia

La conclusión que se obtuvo de estas gráficas, es que se necesita aproximadamente el 10% de componentes notables, para que la energía de una canción sea de aproximadamente el 95% y que muestran con el siguiente ejercicio, que ondelete utilizar para la caracterización.

### **3.3.3. Ejercicio para gráficas de energía contra ondelete**

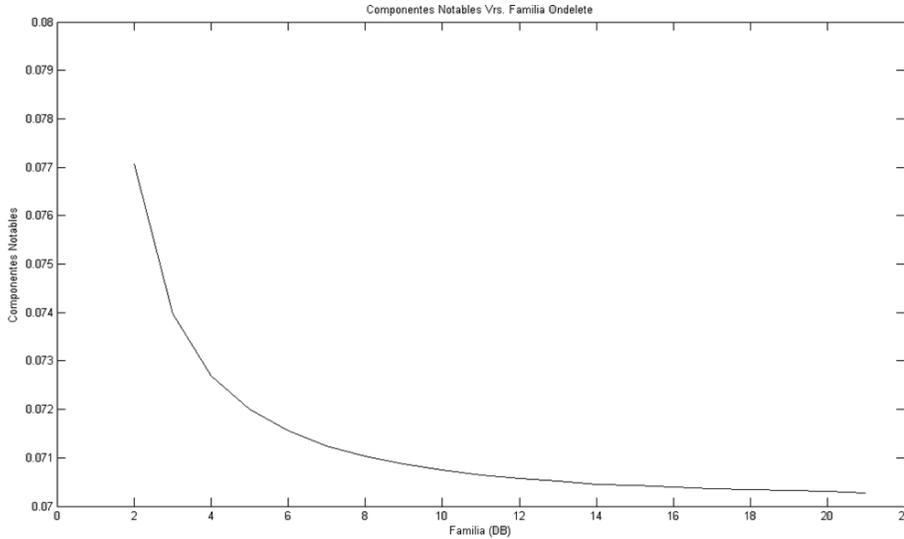
Este ejercicio sirvió para saber que ondelete se iban a usar, para esto se necesitaron los archivos CSV obtenidos de la parte anterior, que pertenecían a distintas canciones con distintos ondeletes utilizados para su creación.

En todos los archivos CSV, existen una columna que corresponde a los valores medios del porcentaje, de la cantidad de componentes notables de toda la canción, para las siguientes filas de porcentajes de energía: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 96, 97, 98, 99 y 100. Esta columna es la clave para obtener las gráficas de esta parte.

Con la columna de los valores medios de todos los archivos CSV, se obtuvo el valor medio de todas las energías, esto con el fin de saber cuál era el promedio de la cantidad de componentes notables entre todos los porcentajes de energía. Por cada canción hay 20 archivos CSV almacenados, cada uno está formado de distintos ondeletes, entonces con el valor medio obtenido y el ondelete al que pertenece se procede a graficar, poniendo para el eje de las “Y” el valor de energía promedio, y para el eje de las “X” el Ondelete utilizado.

En la figura 22, se muestra la gráfica promedio para todas las canciones que se analizaron.

Figura 22. **Gráfica de componentes notables vrs. familia de ondeletes**



Fuente: elaboración propia

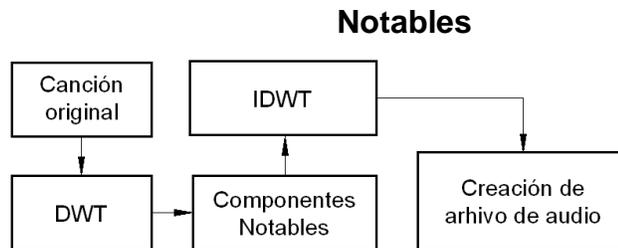
Esta gráfica, se obtiene de las tablas que se encuentran en el apéndice 3, en donde se muestra el valor promedio de todas las energías por base, y la razón de cambio de una base a otra.

De allí se obtiene que el ondelete más adecuado, es el DB12. Esto porque, de este ondelete al siguiente casi no cambia el porcentaje de energía y es el ondelete de menor orden que tiene esta característica.

### 3.3.4. Ejercicio para número de componentes notables

De este ejercicio se concluyó, que cantidad de componentes notables se necesitan, para poder reconocer una canción. La figura 23, muestra el diagrama de flujo.

Figura 23. **Diagrama de flujo de Ejercicio de número de Componentes**



Fuente: elaboración propia

A continuación se explica cada bloque.

#### **3.3.4.1. Canción**

En esta parte, se escogió una canción que se quería representar. Ésta se segmentó con trozos del mismo tamaño. Si al último segmento le hacían falta muestras, se descartaba y se tomaba en cuenta el resto de trozos.

Este bloque, entregaba una matriz en la que se colocan los componentes de cada segmento. Ésta tenía las siguientes dimensiones: número de muestras por segmento por número de segmentos de la canción. Los tamaños que se utilizaron para hacer las pruebas fueron de  $2^{12}$ ,  $2^{13}$ ,  $2^{14}$  y  $2^{15}$  muestras.

#### **3.3.4.2. DWT**

En este bloque, se procedió a obtener la transformada de ondelete discreta, para cada segmento de la matriz del bloque anterior. Esta parte entregaba una matriz del mismo tamaño que la anterior, sólo que con los componentes de la transformada ondelete. La transformada de ondelete que se utilizó, fue la DB12, para esto se utilizaron los resultados del ejercicio anterior.

### **3.3.4.3. Componentes notables**

Como parámetro de entrada se debía ingresar el número de componentes notables que se iban a utilizar para crear el archivo de audio.

Dependiendo del tamaño del segmento se hicieron varias pruebas, para los segmentos de  $2^{15}$  muestras, se probó con 200, 400, 600, 800, y 1 000 componentes, para los de  $2^{14}$  muestras, se utilizaron 150, 300, 450, 600 y 750 componentes, para los de  $2^{13}$  muestras, se utilizaron 100, 200, 300, 400, 500, y finalmente para los de  $2^{12}$  muestras, los mismos que los que se utilizaron para  $2^{13}$ . Para segmentos de  $2^{15}$  el valor de componentes adecuado fue de 600, para  $2^{14}$  fue de 500, para  $2^{13}$  fue de 400 y para  $2^{12}$  fue de 300, estos resultados se resumen en la tabla IV, que se explica en la parte de “Creación de archivos de audio” que pertenece a esta sección.

Ya sabiendo el número de componentes que se iban a utilizar, se procedió a ver cuáles eran los componentes más importantes. Para esto se elevó cada componente de la DWT al cuadrado. De aquí se escogieron los primeros  $s$  (número de componentes a utilizar) componentes que más energía poseían por segmento y se hicieron todos los demás iguales a 0.

Entonces cada columna que pertenecía a un segmento iba a tener  $s$  componentes distintos de cero.

### **3.3.4.4. IDWT**

En este bloque, se obtiene la transformada ondelete inversa de cada segmento obtenido de la parte anterior.

Para poder obtener la IDWT, se utilizaron las ecuaciones 2.25 y 2.26 para los coeficientes de los filtros de esta parte y se procedió a utilizar el algoritmo mostrado en la figura 13, para obtener los coeficientes de la señal a recomponer.

Aquí se volvió a crear otra matriz con las mismas dimensiones que las anteriores solo que con los componentes de la IDWT.

#### **3.3.4.5. Creación de archivo de audio**

Esta es la parte final del *software*, aquí se debe crear un archivo de audio con la matriz de IDWT. Esta matriz se debe convertir en un vector, esto se logra formando un vector columna, con número de filas igual a la multiplicación del número de muestras por la cantidad de segmentos.

Este vector se va llenando de la siguiente forma: se coloca la primer columna de la matriz de IDWT, en los primeros componentes de este vector, luego la segunda en los que le siguen y así sucesivamente.

Ya con el vector lleno, se utiliza la función de Matlab “wavwrite” para que convierta el vector en un archivo de audio con extensión “wav”. Se procedió a escuchar el archivo y concluir si se podía reconocer la canción y a analizar qué tanta interferencia existe.

Los archivos de audio, no pueden ser representados de forma escrita, por tanto necesitan ser escuchados para poder concluir en esta parte. La conclusión se muestra en la tabla IV. Ya con estos resultados y los del ejercicio anterior se procede a crear la base de datos.

**Tabla V. Componentes notables por tamaño**

<b>Tamaño</b>	<b>Número de componentes notables</b>
2 <sup>15</sup>	600
2 <sup>14</sup>	500
2 <sup>13</sup>	400
2 <sup>12</sup>	300

Fuente: elaboración propia

### **3.3.5. Creación de la base de datos**

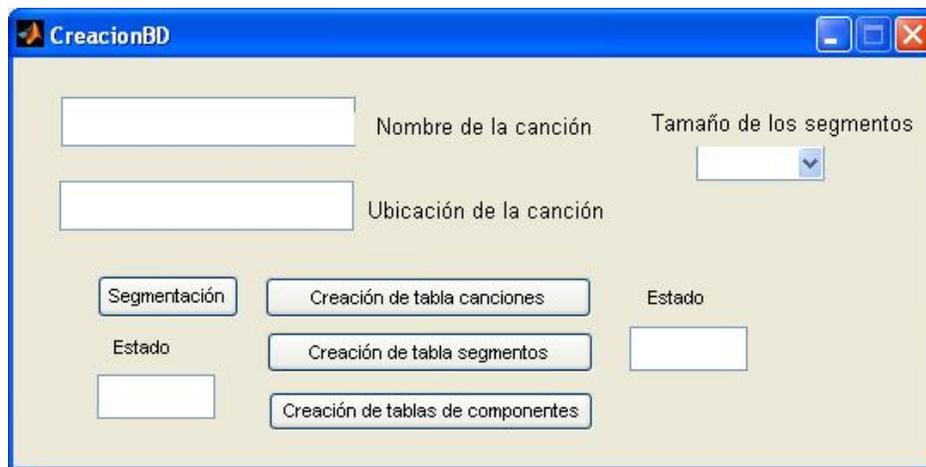
Hasta este punto, no se sabía que tamaño era el adecuado para la recomposición, es por eso que se crearon 4 bases de datos, esto por los distintos tamaños de los segmentos. Éstas, se conformaron de 4 tablas, estas tablas son los archivos “mat” que se crean con la función “save” de Matlab para almacenar variables.

La primera tabla, consistía de dos columnas, una en la que se colocó un identificador, y en la otra el nombre de las canciones a almacenar. La segunda estaba formada de 4 columnas, que se utilizaban para identificar el número de segmento, saber a qué canción pertenecían, dependiendo del identificador de la tabla anterior, ubicar el número de segmento de la canción de la que formaban parte, y colocar el valor de la energía de éste.

La tercera tabla estaba compuesta por los componentes notables de la transformada ondelete y por el identificador de segmento obtenido de la tabla anterior, y finalmente la cuarta tabla, estaba compuesta de los índices a los que pertenecían los componentes notables y el mismo identificador de segmento que la tabla anterior.

Se distribuyó de esta manera la información, para no tener solamente una tabla demasiado grande, en la que se almacenaba todo, y para que se pudiera ir a buscar fácilmente a cada tabla, los parámetros necesarios para recomponer una canción. Esto hace más eficiente el trabajo, porque los datos se cargan solo cuando es necesario. La interfaz gráfica que se utilizó para la creación de bases de datos se muestra en la figura 24.

Figura 24. **Interfaz gráfica de la creación de la base de datos**



Fuente: elaboración propia

Lo primero que se tiene que hacer es escribir el nombre de la canción a colocar en la base de datos y su ubicación exacta con una diagonal “\” al final, luego se escoge el tamaño de los segmentos, entre los que se encuentran  $2^{12}$ ,  $2^{13}$ ,  $2^{14}$  y  $2^{15}$ . Ya con esto elegido se procedía a presionar el botón que se llama Segmentación, éste se encargó de dividir la canción en segmentos y guardarlos en la carpeta que lleva el nombre de la canción que se encontraba en “Base de Datos”. Estos segmentos son los de la sección 3.2.

Ya con estos segmentos almacenados, se procedía a presionar el botón que se llama “Creación de tabla canciones”, éste tenía la función de cargar la tabla que se llama “Canciones” que pertenecía a la carpeta que tiene por nombre el número de la potencia de dos, que pertenece al tamaño de los segmentos. Ya con la tabla cargada se procedía a revisar cuál era el identificador de la última canción almacenada, a éste se le sumaba uno y se grababa en la tabla este número y el nombre de la canción.

El botón “Creación de tabla segmentos” tenía la función de cargar la tabla de nombre “Segmentos”, que pertenecía a la misma carpeta donde estaba ubicada la tabla anterior, luego se procedía a revisar cuál era el identificador del último segmento almacenado, se contaban cuantos segmentos pertenecían a la canción, y se creaba una matriz del número de segmentos por 4.

En la primera columna se colocaban los números desde el último valor del identificador almacenado, más uno hasta la cantidad de segmentos. En la segunda columna se colocaba el identificador de la canción a la que pertenecía. En la tercera se ponían el nombre del segmento, es decir los números desde el uno hasta la cantidad de trozos, finalmente para la cuarta columnas se leían todos los segmentos y se obtenía la energía con la suma de los cuadrados de cada componente. Esta matriz se agregaba a la que ya estaba creada en la tabla y la matriz conjunta se almacenaba de nuevo.

Finalmente el botón de tabla de componentes tenía la tarea de llenar la tabla de “Componentes Notables” y la de “Índices”, cuando se presionaba este botón se cargaban las dos tablas que se ubicaban en la misma dirección que las anteriores, para este botón existía una variable de entrada que era el identificador de segmentos que pertenecía al primero de la canción, ésta se obtenía como parámetro de salida, cuando se presionaba el botón anterior.

Se creaba un vector columna que tenía como primer parámetro a este identificador y conforme se iba pasando en el vector, se le iba sumando uno hasta llegar al último segmento. El vector columna iba a tener las siguientes dimensiones: número de segmentos por 1. Luego se creaban dos matrices de ceros que tenían las siguientes dimensiones: número de segmentos por número de componentes notables. Dependiendo el tamaño del segmento, así variaba el número de componentes notables que se obtuvo del ejercicio anterior.

Con estas matrices creadas, se procedía a obtener la transformada de ondelete DB12 de todos los segmentos, por cada segmento se sacaban los componentes más importantes, con sus índices respectivos en base a su energía. Luego de esto se procedía a normalizar con la norma 2, los componentes importantes de la transformada ondelete, por segmento y éstos se almacenaban en las filas de las matrices de ceros creadas antes, ya con estas matrices llenas, se agregaron a las matrices ya almacenadas de las tablas “Componentes Notables” e “Índices”. De la misma manera se agregaba el vector columna de los identificadores de los segmentos a la columna almacenada que pertenecía a éstos.

De las matrices mencionadas arriba, se procedía a ordenarlas por fila en base a su índice, para esto se tomaba la matriz de “Índices” y al final se le agregaba la columna de identificadores de segmento.

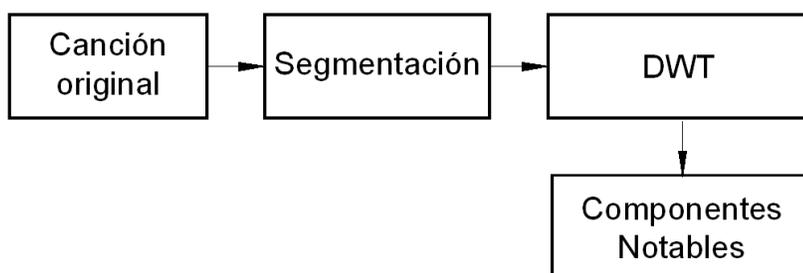
Luego se utilizaba la función “*sortrows*” de Matlab y finalmente se tomaba las columnas de la primera a la penúltima, y todas las filas para formar de nuevo la tabla “Índices” y la última columna era la que formaba el vector columna, de los identificadores de segmentos.

Luego se creaba un ciclo en el que se iban a tomar en cuenta todas las filas del vector columna de los indicadores de segmentos, que pertenecía a la tabla “Índices” y éstos se iban a ir a buscar a los de la tabla “Componentes Notables” entonces se tomaba toda la fila y se ubicaba en otra matriz, de las mismas dimensiones en la fila que correspondía al valor del ciclo que se estaba tomando en cuenta. Finalmente se guardaba cada matriz en su tabla correspondiente y el vector columna en las dos tablas.

### 3.4. Canción original

Con este bloque se iniciaba el proceso de recomposición como se observa en la figura 5, pero antes de esto se debieron haber creado los dos bloques anteriores, la función principal de éste, es segmentar la señal que se quiere recomponer, obtener su transformada de ondelete y sacar los componentes más importantes en base a la energía, por segmento. La figura 25, muestra el diagrama de bloques de esta parte.

Figura 25. **Diagrama de bloques de canción original**



Fuente: elaboración propia

A continuación se explica cada bloque:

### **3.4.1. Canción original**

Aquí se tomó una canción que se quería recomponer, que fuera de extensión “wav” o “au”, el archivo se leía por medio de las funciones de Matlab “wavread” o “wavau”.

Ya guardada en una variable en Matlab, se verificó que ésta perteneciera a un sonido monofónico, de lo contrario se procedió a convertirlo de estereofónico a monofónico. Para lograr hacer esto se sumaron los componentes por fila y se dividieron dentro de 2.

### **3.4.2. Segmentación**

En este bloque, se utilizó el vector monofónico, y se segmentó en trozos del mismo tamaño. Con estos trozos se formó una matriz de las siguientes dimensiones: número de muestras por número de segmentos. En esta matriz se tenían almacenados todos los segmentos que pertenecían a la canción en cada columna.

### **3.4.3. DWT**

En esta parte, se obtuvo la transformada de ondelete DB12 de cada segmento, esto resultó en otra matriz con las mismas dimensiones que las de la anterior sólo que en lugar de las muestras, se colocaron los componentes de la transformada.

#### **3.4.4. Componentes notables**

De la matriz que brindó DWT, se obtuvieron los componentes más importantes en base a su energía. El número de éstos variaba dependiendo el tamaño del segmento y se utilizaron los componentes de la tabla V, para segmentos de  $2^{15}$  muestras se utilizaron 600 componentes importantes, para los de  $2^{14}$  muestras, 500, para los de  $2^{13}$  muestras, 400 y finalmente para los de  $2^{12}$  muestras, 300.

En la matriz, estos componentes importantes se mantuvieron y todos los demás se hicieron iguales a 0. Entonces, dependiendo del tamaño del segmento, las matrices que se obtuvieron de éste bloque, tenían dimensiones de: número de muestras por número de segmentos; y en cada columna se mantenía 600 ó 500 ó 400 ó 300 componentes distintos de cero, que eran los más importantes según fuera el caso del tamaño del segmento.

Hay que tomar en cuenta, que la transformada de ondelete utilizada fue la familia de Daubechies DB12, que fue la misma que se utilizó para formar la base de datos, además los tamaños de segmentos fueron los mismos que los que se usaron para crear la base de datos, y el número de componentes importantes también fue el mismo para todos los casos.

Es importante tomar en cuenta esto, porque los tamaños tienen que coincidir, porque si no, la canción recompuesta no se podría crear, ya que no se podría colocar un vector de  $2^{15}$  en uno de  $2^{12}$ , además, las características obtenidas con esta familia de ondeletes específica, puede variar mucho si se obtuvieran de otra familia. El número de componentes importantes, es esencial porque de éste se desprende la legibilidad de la canción al ser escuchada.

### **3.5. Búsqueda y reconocimiento**

Este bloque, se encargaba de buscar en la base de datos cuáles son los segmentos que pueden representar de la mejor manera, a un trozo de la pieza original.

Esta parte, utilizó el algoritmo de “Búsqueda de Emparejamiento” como su principal herramienta de trabajo, y los bloques de “Canción Original” y “Base de Datos” le brindaron la información necesaria para realizar la búsqueda.

El bloque “Canción Original” se encargaba de brindarle una matriz, en la que se encontraban los componentes importantes de la transformada de ondelete DB12, de cada segmento de la pieza que se quería recomponer. La transformada de Ondelete de cada trozo, iba a tener el mismo número de muestras que el original y dependiendo del tamaño, las columnas de esta matriz podían tener 600 ó 500 ó 400 ó 300 componentes distintos de 0.

En el bloque “Base de Datos” se encontraba toda la información acerca de los componentes notables de cada segmento, que formaba parte de ésta. Aquí se encontraban matrices con vectores comprimidos, es decir, por ejemplo, si los segmentos eran de  $2^{15}$  muestras, entonces cada fila iba a tener únicamente 600 componentes, y existían 2 matrices en una se colocaban los valores de la transformada y en la otra los índices a los que pertenecían cada uno de éstos.

Lo primero que hacía el bloque, era tomar los componentes notables con sus índices respectivos del primer segmento de la matriz que brindaba “Canción Original”, y con éstos ir a buscar a la base de datos los vectores que tuvieran los mismos signos, de los componentes cuando sus índices fueran iguales, es decir, si uno de los componentes notables del segmento de la pieza original, se encontraba en el índice 0 y era positivo, entonces en la base de datos se escogían los vectores que tenían un componente notable en el índice 0 que fueran positivos ó 0, luego se hacía lo mismo para el segundo componente notable y así, hasta llegar al último.

De aquí quedaba un grupo menor de vectores, ya con este grupo se procedía a obtener los productos punto del vector de la pieza original, con los del grupo de vectores comprimidos que se habían escogido de la base de datos.

Para obtener el producto punto se llenaba una matriz de ceros, de las dimensiones del grupo de vectores comprimidos, se escogía los índices y componentes de éstos y se multiplicaba cada uno por los componentes que correspondían al mismo índice del vector original. Estas multiplicaciones se colocaban en la posición que les correspondía según estaban ordenados en la base de datos en los vectores de la matriz de ceros. Al final se sumaban todos los componentes por fila de esta matriz, y esta suma era el producto punto.

Luego de haber obtenido lo anterior, se buscaba cuál segmento era el que tenía el mayor producto punto. Éste era el más parecido al original, entonces se expandía el vector escogido y se multiplicaba por el valor del producto punto, luego se le restaba este vector al original y se obtenía uno nuevo, que era tomado como el nuevo vector original y se volvía a hacer todo el procedimiento por 10 veces.

Luego de haber escogido los 10 segmentos que más se parecían, se procedía a tomar el segundo segmento de “Canción Original”, y se volvía a hacer lo mismo hasta que se terminaba con el último trozo de la pieza a recomponer.

Al final de este bloque teníamos, por trozo; 10 identificadores de segmentos de la base de datos y los valores por los que había que multiplicar cada uno, es decir, los productos punto.

### **3.6. Composición**

Ésta, era la parte final del *software*, su función principal era tomar los datos obtenidos del bloque anterior, e ir a buscar los segmentos al bloque “Sonidos Originales de la Base de Datos”, ya con esto recomponer la pieza original.

Este bloque utilizaba los identificadores que se obtenían de la parte de “Búsqueda y Reconocimiento”, para ir a buscar a las tablas “Canciones” y “Segmentos” los trozos musicales a los que pertenecían. Para hacer esto, primeramente el *software* utilizaba los identificadores y los iba a buscar a la primera columna de los datos obtenidos de la tabla “Segmentos”, y revisaba en que filas se encontraban estos valores. De estas filas escogidas obtenía el identificador de canción, el número de segmento al que pertenecían y la energía de cada segmento. A continuación, el *software* utilizaba estos identificadores de canción para ir a buscar a la tabla “Canciones”, el nombre de las canciones que coincidían con los números de identificador escogidos.

Con esta información recolectada, el *software* apuntaba hacia la carpeta de nombre “Base de Datos”, y buscaba las carpetas que correspondían con los nombres de las canciones elegidas, se ingresaba a estas carpetas y se leían los segmentos escogidos, de cada carpeta con la función de Matlab “wavread”.

Con los segmentos convertidos en vectores se procedía a multiplicarlos por el valor del producto punto, pero este valor era para los componentes normalizados, por tanto se necesitaba hacer otra multiplicación, cómo se escogían 10 segmentos a los que se nombraron S1, S2, S3, S4, S5, S6, S7, S8, S9 y S10 cuyas transformadas de ondeletes normalizadas se representaban por; W1, W2, W3, W4, W5, W6, W7, W8, W9 y W10, los valores de los productos puntos obtenidos eran  $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9$  y  $\alpha_{10}$ , y las energías de cada segmento era E1, E2, E3, E4, E5, E6, E7, E8, E9 y E10, entonces para representar el vector de transformada de ondelete normalizado había que obtenerlo con la ecuación 3.3.

$$W1 = \text{sqrt}\left(\frac{1}{E1}\right) * W_{nn}1 \quad (\text{Ecuación 3.3})$$

Así fue como se obtuvieron los componentes normalizados de la transformada de ondelete, entonces la transformada de ondelete del vector original es  $W_{nn}0$  y es igual a:

$$W_{nn}0 = \alpha_1 * W1 + \alpha_2 * W2 + \alpha_3 * W3 + \alpha_4 * W4 + \alpha_5 * W5 + \alpha_6 * W6 + \alpha_7 * W7 + \alpha_8 * W8 + \alpha_9 * W9 + \alpha_{10} * W10 \quad (\text{Ecuación 3.4})$$

Pero esta ecuación, tenía los vectores normalizados y los que se obtuvieron, fueron los vectores no normalizados.

La ecuación 3.5 representa como debe ser formado el vector ondelete de la pieza original:

$$\begin{aligned}
 W_{nn}0 = & \alpha_1 * \text{sqrt}\left(\frac{1}{E1}\right) W_{nn}1 + \alpha_2 * \text{sqrt}\left(\frac{1}{E2}\right) W_{nn}2 + \alpha_3 * \text{sqrt}\left(\frac{1}{E3}\right) W_{nn}3 + \alpha_4 * \\
 & \text{sqrt}\left(\frac{1}{E4}\right) W_{nn}4 + \alpha_5 * \text{sqrt}\left(\frac{1}{E5}\right) W_{nn}5 + \alpha_6 * \text{sqrt}\left(\frac{1}{E6}\right) W_{nn}6 + \alpha_7 * \\
 & \text{sqrt}\left(\frac{1}{E7}\right) W_{nn}7 + \alpha_8 * \text{sqrt}\left(\frac{1}{E8}\right) W_{nn}8 + \alpha_9 * \text{sqrt}\left(\frac{1}{E9}\right) W_{nn}9 + \alpha_{10} * \\
 & \text{sqrt}\left(\frac{1}{E10}\right) W_{nn}10 \quad (\text{Ecuación 3.5})
 \end{aligned}$$

Y como la transformada de ondelete tiene la propiedad de ser lineal, entonces la señal original puede ser representada por:

$$\begin{aligned}
 S0 = & \alpha_1 * \text{sqrt}\left(\frac{1}{E1}\right) S1 + \alpha_2 * \text{sqrt}\left(\frac{1}{E2}\right) S2 + \alpha_3 * \text{sqrt}\left(\frac{1}{E3}\right) S3 + \alpha_4 * \text{sqrt}\left(\frac{1}{E4}\right) S4 + \\
 & \alpha_5 * \text{sqrt}\left(\frac{1}{E5}\right) S5 + \alpha_6 * \text{sqrt}\left(\frac{1}{E6}\right) S6 + \alpha_7 * \text{sqrt}\left(\frac{1}{E7}\right) S7 + \alpha_8 * \text{sqrt}\left(\frac{1}{E8}\right) S8 + \alpha_9 * \\
 & \text{sqrt}\left(\frac{1}{E9}\right) S9 + \alpha_{10} * \text{sqrt}\left(\frac{1}{E10}\right) S10 \quad (\text{Ecuación 3.6})
 \end{aligned}$$

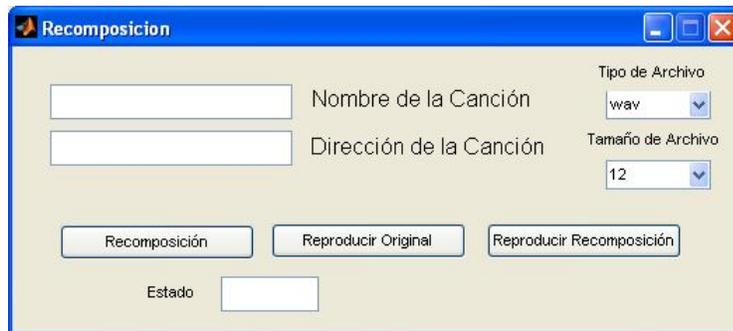
Con esta ecuación se aproximaba cada segmento de la pieza original, esto se hacía para todos los trozos y para cada uno quedaba un vector de respuesta que era la suma de todos los vectores.

Finalmente, el *software* creaba un vector columna de ceros, con dimensiones: número de muestras por número de segmentos por 1, entonces se divide este vector en trozos del tamaño de forma lógica, el primer trozo lógico se llena con el vector resultante de la ecuación 3.6, para el primer segmento, el segundo con el vector resultante de la ecuación 3.6, para el segundo segmento y así sucesivamente, hasta llenar todos los trozos. Luego de llenar los trozos se procede a escribir la canción resultante, con la función de Matlab “Escribir canción” y se coloca en una carpeta de nombre “Resultados”.

### 3.7. Interfaz gráfica

El *software* final se presentó como se muestra en la figura 26.

Figura 26. Interfaz gráfica de recomposición



Fuente: elaboración propia

Como se puede ver, la interfaz gráfica era bastante sencilla, contaba con 2 campos editables llamados “Nombre de la Canción” y “Dirección de la canción”, 3 botones denominados “Recomposición”, “Reproducir Original”, “Reproducir Recomposición”, 2 menús desplegables nombrados “Tipo de Archivo” y “Tamaño de Archivo” y finalmente un cuadro de Estado.

#### 3.7.1. Funcionamiento

Para recomponer una canción, se escribía el nombre y su ubicación en los cuadros editables, se escogía que tipo de archivo era el que se quería descomponer, solamente dos tipos se podían utilizar, los “wav” o los “au”. Luego se procedía a elegir el tamaño de segmentos, que se quería para la recomposición entre  $2^{15}$ ,  $2^{14}$ ,  $2^{13}$  y  $2^{12}$ . Con todo esto elegido se procedía a presionar el botón “Recomposición”, cuya función era que por medio de las herramientas, que se explicaron en este capítulo recompusiera una canción.

Conforme el *software* iba recomponiendo la canción, le iba informando al usuario en el cuadro “Estado”, que era lo que estaba realizando. Cuando terminaba de recomponer la canción, aparecía la palabra “Realizado. Si este cuadro tenía esta palabra se podía presionar el botón de “Reproducir Recomposición”.

El botón “Reproducir Original” se encarga de ir a buscar a la carpeta especificada en “Dirección”, el archivo puesto en el cuadro “Nombre”, convertir el archivo en un vector y luego utilizar la función de Matlab “Sonido” para reproducirlo.

El botón “Reproducir Recomposición” se encarga de ir a la carpeta “Resultados” a buscar el archivo especificado en el cuadro “Nombre”, convertirlo en un vector y reproducirlo.

Para poder hacer una comparación entre ambos archivos, se necesita escucharlos, y para esto sirven estos botones.

Para saber que tamaño era el adecuado, se recompusieron distintas canciones a distintos tamaños, y se concluyó que el tamaño adecuado era de  $2^{12}$  muestras.



## CONCLUSIONES

1. La transformada de ondelete, por su forma de construcción con filtros de Q constante, caracteriza a las señales muy bien en tiempo y en frecuencia, lo que la hace muy útil, como herramienta de análisis y comparación para señales de audio.
2. La indexación significativa y compacta de segmentos de audio, puede lograrse con los índices derivados de los componentes notables de la transformada de ondelete y los componentes mismos, ya que ambas cosas se utilizan para comparar las señales de los segmentos originales con los de la base de datos.
3. Entre la familia de Ondeletes Daubechies, la DB12 fue la que se escogió para caracterizar las señales de audio, porque, como se muestra en las tablas del apéndice 3, a partir de ésta, los valores promedio de todas las canciones no variaban demasiado.
4. Para la recomposición musical, el tamaño de los segmentos que mejor se adecua a las canciones es de  $2^{12}$  (4 096) muestras, debido a que las piezas musicales, cuentan con cambios de notas muy rápidos, y este valor, según los archivos escuchados es bueno para la representación.
5. Con los sonidos de la naturaleza se puede representar la mayor parte de notas musicales, creadas por instrumentos.

6. Según experimentos, utilizando 10 vectores de un diccionario variado y amplio, y el algoritmo búsqueda de emparejamiento, se puede representar un vector. Sin embargo, en los experimentos realizados por el Dr. Juan Carlos Córdova, él verificó que con 5 segmentos no se pierde el reconocimiento.
7. La indexación de segmentos, es una alternativa para medir la similitud entre dos señales computacionalmente más económica que la correlación.
8. Experimentalmente encontramos que un grupo de segmentos, puede representar cualquier sonido con 5 octavas y media del rango de un piano.
9. Con la transformada de ondelete, se necesita un poco menos del 10% de los componentes para representar de forma inteligible una pieza musical.
10. Este trabajo se puede representar como un arte musical, en que se emplean sonidos de la naturaleza para recomponer una pieza musical.

## RECOMENDACIONES

1. En un artículo de la IEEE nombrado “Diseño eficiente de bases Ortonormales para la representación de señales” de Jian-Kang Zhang, Timothy N. Davidson<sup>23</sup>, se explica una forma de crear Ondeletes que se adecuen a las señales, por tanto es probable que el número de componentes notables disminuya debido a la elección de una mejor familia de Ondeletes.
2. Se puede investigar la cuantización de los componentes notables, tarea que puede llevar a la compresión de archivos de audio.
3. Los Ondeletes también se pueden utilizar para imágenes, entonces se puede investigar acerca de esto, para lograr algo similar a los cuadros de Octavio Ocampo para que ayude al arte de imágenes.
4. Como se puede ver, en el presente únicamente se utilizan tamaños de segmentos fijos, razón por la cual se puede explorar el hecho de tener segmentación de longitud variable o quizás transiciones entre segmentos. Esto podría servir para resolver problemas como el que existe en canciones con cambios de notas demasiado rápidos por ejemplo la Novena Sinfonía de Beethoven.

---

<sup>23</sup> ZHANG, Jian-Kang; DAVIDSON, Timothy N. “*Efficient Design of Orthonormal Wavelet Bases for Signal Representation*”. IEEE Transacción en procesado de señales. 2004, vol. 52, num. 7, pp. 1983-1996.



## BIBLIOGRAFÍA

1. MALLAT, S. G. y ZHANG, Z. "Matching Pursuit with Time-Frequency Dictionaries". *IEEE Transacción en procesado de señales*. 1993, vol. 41, num. 12, 18 p.
2. MIX, Dwight; OLEJNICZAK, Kraig. *Elements of Wavelets for Engineers and Scientists*. Estados Unidos: Willey Interscience, 2003. 236 p. ISBN: 0-471-46617-4.
3. NILSSON, James W. *Circuitos eléctricos*. 4a. ed. Massachusetts: Eddison-Wesley, 1995. 981 p.
4. OPPENHEIM, Alan V.; SCHAFFER, Ronald W.; BUCK, John. *Tratamiento de señales en tiempo discreto*. 2a. ed. México: Prantice Hall. 863 p.
5. RÍOS RIVAS, Manuel Alejandro. "La transformación Wavelet en el procesado digital de audio". Facultad de Ingeniería: Ingeniería Electrónica. Universidad de San Carlos de Guatemala. 2009. 326 p.
6. SARKAR, Tapan K.; SU, C. "A tutorial on Wavelets from an Electrical Engineering Perspective, part 1: Discrete Case". *IEEE Antenas y propagación*. 1998, vol. 40, num. 60, 30 p.

7. SARKAR, Tapan K.; SU, C. "A tutorial on Wavelets from an Electrical Engineering Perspective, part 2: Continuous Case". *IEEE Antenas y propagación*. 1998, vol. 40, num. 60, 15 p.
8. STRANG, Gilbert. *Maxflat filters: Daubechies and Meyer formulas. Spectral Factorization* [en línea]. MIT Open course ware, Course 18.327 Wavelets and Filter Banks. Disponible en Web: <<http://ocw.mit.edu/courses/mathematics/18-327-wavelets-filter-banks-and-applications-spring-2003/lecture-notes/Slides8.pdf>>.
9. TAUB, Herbert; SCHILLING, Donald. *Principles of Communication Systems*. 2a. ed. New York: McGraw Hill, 1986. 759 p.
10. THE MATHWORKS. Image Processing Toolbox™ 7 User's Guide [en línea]. Estados Unidos. 1997. Revisión: marzo 2010 [Consulta: junio 2010]. Disponible en Web: <<http://www.mathworks.com/help/toolbox/images/bqj5b2j.html>>.
11. WIKIPEDIA, The Free Encyclopedia. Piano Key Frecuencias. 2003. Revisión: Marzo 2010. [Consulta: Abril 2010]. Disponible en Web: <[http://en.wikipedia.org/wiki/Piano\\_key\\_frequencies](http://en.wikipedia.org/wiki/Piano_key_frequencies)>.
12. ZHANG, Jian-Kang; DAVIDSON, Timothy N. "Efficient Design of Orthonormal Wavelet Bases for Signal Representation". *IEEE Transacción en procesado de señales*. 2004, vol. 52, num. 7, 13 p.

**APÉNDICE 1**  
**TABLAS DE EXCEL DEL EJERCICIO PARA ELECCIÓN DEL**  
**TAMAÑO ADECUADO**



## Tamaño para canción No.1

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	7	Correcto	0,5
A	5	Correcto	1
A	2	Correcto	1,5
A	1	Correcto	2
A	1	Correcto	2,5
B	4	Correcto	0,5
B	3	Correcto	1
B	1	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	4	Incorrecto	0,5
C	2	Incorrecto	1
C	1	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	5	Correcto	0,5
D	3	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	6	Incorrecto	0,5
E	4	Incorrecto	1
E	3	Incorrecto	1,5
E	1	Correcto	2
E	1	Correcto	2,5
F	3	Correcto	0,5
F	2	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	6	Incorrecto	0,5
G	3	Correcto	1
G	1	Incorrecto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	3	Correcto	0,5
H	2	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,203</b>			

## Tamaño para canción No. 2

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	6	Incorrecto	0,5
A	6	Incorrecto	1
A	5	Correcto	1,5
A	3	Correcto	2
A	1	Correcto	2,5
B	6	Correcto	0,5
B	4	Correcto	1
B	3	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	3	Correcto	0,5
C	3	Correcto	1
C	2	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	7	Incorrecto	0,5
D	6	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	9	Incorrecto	0,5
E	7	Incorrecto	1
E	6	Incorrecto	1,5
E	5	Incorrecto	2
E	1	Correcto	2,5
F	3	Correcto	0,5
F	2	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	5	Correcto	0,5
G	3	Correcto	1
G	2	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	2	Correcto	0,5
H	2	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,271</b>			

### Tamaño para canción No. 3

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	8	Correcto	0,5
A	5	Correcto	1
A	5	Correcto	1,5
A	3	Correcto	2
A	1	Correcto	2,5
B	4	Correcto	0,5
B	2	Correcto	1
B	1	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	3	Correcto	0,5
C	2	Correcto	1
C	1	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	2	Correcto	0,5
D	2	Correcto	1
D	1	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	5	Correcto	0,5
E	5	Correcto	1
E	3	Correcto	1,5
E	3	Correcto	2
E	1	Correcto	2,5
F	2	Correcto	0,5
F	2	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	3	Correcto	0,5
G	2	Correcto	1
G	1	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	2	Correcto	0,5
H	2	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,194</b>			

### Tamaño para canción No. 4

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	9	Incorrecto	0,5
A	7	Incorrecto	1
A	7	Correcto	1,5
A	7	Correcto	2
A	1	Correcto	2,5
B	8	Correcto	0,5
B	3	Incorrecto	1
B	5	Correcto	1,5
B	3	Correcto	2
B	3	Correcto	2,5
C	6	Correcto	0,5
C	4	Correcto	1
C	2	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	5	Incorrecto	0,5
D	5	Correcto	1
D	3	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	8	Incorrecto	0,5
E	8	Incorrecto	1
E	5	Incorrecto	1,5
E	5	Correcto	2
E	5	Correcto	2,5
F	4	Correcto	0,5
F	3	Correcto	1
F	2	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	4	Correcto	0,5
G	2	Correcto	1
G	1	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	5	Correcto	0,5
H	3	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,384</b>			

### Tamaño para canción No. 5

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	5	Correcto	0,5
A	4	Correcto	1
A	3	Correcto	1,5
A	1	Correcto	2
A	1	Correcto	2,5
B	2	Correcto	0,5
B	2	Correcto	1
B	1	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	3	Correcto	0,5
C	2	Correcto	1
C	1	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	3	Correcto	0,5
D	3	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	4	Correcto	0,5
E	4	Correcto	1
E	1	Correcto	1,5
E	1	Correcto	2
E	1	Correcto	2,5
F	6	Correcto	0,5
F	4	Correcto	1
F	2	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	3	Correcto	0,5
G	2	Correcto	1
G	2	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	5	Correcto	0,5
H	3	Correcto	1
H	2	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,135</b>			

### Tamaño para canción No. 6

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	5	Incorrecto	0,5
A	5	Incorrecto	1
A	3	Correcto	1,5
A	1	Correcto	2
A	1	Correcto	2,5
B	3	Correcto	0,5
B	2	Correcto	1
B	2	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	4	Incorrecto	0,5
C	3	Correcto	1
C	3	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	5	Correcto	0,5
D	3	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	9	Incorrecto	0,5
E	9	Incorrecto	1
E	3	Correcto	1,5
E	1	Correcto	2
E	1	Correcto	2,5
F	3	Incorrecto	0,5
F	3	Correcto	1
F	2	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	5	Correcto	0,5
G	4	Correcto	1
G	3	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	4	Correcto	0,5
H	3	Correcto	1
H	2	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,302</b>			

### Tamaño para canción No. 7

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	7	Incorrecto	0,5
A	5	Incorrecto	1
A	6	Correcto	1,5
A	3	Correcto	2
A	1	Correcto	2,5
B	5	Correcto	0,5
B	3	Correcto	1
B	2	Correcto	1,5
B	2	Correcto	2
B	1	Correcto	2,5
C	4	Correcto	0,5
C	2	Correcto	1
C	2	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	3	Correcto	0,5
D	2	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	7	Correcto	0,5
E	4	Correcto	1
E	3	Incorrecto	1,5
E	2	Correcto	2
E	1	Correcto	2,5
F	3	Correcto	0,5
F	2	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	4	Correcto	0,5
G	3	Correcto	1
G	2	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	3	Correcto	0,5
H	3	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia Promedio=1,208</b>			

### Tamaño para canción No. 8

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	5	Correcto	0,5
A	5	Correcto	1
A	3	Correcto	1,5
A	2	Correcto	2
A	1	Correcto	2,5
B	3	Correcto	0,5
B	3	Correcto	1
B	2	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	7	Correcto	0,5
C	5	Correcto	1
C	3	Correcto	1,5
C	2	Correcto	2
C	1	Correcto	2,5
D	4	Correcto	0,5
D	3	Correcto	1
D	2	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	4	Correcto	0,5
E	2	Correcto	1
E	2	Correcto	1,5
E	2	Correcto	2
E	1	Correcto	2,5
F	3	Correcto	0,5
F	3	Correcto	1
F	2	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	5	Correcto	0,5
G	5	Correcto	1
G	3	Correcto	1,5
G	2	Correcto	2
G	1	Correcto	2,5
H	3	Correcto	0,5
H	3	Correcto	1
H	2	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,162</b>			

### Tamaño para canción No. 9

Usuario	Cantidad de segmentos escuchados	Correcto Incorrecto	Tamaño del segmento (segundos)
A	7	Correcto	0,5
A	7	Correcto	1
A	5	Correcto	1,5
A	3	Correcto	2
A	1	Correcto	2,5
B	5	Correcto	0,5
B	3	Correcto	1
B	3	Correcto	1,5
B	2	Correcto	2
B	1	Correcto	2,5
C	5	Correcto	0,5
C	5	Correcto	1
C	3	Correcto	1,5
C	2	Correcto	2
C	1	Correcto	2,5
D	6	Correcto	0,5
D	5	Correcto	1
D	3	Correcto	1,5
D	2	Correcto	2
D	1	Correcto	2,5
E	6	Incorrecto	0,5
E	6	Incorrecto	1
E	3	Incorrecto	1,5
E	3	Correcto	2
E	2	Correcto	2,5
F	4	Correcto	0,5
F	2	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	4	Correcto	0,5
G	4	Correcto	1
G	3	Correcto	1,5
G	2	Correcto	2
G	1	Correcto	2,5
H	4	Correcto	0,5
H	3	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,199</b>			

### Tamaño para canción No. 10

Usuario	Cantidad de segmentos escuchados	Correcto / Incorrecto	Tamaño del segmento (segundos)
A	5	Correcto	0,5
A	4	Correcto	1
A	1	Correcto	1,5
A	1	Correcto	2
A	1	Correcto	2,5
B	4	Correcto	0,5
B	2	Correcto	1
B	1	Correcto	1,5
B	1	Correcto	2
B	1	Correcto	2,5
C	4	Correcto	0,5
C	3	Correcto	1
C	1	Correcto	1,5
C	1	Correcto	2
C	1	Correcto	2,5
D	3	Correcto	0,5
D	2	Correcto	1
D	1	Correcto	1,5
D	1	Correcto	2
D	1	Correcto	2,5
E	5	Correcto	0,5
E	3	Correcto	1
E	1	Correcto	1,5
E	1	Correcto	2
E	1	Correcto	2,5
F	3	Correcto	0,5
F	1	Correcto	1
F	1	Correcto	1,5
F	1	Correcto	2
F	1	Correcto	2,5
G	5	Correcto	0,5
G	3	Correcto	1
G	1	Correcto	1,5
G	1	Correcto	2
G	1	Correcto	2,5
H	3	Correcto	0,5
H	1	Correcto	1
H	1	Correcto	1,5
H	1	Correcto	2
H	1	Correcto	2,5
<b>Frecuencia promedio=1,106</b>			

**APÉNDICE 2**  
**TABLAS PARA LAS GRÁFICAS DE ENERGÍAS CONTRA**  
**COMPONENTES NOTABLES, CON ONDELTES DB12 Y**  
**TAMAÑO  $2^{15}$  MUESTRAS**



### Energía vrs. componentes notables en canción No. 1

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,00020272	9,32E-05	0,000295871	0,000109569	5
0,00045935	0,000210609	0,000669959	0,000248741	10
0,00076989	0,000343772	0,001113662	0,000426118	15
0,0011333	0,00049513	0,00162843	0,00063817	20
0,0015546	0,000667884	0,002222484	0,000886716	25
0,0020343	0,000865745	0,002900045	0,001168555	30
0,002585	0,00108991	0,00367491	0,00149509	35
0,0032163	0,001347448	0,004563748	0,001868852	40
0,0039404	0,001644987	0,005585387	0,002295413	45
0,0047803	0,001990763	0,006771063	0,002789537	50
0,0057595	0,002397394	0,008156894	0,003362106	55
0,0069124	0,002878324	0,009790724	0,004034076	60
0,0082926	0,003459205	0,011751805	0,004833395	65
0,0099792	0,004174376	0,014153576	0,005804824	70
0,012097	0,005091804	0,017188804	0,007005196	75
0,014847	0,006327241	0,021174241	0,008519759	80
0,018637	0,008113427	0,026750427	0,010523573	85
0,024442	0,011034848	0,035476848	0,013407152	90
0,035724	0,017308193	0,053032193	0,018415807	95
0,039784	0,019755201	0,059539201	0,020028799	96
0,045404	0,023256465	0,068660465	0,022147535	97
0,054161	0,028904137	0,083065137	0,025256863	98
0,07189	0,040408569	0,112298569	0,031481431	99
0,46175	0,243386293	0,705136293	0,218363707	100

## Energía vrs. componentes notables en canción No. 2

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,00024232	0,0002879	0,00053022	-4,56E-05	5
0,00060523	0,00074751	0,0013527	-0,00014228	10
0,0010874	0,0013619	0,0024492	-0,00027449	15
0,0017101	0,0021336	0,0038437	-0,00042348	20
0,0024897	0,0030826	0,0055722	-0,00059287	25
0,0034532	0,0042312	0,0076844	-0,00077805	30
0,0046331	0,0056079	0,010241	-0,00097477	35
0,0060762	0,0072521	0,013328	-0,0011759	40
0,0078446	0,0092121	0,017057	-0,0013675	45
0,010008	0,011555	0,021563	-0,0015475	50
0,012669	0,014369	0,027038	-0,0017001	55
0,01596	0,017782	0,033742	-0,0018221	60
0,020084	0,021972	0,042056	-0,0018875	65
0,025339	0,027175	0,052513	-0,001836	70
0,032232	0,033772	0,066004	-0,0015392	75
0,041673	0,042367	0,08404	-0,00069398	80
0,055609	0,053991	0,1096	0,0016179	85
0,078873	0,070884	0,14976	0,0079889	90
0,12899	0,099526	0,22852	0,029464	95
0,1479	0,10821	0,25611	0,039689	96
0,1741	0,11877	0,29288	0,05533	97
0,21411	0,1322	0,34631	0,081907	98
0,28829	0,15131	0,4396	0,13699	99
0,95852	0,18022	1,1387	0,7783	100

### Energía vrs. componentes notables en canción No. 3

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
8,42462E-05	6,79044E-05	0,000152151	1,63418E-05	5
0,000203566	0,000172302	0,000375868	3,12636E-05	10
0,000386241	0,000308775	0,000695016	7,74657E-05	15
0,000636829	0,000481953	0,001118782	0,000154876	20
0,00095705	0,00069778	0,00165483	0,000259271	25
0,001357559	0,000957903	0,002315462	0,000399655	30
0,001851082	0,00127125	0,003122333	0,000579832	35
0,002452667	0,001647164	0,004099832	0,000805503	40
0,003190587	0,002099118	0,005289705	0,001091469	45
0,004102936	0,00264054	0,006743476	0,001462396	50
0,00523733	0,00329109	0,00852842	0,00194624	55
0,006678629	0,004078713	0,010757342	0,002599916	60
0,008546023	0,005049341	0,013595364	0,003496682	65
0,011044865	0,006272221	0,017317086	0,004772644	70
0,014535566	0,007878047	0,022413614	0,006657519	75
0,019673866	0,010103842	0,029777708	0,009570024	80
0,027906698	0,013407272	0,041313969	0,014499426	85
0,042964279	0,018881179	0,061845458	0,0240831	90
0,079091611	0,029767473	0,108859084	0,049324138	95
0,093582966	0,033405508	0,126988475	0,060177458	96
0,114181338	0,03803318	0,152214518	0,076148158	97
0,146471248	0,044356041	0,190827289	0,102115206	98
0,208796842	0,054463494	0,263260337	0,154333348	99
0,824638085	0,075084872	0,899722957	0,749553212	100

### Energía vrs. componentes notables en canción No. 4

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
7,30876E-05	4,90344E-05	0,000122122	2,40532E-05	5
0,000156238	0,000119591	0,000275829	3,66465E-05	10
0,000266145	0,00021093	0,000477075	5,52145E-05	15
0,000407344	0,000322614	0,000729958	8,47292E-05	20
0,000594319	0,000463711	0,00105803	0,000130608	25
0,000843769	0,000638521	0,00148229	0,000205248	30
0,001176586	0,000864091	0,002040676	0,000312495	35
0,001614558	0,001147295	0,002761852	0,000467263	40
0,002189743	0,001506459	0,003696202	0,000683285	45
0,002935766	0,001956888	0,004892654	0,000978877	50
0,003897841	0,002515299	0,00641314	0,001382543	55
0,005148844	0,003209741	0,008358586	0,001939103	60
0,006781323	0,004073689	0,010855012	0,002707634	65
0,008956032	0,005167944	0,014123976	0,003788088	70
0,011939787	0,006589771	0,018529558	0,005350017	75
0,016217861	0,008488616	0,024706476	0,007729245	80
0,022806708	0,011186911	0,033993619	0,011619797	85
0,03435298	0,015625573	0,049978554	0,018727407	90
0,06075836	0,025494959	0,086253319	0,035263401	95
0,071055587	0,029320841	0,100376428	0,041734745	96
0,085561938	0,034640373	0,120202312	0,050921565	97
0,108298897	0,042765689	0,151064586	0,065533208	98
0,153027851	0,0577398	0,210767651	0,095288051	99
0,758474094	0,099013162	0,857487256	0,659460932	100

### Energía vrs. componentes notables en canción No. 5

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,00017257	6,99734E-05	0,000242544	0,000102597	5
0,000379654	0,000161576	0,00054123	0,000218077	10
0,000625975	0,000269154	0,000895129	0,000356822	15
0,000910441	0,000392205	0,001302646	0,000518236	20
0,001240503	0,000531926	0,001772429	0,000708577	25
0,001615977	0,000686147	0,002302125	0,00092983	30
0,002043951	0,000859449	0,0029034	0,001184502	35
0,002534044	0,001055426	0,00358947	0,001478618	40
0,003094631	0,00127795	0,004372581	0,001816682	45
0,003743311	0,001535944	0,005279255	0,002207367	50
0,004495714	0,001834096	0,00632981	0,002661619	55
0,005383624	0,002191959	0,007575583	0,003191664	60
0,006450279	0,002637044	0,009087323	0,003813234	65
0,007776175	0,0032245	0,011000675	0,004551674	70
0,009484804	0,004021009	0,013505813	0,005463794	75
0,011801389	0,005193282	0,016994671	0,006608108	80
0,015249538	0,007051796	0,022301335	0,008197742	85
0,021309457	0,01050544	0,031814897	0,010804017	90
0,036107746	0,019044291	0,055152037	0,017063456	95
0,042266305	0,02247314	0,064739446	0,019793165	96
0,051300099	0,027332785	0,078632884	0,023967315	97
0,066078061	0,034865765	0,100943826	0,031212297	98
0,096833552	0,049055013	0,145888565	0,047778538	99
0,714481804	0,120596862	0,835078665	0,593884942	100

### Energía vrs. componentes notables en canción No. 6

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,000100396	6,48358E-05	0,000165232	3,55605E-05	5
0,000235298	0,000157521	0,000392819	7,77772E-05	10
0,000417999	0,000278086	0,000696085	0,000139912	15
0,000653932	0,000430401	0,001084333	0,000223531	20
0,000948934	0,000619464	0,001568398	0,000329471	25
0,001314858	0,000850518	0,002165376	0,000464339	30
0,001766784	0,001131555	0,002898339	0,000635229	35
0,002325464	0,001469526	0,00379499	0,000855939	40
0,003017366	0,001874887	0,004892254	0,001142479	45
0,003874113	0,002354327	0,00622844	0,001519786	50
0,004946437	0,00292712	0,007873557	0,002019317	55
0,006297605	0,003614195	0,0099118	0,00268341	60
0,008028957	0,004447405	0,012476363	0,003581552	65
0,010287028	0,005473563	0,015760591	0,004813465	70
0,013310494	0,006769323	0,020079817	0,006541172	75
0,017533855	0,008457925	0,02599178	0,009075929	80
0,023823714	0,010801662	0,034625376	0,013022052	85
0,034322024	0,014439411	0,048761434	0,019882613	90
0,057201443	0,021781527	0,07898297	0,035419916	95
0,066124621	0,024539795	0,090664416	0,041584825	96
0,078919154	0,02847657	0,107395724	0,050442584	97
0,099631688	0,034840643	0,134472331	0,064791044	98
0,142486909	0,047708013	0,190194923	0,094778896	99
0,754167292	0,077965116	0,832132408	0,676202175	100

### Energía vrs. componentes notables en canción No. 7

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
6,8567E-05	5,3225E-05	0,000121792	1,53419E-05	5
0,000148382	0,000126351	0,000274733	2,20319E-05	10
0,000259107	0,000231357	0,000490464	2,77497E-05	15
0,000402108	0,0003634	0,000765509	3,87082E-05	20
0,000584529	0,000527101	0,00111163	5,74285E-05	25
0,000815955	0,000733743	0,001549698	8,22116E-05	30
0,001111349	0,000991374	0,002102723	0,000119975	35
0,001490472	0,001314233	0,002804705	0,000176239	40
0,001981683	0,001721163	0,003702846	0,00026052	45
0,002620307	0,002232061	0,004852368	0,000388247	50
0,003471076	0,00286818	0,006339256	0,000602896	55
0,004620773	0,003669753	0,008290526	0,00095102	60
0,006208868	0,004689234	0,010898102	0,001519634	65
0,008447098	0,006019684	0,014466783	0,002427414	70
0,011705551	0,007796249	0,0195018	0,003909302	75
0,016703703	0,010267564	0,026971267	0,006436139	80
0,024943436	0,013897063	0,038840498	0,011046373	85
0,040306889	0,019623993	0,059930882	0,020682897	90
0,076876032	0,030503026	0,107379058	0,046373006	95
0,09123517	0,034240122	0,125475292	0,056995048	96
0,111427327	0,039212178	0,150639505	0,072215149	97
0,142869228	0,04644551	0,189314738	0,096423717	98
0,20391241	0,059015346	0,262927757	0,144897064	99
0,962484679	0,16363264	1,126117319	0,79885204	100

### Energía vrs. componentes notables en canción No. 8

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,000115	6,2E-05	0,000177	5,31E-05	5
0,000271	0,000145	0,000417	0,000126	10
0,000485	0,000248	0,000733	0,000237	15
0,000759	0,000376	0,001135	0,000383	20
0,001097	0,000535	0,001632	0,000562	25
0,001506	0,00073	0,002235	0,000776	30
0,001995	0,000968	0,002962	0,001027	35
0,002582	0,001262	0,003844	0,00132	40
0,003285	0,00163	0,004915	0,001655	45
0,004129	0,002102	0,006231	0,002027	50
0,005158	0,002719	0,007877	0,002439	55
0,006422	0,003521	0,009943	0,002902	60
0,008012	0,004565	0,012577	0,003447	65
0,010081	0,005974	0,016055	0,004107	70
0,012875	0,008004	0,020879	0,004871	75
0,01682	0,01104	0,02786	0,00578	80
0,022856	0,015671	0,038527	0,007185	85
0,033296	0,023283	0,056579	0,010013	90
0,056517	0,038244	0,094761	0,018273	95
0,065375	0,043445	0,10882	0,02193	96
0,077732	0,050328	0,12806	0,027405	97
0,096749	0,060275	0,157024	0,036473	98
0,133013	0,077499	0,210512	0,055514	99
0,982666	0,031943	1,014609	0,950722	100

### Energía vrs. componentes notables en canción No. 9

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,000140354	7,89564E-05	0,00021931	6,13977E-05	5
0,000328958	0,000188058	0,000517017	0,0001409	10
0,000578735	0,000325358	0,000904093	0,000253376	15
0,000893533	0,000492685	0,001386218	0,000400848	20
0,001280778	0,000691514	0,001972292	0,000589264	25
0,00174871	0,000932099	0,00268081	0,000816611	30
0,002306415	0,001215059	0,003521474	0,001091355	35
0,002974913	0,001553083	0,004527996	0,00142183	40
0,003770435	0,001954926	0,005725362	0,001815509	45
0,004719913	0,002431014	0,007150927	0,002288899	50
0,005867072	0,003000257	0,008867329	0,002866815	55
0,007266888	0,003690779	0,010957667	0,00357611	60
0,009007505	0,004533448	0,013540952	0,004474057	65
0,011230315	0,005593253	0,016823568	0,005637062	70
0,014171559	0,006970282	0,02114184	0,007201277	75
0,018252396	0,008849243	0,027101639	0,009403154	80
0,02435262	0,011615435	0,035968055	0,012737185	85
0,034827994	0,016079645	0,050907638	0,018748349	90
0,059427476	0,024773945	0,084201421	0,034653531	95
0,069309093	0,027853968	0,097163061	0,041455124	96
0,083480092	0,032054606	0,115534698	0,051425486	97
0,106122689	0,038408974	0,144531663	0,067713716	98
0,151586725	0,050545061	0,202131786	0,101041664	99
0,976011509	0,09869915	1,074710659	0,877312359	100

### Energía vrs. componentes notables en canción No. 10

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
9,51E-05	0,00010072	0,00019587	-5,57E-06	5
0,00020284	0,00023491	0,00043775	-3,21E-05	10
0,00033532	0,00040484	0,00074016	-6,95E-05	15
0,00049686	0,00061137	0,0011082	-0,00011451	20
0,00069093	0,00086055	0,0015515	-0,00016962	25
0,00092936	0,0011656	0,0020949	-0,0002362	30
0,0012202	0,0015368	0,0027571	-0,00031657	35
0,0015765	0,0019927	0,0035692	-0,00041624	40
0,0020116	0,0025496	0,0045612	-0,00053804	45
0,002547	0,0032423	0,0057893	-0,00069532	50
0,0032116	0,0041178	0,0073294	-0,00090616	55
0,0040555	0,0052385	0,0092941	-0,001183	60
0,0051404	0,0066956	0,011836	-0,0015551	65
0,0065696	0,0086026	0,015172	-0,0020331	70
0,0085235	0,011146	0,019669	-0,002622	75
0,011371	0,014702	0,026073	-0,0033316	80
0,015944	0,020124	0,036068	-0,0041801	85
0,024468	0,029442	0,05391	-0,0049737	90
0,045985	0,04881	0,094794	-0,0028248	95
0,055122	0,055579	0,1107	-0,00045687	96
0,06845	0,064557	0,13301	0,0038928	97
0,090108	0,077598	0,16771	0,01251	98
0,13432	0,10047	0,23479	0,033845	99
0,94936	0,0933	1,0427	0,85606	100

### Energía vrs. componentes notables en canción No. 11

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
5,78E-05	2,34E-05	8,12E-05	3,43E-05	5
0,00010697	5,59E-05	0,00016287	5,11E-05	10
0,00016917	9,83E-05	0,00026749	7,08E-05	15
0,0002453	0,00015119	0,00039649	9,41E-05	20
0,00033685	0,00022004	0,0005569	0,00011681	25
0,00044837	0,00030028	0,00074865	0,00014808	30
0,0005887	0,00040044	0,00098915	0,00018826	35
0,00076072	0,00052415	0,0012849	0,00023657	40
0,00097962	0,00068231	0,0016619	0,00029732	45
0,0012562	0,00088557	0,0021418	0,00037061	50
0,0016152	0,0011547	0,0027699	0,00046055	55
0,0020859	0,0015183	0,0036042	0,00056754	60
0,0027249	0,0020394	0,0047643	0,00068556	65
0,0036317	0,0028121	0,0064438	0,00081957	70
0,0049937	0,0040028	0,0089964	0,00099094	75
0,0072268	0,0059412	0,013168	0,0012856	80
0,011364	0,009425	0,020789	0,001939	85
0,020426	0,016329	0,036755	0,0040969	90
0,047425	0,03177	0,079196	0,015655	95
0,059483	0,037324	0,096807	0,022159	96
0,077292	0,044777	0,12207	0,032515	97
0,10623	0,055774	0,16201	0,050457	98
0,16401	0,075532	0,23954	0,088475	99
0,98642	0,016499	1,0029	0,96992	100

## Energía vrs. componentes notables en canción No. 12

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
6,27E-05	7,16E-05	0,00013432	-8,97E-06	5
0,0001394	0,00016616	0,00030552	-2,68E-05	10
0,0002544	0,00027954	0,00053397	-2,51E-05	15
0,0004148	0,00041644	0,00083124	-1,65E-06	20
0,0006398	0,00058623	0,0012261	5,36E-05	25
0,0009456	0,00080317	0,0017488	0,00014244	30
0,0013571	0,0010889	0,002446	0,00026824	35
0,0019052	0,0014639	0,0033692	0,00044129	40
0,0026289	0,0019484	0,0045773	0,00068054	45
0,0035827	0,002569	0,0061517	0,0010138	50
0,0048293	0,003351	0,0081803	0,0014783	55
0,0064587	0,0043359	0,010795	0,0021229	60
0,0086268	0,0055852	0,014212	0,0030416	65
0,011588	0,0071998	0,018788	0,0043885	70
0,015797	0,0093281	0,025125	0,0064685	75
0,022184	0,012215	0,034399	0,0099681	80
0,03268	0,016274	0,048955	0,016406	85
0,051949	0,022464	0,074413	0,029485	90
0,095443	0,033804	0,12925	0,061639	95
0,11171	0,037567	0,14927	0,074138	96
0,13397	0,042457	0,17642	0,091509	97
0,16743	0,049273	0,2167	0,11816	98
0,22889	0,060398	0,28929	0,16849	99
0,9879	0,037484	1,0254	0,95042	100

### Energía vrs. componentes notables en canción No. 13

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
6,56E-05	7,53E-05	0,00014087	-9,71E-06	5
0,00014111	0,00020337	0,00034448	-6,23E-05	10
0,00025547	0,00041179	0,00066726	-0,00015632	15
0,00041908	0,00072698	0,0011461	-0,0003079	20
0,00065026	0,0011559	0,0018062	-0,00050567	25
0,00096748	0,0017014	0,0026688	-0,00073388	30
0,0014012	0,0023819	0,0037831	-0,00098079	35
0,0019926	0,0032212	0,0052139	-0,0012286	40
0,0027897	0,0042445	0,0070342	-0,0014548	45
0,0038634	0,005491	0,0093544	-0,0016277	50
0,0053225	0,0070122	0,012335	-0,0016897	55
0,0073393	0,0088758	0,016215	-0,0015364	60
0,010166	0,011156	0,021322	-0,00099063	65
0,014179	0,013959	0,028138	0,00022057	70
0,01993	0,017428	0,037359	0,0025021	75
0,028303	0,021795	0,050097	0,0065076	80
0,040883	0,027492	0,068375	0,013391	85
0,061265	0,03553	0,096795	0,025735	90
0,10134	0,04898	0,15032	0,052356	95
0,11539	0,053182	0,16857	0,062207	96
0,13431	0,058476	0,19279	0,075836	97
0,16262	0,065662	0,22829	0,096962	98
0,21589	0,077186	0,29308	0,13871	99
0,99196	0,002705	0,99466	0,98925	100

### Energía vrs. componentes notables en canción No. 14

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
0,00014435	9,65E-05	0,00024087	4,78E-05	5
0,00033452	0,00024019	0,00057471	9,43E-05	10
0,00058425	0,00042778	0,001012	0,00015647	15
0,00089718	0,00066198	0,0015592	0,0002352	20
0,0012845	0,00095049	0,002235	0,000334	25
0,0017561	0,0012979	0,003054	0,00045818	30
0,0023238	0,0017071	0,0040308	0,00061666	35
0,0030074	0,0021895	0,0051969	0,00081788	40
0,0038274	0,002763	0,0065903	0,0010644	45
0,0048164	0,0034418	0,0082582	0,0013746	50
0,0060181	0,0042514	0,01027	0,0017667	55
0,0074979	0,0052256	0,012724	0,0022723	60
0,0093454	0,0064178	0,015763	0,0029276	65
0,011689	0,0078973	0,019586	0,0037919	70
0,014752	0,009782	0,024534	0,0049701	75
0,018917	0,012268	0,031185	0,0066486	80
0,024939	0,015702	0,040641	0,0092374	85
0,034576	0,020854	0,05543	0,013722	90
0,053864	0,03006	0,083924	0,023804	95
0,060763	0,033082	0,093844	0,027681	96
0,070114	0,037012	0,10713	0,033103	97
0,084127	0,0426	0,12673	0,041526	98
0,11	0,052348	0,16235	0,057654	99
0,95497	0,032084	0,98705	0,92288	100

### Energía vrs. componentes notables en canción No. 15

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
6,81E-05	5,50E-05	0,00012311	1,32E-05	5
0,00013659	0,00013338	0,00026996	3,21E-06	10
0,00022251	0,00022746	0,00044997	-4,94E-06	15
0,00032463	0,00034143	0,00066607	-1,68E-05	20
0,00044463	0,00047445	0,00091908	-2,98E-05	25
0,00058922	0,00063066	0,0012199	-4,14E-05	30
0,00076017	0,00081221	0,0015724	-5,20E-05	35
0,00096955	0,0010242	0,0019937	-5,46E-05	40
0,0012289	0,0012759	0,0025048	-4,70E-05	45
0,0015554	0,0015717	0,0031271	-1,63E-05	50
0,0019699	0,001923	0,0038929	4,69E-05	55
0,0025078	0,002348	0,0048558	0,00015975	60
0,003218	0,0028654	0,0060833	0,00035262	65
0,0041957	0,0035138	0,0077096	0,00068189	70
0,005622	0,0043765	0,0099985	0,0012456	75
0,0078639	0,0056367	0,013501	0,0022272	80
0,011718	0,007715	0,019433	0,0040034	85
0,019286	0,011987	0,031273	0,0072995	90
0,038954	0,024107	0,063062	0,014847	95
0,047293	0,029216	0,076509	0,018076	96
0,059628	0,036532	0,09616	0,023097	97
0,080356	0,047877	0,12823	0,032479	98
0,12539	0,069411	0,1948	0,055981	99
0,98298	0,022182	1,0052	0,9608	100

### Energía vrs. componentes notables en canción No. 16

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
8,97E-05	7,68E-05	0,00016652	1,29E-05	5
0,00019103	0,00019485	0,00038588	-3,81E-06	10
0,00032483	0,00035661	0,00068145	-3,18E-05	15
0,00049447	0,00056698	0,0010614	-7,25E-05	20
0,00070516	0,00083617	0,0015413	-0,00013101	25
0,00097059	0,0011706	0,0021412	-0,00020002	30
0,0013012	0,0015822	0,0028833	-0,00028099	35
0,0017207	0,0020879	0,0038086	-0,00036721	40
0,00226	0,0027003	0,0049603	-0,00044023	45
0,0029632	0,0034473	0,0064105	-0,00048405	50
0,0039063	0,0043531	0,0082593	-0,00044681	55
0,0052086	0,0054548	0,010663	-0,00024615	60
0,0070496	0,0067859	0,013835	0,00026369	65
0,0097412	0,008389	0,01813	0,0013522	70
0,013795	0,01032	0,024115	0,0034748	75
0,020117	0,012661	0,032778	0,0074558	80
0,030582	0,015527	0,046109	0,015055	85
0,049706	0,01907	0,068776	0,030636	90
0,092974	0,024245	0,11722	0,068729	95
0,10931	0,025904	0,13521	0,083403	96
0,13184	0,028179	0,16002	0,10366	97
0,16622	0,031695	0,19792	0,13453	98
0,23149	0,038631	0,27012	0,19286	99
0,9935	0,0027605	0,99626	0,99074	100

### Energía vrs. componentes notables en canción No. 17

Promedio	Desv. Est.	Prom.+D. E.	Prom.-D.E.	Energía
8,20E-05	7,80E-05	0,00016002	3,96E-06	5
0,0001707	0,00018248	0,00035318	-1,18E-05	10
0,0002868	0,00031809	0,00060494	-3,13E-05	15
0,0004327	0,00048356	0,00091627	-5,09E-05	20
0,0006158	0,00068218	0,001298	-6,64E-05	25
0,0008555	0,00092025	0,0017758	-6,47E-05	30
0,0011654	0,0012015	0,0023669	-3,62E-05	35
0,001577	0,0015417	0,0031187	3,53E-05	40
0,0021213	0,0019602	0,0040815	0,00016116	45
0,0028413	0,0024785	0,0053198	0,0003628	50
0,0037916	0,0031232	0,0069147	0,00066843	55
0,0050657	0,0039233	0,008989	0,0011424	60
0,0068109	0,0049123	0,011723	0,0018985	65
0,0092736	0,0061572	0,015431	0,0031164	70
0,012893	0,0077475	0,02064	0,0051452	75
0,018465	0,0098829	0,028348	0,0085824	80
0,027584	0,013004	0,040588	0,014581	85
0,044148	0,018514	0,062662	0,025635	90
0,082677	0,032306	0,11498	0,050371	95
0,097717	0,037993	0,13571	0,059724	96
0,11879	0,046082	0,16487	0,072705	97
0,15136	0,058625	0,20998	0,092732	98
0,21343	0,081782	0,29522	0,13165	99
0,9893	0,011545	1,0008	0,97776	100



**APÉNDICE 3**  
**TABLAS PARA GRÁFICA COMPONENTES NOTABLES CONTRA**  
**FAMILIAS ONDELETES CON SEGMENTOS DE TAMAÑO  $2^{15}$**   
**MUESTRAS**



### Componentes notables vrs. familia ondelete en canción No. 1

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0874	2	-0,0038
0,0836	3	-0,0016
0,082	4	-0,0008
0,0811	5	-0,0005
0,0806	6	-0,0003
0,0803	7	-0,0002
0,0801	8	-0,0002
0,0799	9	-0,0002
0,0797	10	-0,0001
0,0796	11	-0,0001
0,0796	12	-0,0001
0,0795	13	-0,0001
0,0794	14	-0,0001
0,0794	15	0
0,0794	16	0
0,0793	17	0
0,0793	18	0
0,0793	19	0
0,0792	20	0
0,0792	21	

## Componentes notables vrs. familia ondelete en canción No. 2

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0712	2	-0,0026
0,0686	3	-0,0011
0,0675	4	-0,0005
0,0671	5	-0,0003
0,0668	6	-0,0002
0,0666	7	-0,0001
0,0664	8	-0,0001
0,0664	9	-0,0001
0,0663	10	0
0,0662	11	0
0,0662	12	0
0,0662	13	0
0,0662	14	0
0,0662	15	0
0,0662	16	0
0,0661	17	0
0,0662	18	0
0,0661	19	0
0,0662	20	0
0,0662	21	

### Componentes notables vrs. familia ondelete en canción No. 3

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0954	2	-0,0041
0,0913	3	-0,0018
0,0895	4	-0,001
0,0885	5	-0,0006
0,0878	6	-0,0003
0,0875	7	-0,0003
0,0872	8	-0,0002
0,087	9	-0,0002
0,0868	10	-0,0001
0,0867	11	-0,0001
0,0866	12	-0,0001
0,0865	13	0
0,0864	14	0
0,0864	15	0
0,0863	16	-0,0001
0,0863	17	0
0,0863	18	0
0,0862	19	-0,0001
0,0862	20	0
0,0861	21	

### Componentes notables vrs. familia ondelete en canción No. 4

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0788	2	-0,0029
0,0759	3	-0,0014
0,0744	4	-0,0008
0,0736	5	-0,0006
0,073	6	-0,0004
0,0726	7	-0,0003
0,0722	8	-0,0003
0,0719	9	-0,0002
0,0717	10	-0,0002
0,0716	11	-0,0001
0,0714	12	-0,0001
0,0713	13	-0,0001
0,0712	14	-0,0001
0,0711	15	-0,0001
0,071	16	-0,0001
0,0709	17	-0,0001
0,0709	18	0
0,0708	19	-0,0001
0,0708	20	0
0,0707	21	

### Componentes notables vrs. familia ondelete en canción No. 5

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,081	2	-0,003
0,078	3	-0,0013
0,0767	4	-0,0007
0,0759	5	-0,0005
0,0754	6	-0,001
0,0744	7	-0,0002
0,0742	8	-0,0002
0,074	9	-0,0001
0,0739	10	-0,0001
0,0738	11	-0,0001
0,0737	12	-0,0001
0,0736	13	-0,0001
0,0735	14	0
0,0735	15	0
0,0735	16	0
0,0734	17	0
0,0734	18	0
0,0734	19	0
0,0734	20	0
0,0733	21	

### Componentes notables vrs. familia ondelete en canción No. 6

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0727	2	-0,0031
0,0695	3	-0,0013
0,0682	4	-0,0007
0,0675	5	-0,0005
0,067	6	-0,0003
0,0667	7	-0,0002
0,0664	8	-0,0002
0,0662	9	-0,0002
0,0661	10	-0,0001
0,0659	11	-0,0001
0,0658	12	-0,0001
0,0658	13	-0,0001
0,0657	14	-0,0001
0,0656	15	0
0,0656	16	0
0,0655	17	0
0,0655	18	0
0,0655	19	0
0,0654	20	0
0,0654	21	

### Componentes notables vrs. familia ondelete en canción No. 7

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,102	2	-0,0037
0,0983	3	-0,0018
0,0965	4	-0,0011
0,0954	5	-0,0006
0,0947	6	-0,0005
0,0942	7	-0,0004
0,0939	8	-0,0003
0,0936	9	-0,0002
0,0934	10	-0,0002
0,0932	11	-0,0002
0,093	12	-0,0001
0,0929	13	-0,0001
0,0928	14	-0,0001
0,0928	15	-0,0001
0,0926	16	-0,0001
0,0926	17	-0,0001
0,0925	18	0
0,0925	19	-0,0001
0,0924	20	0
0,0924	21	

### Componentes notables vrs. familia ondelete en canción No. 8

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0727	2	-0,0023
0,0704	3	-0,001
0,0694	4	-0,0006
0,0689	5	-0,0003
0,0685	6	-0,0002
0,0683	7	-0,0002
0,0681	8	-0,0001
0,068	9	-0,0001
0,0679	10	-0,0001
0,0678	11	-0,0001
0,0677	12	-0,0001
0,0677	13	-0,0001
0,0676	14	0
0,0676	15	0
0,0675	16	0
0,0675	17	0
0,0675	18	0
0,0675	19	0
0,0675	20	0
0,0674	21	

### Componentes notables vrs. familia ondelete en canción No. 9

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0726	2	-0,005
0,0676	3	-0,0008
0,0668	4	-0,0003
0,0666	5	-0,0002
0,0664	6	0
0,0664	7	0
0,0663	8	0
0,0664	9	0
0,0664	10	0,0001
0,0665	11	0
0,0665	12	0,0001
0,0666	13	0
0,0666	14	0,0001
0,0666	15	0,0001
0,0667	16	0
0,0667	17	0,0001
0,0668	18	0
0,0668	19	0
0,0669	20	0
0,0669	21	

### Componentes notables vrs. familia ondelete en canción No. 10

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0696	2	-0,0025
0,0671	3	-0,0011
0,066	4	-0,0006
0,0655	5	-0,0004
0,0651	6	-0,0002
0,0649	7	-0,0002
0,0647	8	-0,0001
0,0646	9	-0,0001
0,0645	10	-0,0001
0,0644	11	-0,0001
0,0643	12	0
0,0643	13	-0,0001
0,0642	14	-0,0001
0,0642	15	0
0,0641	16	0
0,0641	17	0
0,0641	18	0
0,0641	19	0
0,0641	20	0
0,064	21	

### Componentes notables vrs. familia ondelete en canción No. 11

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0644	2	-0,0021
0,0623	3	-0,0009
0,0614	4	-0,0005
0,0609	5	-0,0003
0,0606	6	-0,0002
0,0605	7	-0,0001
0,0603	8	-0,0001
0,0602	9	-0,0001
0,0601	10	-0,0001
0,0601	11	0
0,06	12	0
0,06	13	0
0,0599	14	0
0,0599	15	0
0,0599	16	0
0,0599	17	0
0,0599	18	0
0,0599	19	0
0,0598	20	0
0,0598	21	0

### Componentes notables vrs. familia ondelete en canción No. 12

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0666	2	-0,0018
0,0648	3	-0,0009
0,0639	4	-0,0005
0,0634	5	-0,0003
0,0632	6	-0,0002
0,063	7	-0,0001
0,0628	8	-0,0001
0,0627	9	-0,0001
0,0626	10	-0,0001
0,0625	11	0
0,0625	12	0
0,0625	13	0
0,0624	14	0
0,0624	15	0
0,0624	16	0
0,0624	17	0
0,0623	18	0
0,0623	19	0
0,0623	20	0
0,0623	21	0

### Componentes notables vrs. familia ondelete en canción No. 13

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0632	2	-0,0023
0,061	3	-0,001
0,06	4	-0,0003
0,0597	5	-0,0002
0,0595	6	-0,0002
0,0593	7	0,0001
0,0594	8	0,0001
0,0595	9	-0,0001
0,0594	10	0,0001
0,0595	11	0,0002
0,0597	12	-0,0001
0,0596	13	0,0001
0,0597	14	0,0002
0,0599	15	0
0,0598	16	0
0,0599	17	0,0002
0,06	18	0
0,06	19	0
0,0601	20	0,0002
0,0602	21	

### Componentes notables vrs. familia ondelete en canción No. 14

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0865	2	-0,0035
0,0831	3	-0,0015
0,0815	4	-0,0009
0,0807	5	-0,0005
0,0801	6	-0,0004
0,0797	7	-0,0003
0,0795	8	-0,0002
0,0792	9	-0,0002
0,0791	10	-0,0001
0,0789	11	-0,0001
0,0788	12	-0,0001
0,0787	13	-0,0001
0,0786	14	-0,0001
0,0785	15	0
0,0785	16	0
0,0785	17	-0,0001
0,0784	18	0
0,0784	19	0
0,0783	20	0
0,0783	21	

### Componentes notables vrs. familia ondelete en canción No. 15

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0906	2	-0,0048
0,0858	3	-0,0021
0,0837	4	-0,0012
0,0824	5	-0,0007
0,0817	6	-0,0005
0,0812	7	-0,0003
0,0809	8	-0,0003
0,0806	9	-0,0002
0,0804	10	-0,0002
0,0802	11	-0,0001
0,0801	12	-0,0001
0,0799	13	-0,0001
0,0798	14	-0,0001
0,0797	15	-0,0001
0,0796	16	-0,0001
0,0796	17	-0,0001
0,0795	18	0
0,0795	19	-0,0001
0,0794	20	0
0,0794	21	

### Componentes notables vrs. familia ondelete en canción No. 16

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0694	2	-0,0032
0,0662	3	-0,0015
0,0647	4	-0,0009
0,0637	5	-0,0005
0,0632	6	-0,0005
0,0628	7	-0,0003
0,0624	8	-0,0002
0,0622	9	-0,0002
0,062	10	-0,0002
0,0618	11	-0,0002
0,0617	12	-0,0001
0,0616	13	-0,0001
0,0615	14	-0,0001
0,0614	15	-0,0001
0,0613	16	0
0,0612	17	-0,0001
0,0612	18	-0,0001
0,0611	19	-0,0001
0,0611	20	0
0,061	21	

### Componentes notables vrs. familia ondelete en canción No. 17

Comp. Not.	Familia	Razón de cambio de los Comp. Not.
0,0623	2	-0,0019
0,0604	3	-0,0008
0,0596	4	-0,0005
0,0592	5	-0,0003
0,0589	6	-0,0002
0,0587	7	-0,0001
0,0586	8	-0,0001
0,0584	9	-0,0001
0,0584	10	-0,0001
0,0583	11	-0,0001
0,0582	12	-0,0001
0,0582	13	0
0,0581	14	0
0,0581	15	0
0,0581	16	0
0,0581	17	0
0,058	18	0
0,058	19	0
0,058	20	0
0,058	21	0

### Componentes notables vrs. familia ondelete en canción No. 18

Comp. Not.	Familia	Valor de derivada
0,0806	2	-0,0028
0,0778	3	-0,0013
0,0766	4	-0,0006
0,0759	5	-0,0004
0,0755	6	-0,0002
0,0753	7	-0,0002
0,0751	8	-0,0002
0,0749	9	0
0,0748	10	-0,0001
0,0747	11	-0,0001
0,0746	12	0
0,0746	13	0
0,0746	14	-0,0001
0,0745	15	0
0,0745	16	0
0,0745	17	-0,0001
0,0744	18	0,0001
0,0745	19	0
0,0745	20	-0,0001
0,0744	21	

**APÉNDICE 4**  
**PROGRAMAS DE MATLAB**



## Programa de recomposición

```
function y=composicionppunto(filename,familia,tam,ubic)
%Programa creado por Kevin Francisco Hernández Ramírez
%Recomposición de piezas musicales con sonidos de la naturaleza
%Si no se pone una ubicación se coloca una por default
if nargin<4
    ubic='C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Primeras pruebas\';
end
%Se escoge las tablas de la base de datos según el tamaño de
%la recomposición.
if tam==15
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\15\canciones.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\15\segmentos.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\15\cn2.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\15\indices1.mat');
    cn=400;
elseif tam==14
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\14\canciones.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\14\segmentos.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\14\cn1.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\14\indices1.mat');
    data1a=data1;
    cn=200;
elseif tam==13
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\13\canciones.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\13\segmentos.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\13\cn1.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\13\indices1.mat');
    data1a=data1;
    cn=150;
```

```

elseif tam==12
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\12\canciones.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\12\segmentos.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\12\cn3.mat');
    load('C:\Documents and Settings\Kevin\Mis documentos\Kevin\USAC\
    Electronica\Tesis\Tesis wavelet\Bases de Datos\12\indices3.mat');
    data1a=data1;
    cn=300;
end
%Se lee el archivo de audio original
nombre=sprintf('%s',ubic,filename);
yx=wavread(nombre);
l=length(yx);
yx1=zeros(l,1);
dir='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\
Tesis\Tesis wavelet\Bases de Datos\';
dir2='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\
Tesis\Tesis wavelet\Resultados\';
diag='\';
ext='.wav';
%Se verifica si la pieza es monofónica o estereofónica, si es estereofónica
%se convierte a monofónica.
if size(yx,2)==2
    for s=1:1:l
        yx1(s,1)=sum(yx(s,:))/2;
    end
else
    yx1=yx;
end
r=2^tam;
x=floor(l/r)
y=zeros(r*x,1);
pra=zeros(1,cn);
%Se separa la señal en segmentos del mismo tamaño para caracterizarla
for co=1:1:x
    ym1=zeros(r,1);
    for t=1:1:r
        ym1(t,1)=yx1(r*(co-1)+t,1);
    end
end

```

```

wavwrite(ym1,44100,'segmento.wav');
%Esta función sirve para escoger los componentes notables
[componentes indices energ]=compnot1('segmento.wav', tam, familia, cn);
if indices==pra
    if tam==15
        yo=wavread('C:\Documents and Settings\Kevin\Mis documentos\
Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\
cero.au\1.wav');
        norm=1;
    elseif tam==14
        yo=wavread('C:\Documents and Settings\Kevin\Mis documentos\
Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\
cero.au\2.wav');
        norm=1;
    elseif tam==13
        yo=wavread('C:\Documents and Settings\Kevin\Mis documentos\
Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\
cero.au\3.wav');
        norm=1;
    elseif tam==12
        yo=wavread('C:\Documents and Settings\Kevin\Mis documentos\
Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\
cero.au\4.wav');
        norm=1;
    end
else
%Se escogen los 10 segmentos que van a recomponer la canción con sus
%productos puntos.

[idsegme1,comp1]=elecidseg4(idseg,data2,data1a,tam,componentes,indices);
[idsegme2,comp2]=elecidseg4(idseg,data2,data1a,tam,comp1);
[idsegme3,comp3]=elecidseg4(idseg,data2,data1a,tam,comp2);
[idsegme4,comp4]=elecidseg4(idseg,data2,data1a,tam,comp3);
[idsegme5,comp5]=elecidseg4(idseg,data2,data1a,tam,comp4);
[idsegme6,comp6]=elecidseg4(idseg,data2,data1a,tam,comp5);
[idsegme7,comp7]=elecidseg4(idseg,data2,data1a,tam,comp6);
[idsegme8,comp8]=elecidseg4(idseg,data2,data1a,tam,comp7);
[idsegme9,comp9]=elecidseg4(idseg,data2,data1a,tam,comp8);
[idsegme10,comp10]=elecidseg4(idseg,data2,data1a,tam,comp9);

idsegme=[idsegme1;idsegme2;idsegme3;idsegme4;idsegme5;idsegme6;idsegme7;
idsegme8;idsegme9;idsegme10];
idsegme=sortrows(idsegme);

```

```

%Se leen los segmentos y se multiplican por el valor necesario para
%recomponer la canción, y se suman todos los segmentos para la
%recomposición por segmento
[energia segmento cancion]=elecbs(idsegme,datos,u);
yo=zeros(r,1);
beta=0;
for si=1:1:size(idsegme,1)
    mul=idsegme(si,2)*sqrt(energ/energia(si,1));
    can=cell2str2(cancion(si,1));
    ui=num2str(segmento(si,1));
    um=sprintf('%s',ui,ext);
    dire=sprintf('%s',dir,can,diag,um);
    yu=wavread(dire);
    yo=yo+(mul.*yu);
end
%Se normalizan los segmentos
pot=sum(yo.^2);
norm=energ/pot;
end
yi=(sqrt(norm)).*yo;
size(yi)
%Se llena el vector con todos los segmentos
y(r*(co-1)+1:r*co,1)=yi;
co
end
%Se convierte el vector en una señal de audio
nom=sprintf('%s',dir2,filename);
wavwrite(y,44100,nom);
end

```

## Programa de búsqueda y reconocimiento

```

function [idsegme
compo]=elecidsseg4(idseg,data2,cos,tam,componentes,indices)
%Programa creado por Kevin Francisco Hernández Ramírez
%Búsqueda y reconocimiento
%Se toma en cuenta si los vectores de entrada son comprimidos o expandidos
%si son comprimidos los expande de lo contrario los deja como están
if nargin<6
    com=componentes;
else
    pro=zeros(1,2^tam);

```

```

    for m=1:1:size(indices,2)
        pos=indices(1,m)+1;
        pro(1,pos)=componentes(1,m);
    end
    com=pro;
end
mi=size(data2,1);
can=zeros(mi,1);
%Se recorren todos los componentes de los vectores almacenados para
%realizar el producto punto entre la señal y todos los vectores de la base
%de datos
for s=1:1:mi
    pra=data2(s,:);
    pre=cos(s,:);
    con=zeros(1,size(data2,2));
    for si=1:1:size(data2,2)
        ub=pra(1,si)+1;
        mul=pre(1,si);
        mul1=com(1,ub);
        con(1,si)=mul*mul1;
    end
    can1=sum(con);
    can(s,1)=can1;
end
%Se escoge el segmento que está relacionado con el segmento original.
[c i]=max(can);
%Se almacenan los datos del ID del segmento y el producto punto
idsegme=[idseg(i,1) can(i,1)];
%Se multiplica el vector escogido por el producto punto y se le resta este
%vector al original para que vuelva a hacerse los mismo.
new=idsegme(1,2)*cos(i,:);
ind=data2(i,:);
compo1=zeros(1,2^tam);
for so=1:1:size(data2,2)
    ub1=ind(1,so)+1;
    compo1(1,ub1)=new(1,so);
end
compo=com-compo1;
end

```

### **Programa de creación de los sonidos originales de la base de datos**

```

function [y l]=segmentacionbase(filename,tam,elec,ubic)
%Programa de creación de los sonidos originales de la base de datos

```

```

%Creado por Kevin Hernandez
if nargin<4
    ubic='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Primeras pruebas\';
end
esp=' ';
tam1=num2str(tam);
filename2=sprintf('%s',filename,esp,tam1);
mkdir('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos',filename2);
nombre=sprintf('%s',ubic,filename);
if elec==1
[y fs]=auread(nombre);
elseif elec==2
    [y fs]=wavread(nombre);
end
l=length(y);
yx=zeros(l,1);
if size(y,2)==2
    for m=1:1:l
        yx(m,1)=sum(y(m,:))/2;
    end
else
    yx=y;
end
if isstr(tam)==1
    tam=str2num(tam);
end
r=2^tam;
x=floor(l/r);
dia='\';
for t=1:1:x
    y1=zeros(r,1);
    for ti=1:1:r
        y1(ti,1)=yx(r*(t-1)+ti,1);
    end
    mu=num2str(t);
    tu='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\';
    dir=sprintf('%s',tu,filename2,dia);
    nom=sprintf('%s.wav',mu);
    fin=sprintf('%s',dir,nom);
    wavwrite(y1,fs,fin);

```

```
end  
end
```

## Programas de base de datos

### 1. Programa creación de tabla canciones

```
function tablacanciones(filename)  
%Programa creado por Kevin Francisco Hernández Ramírez  
%Creación de tabla Canciones.mat  
%Se carga la tabla donde se encuentra el ID de la canción y su nombre.  
dir='C:\Documents and Settings\Kevin\Mis  
documentos\Kevin\USAC\Electronica\  
Tesis\Tesis wavelet\Bases de Datos\12\canciones.mat';  
load(dir);  
m={filename};  
t=size(u,1);  
s=cell2num(u(t,1))+1;  
u=[u;s m];  
%Se guarda el ID de esta canción que es el ID de la última canción más uno  
%y el nombre de la canción en la tabla Canciones.mat  
save('C:\Documents and Settings\Kevin\Mis  
documentos\Kevin\USAC\Electronica\  
Tesis\Tesis wavelet\Bases de Datos\12\canciones.mat','u');  
end
```

### 2. Programa creación de tabla segmentos

```
function tablasegmentos(filename,seg,num)  
%Programa creado por Kevin Francisco Hernández Ramírez  
%Creación de tabla Segmentos.mat  
%Se carga la tabla donde están almacenados el ID de segmento, el ID de la  
%canción a la que pertenece el segmento, la energía y el número del  
%segmento  
dia='\  
dir='C:\Documents and Settings\Kevin\Mis  
documentos\Kevin\USAC\Electronica\  
Tesis\Tesis wavelet\Bases de Datos\  
ext='.wav';  
dir1=sprintf('%s',dir,filename,dia);  
load('C:\Documents and Settings\Kevin\Mis  
documentos\Kevin\USAC\Electronica\  
Tesis\Tesis wavelet\Bases de Datos\12\segmentos.mat');
```

```

id=num;
sipi1=zeros(seg,4);
ru=datos;
sip=size(datos,1);
dat=datos(sip,1);
for m=1:1:seg
    m1=num2str(m);
    dire=sprintf('%s',dir1,m1,ext);
    y=wavread(dire);
    energ=sum(y.^2);
    sipi=[dat+m id energ m];
    sipi1(m,:)=sipi;
end
%Se utiliza una variable en donde esta almacenada la matriz de todos los
%datos que estaban guardados y se agrega los nuevos datos.
datos=[ru;sipi1];
%Se guarda la matriz completa en la tabla Segmentos.mat
save('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos\12\segmentos.mat','datos');
end

```

### 3. Programa de creación de tablas de componentes notables y de sus índices

```

function tablasindcn(filename,seg,cn,familia,tam,in)
%Programa creado por Kevin Francisco Hernández Ramírez
%Creación de las tablas de Componentes Notables y sus Índices
%Se cargan las dos tablas: en la primera aparece el ID del segmento al
%que pertenecen con sus componentes notables y en la segunda el ID del
%segmento al que pertenece y los índices de los componentes notables y se
%almacenan en una matriz
if nargin<6
    in=1;
end
pru={filename};
dia='\';
dir='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de Datos\';
ext='.wav';
dir1=sprintf('%s',dir,filename,dia);

```

```

load('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos\12\cn3.mat');
load('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos\12\indices3.mat');
pra=data1;
pre=data2;
pit=idseg;
pes=size(idseg,1)-1;
pri=zeros(seg-in,1);
pro=zeros(seg-in,cn);
pru=zeros(seg-in,cn);
for m=in:1:seg
    m1=num2str(m);
    dire=sprintf('%s',m1,ext);
    [componentes indices]=compnot1(dire, tam, familia, cn, dir1);
    pro(m,:)=componentes;
    pru(m,:)=indices;
    pri(m,:)=pes+m;
end
%Se almacenan todos los datos en una matriz y luego se guardan en las
%tablas de Componentes Notables e Índices.
data1=[pra;pro];
data2=[pre;pru];
idseg=[pit;pri];
save('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos\12\cn3.mat','idseg','data1');
save('C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Bases de
Datos\12\indices3.mat','idseg','data2');
end

```

## Programas de ejercicios

### 1. Programa de creación de gráficas de componentes notables vrs. familias Daubechies

```

function [deri u prome enc]=cnvrsfamilia(nombre, tam)
%Programa de graficas de Componentes notables Vrs. Familia Daubechies
%Creado por Kevin Hernandez

```

```

tama=num2str(tam);
vec=[];
for t=2:1:21
    r=sprintf('db%01d',t);
    name=[nombre ' ' r ' ' tama ' promedio.csv'];
    num=ploteo(name,2);
    if length(num)>24
        nume=zeros(24,1);
        n=1;
        for m=1:1:24
            if (m==20 || m==21 || m==22 || m==23)
                n=n+4;
            end
            nume(m,1)=num(n,1);
            n=n+1;
        end
    else
        nume=num;
    end

    vec=[vec nume];
end
enc=[2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21];
prome=zeros(20,1);
for x=1:1:20
    prome(x,1)=mean(vec(:,x));
end
u=[enc prome];
deri=diff(prome)';
plot(enc,prome);
end

```

## 2. Programa de gráficas energía vrs. componentes notables

```

function varargout = Graficas(varargin)
% Programa de interface GUI para graficas de Energia Vrs. Componentes
% Notables.
%Creado por Kevin Hernandez
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Graficas_OpeningFcn, ...

```

```

        'gui_OutputFcn', @Graficas_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Graficas is made visible.
function Graficas_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Graficas (see VARARGIN)
% Choose default command line output for Graficas
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes Graficas wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = Graficas_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1
% --- Executes during object creation, after setting all properties.
c=get(hObject,'Value');

```

```
switch c
  case 1
    m='db2';
  case 2
    m='db3';
  case 3
    m='db4';
  case 4
    m='db5';
  case 5
    m='db6';
  case 6
    m='db7';
  case 7
    m='db8';
  case 8
    m='db9';
  case 9
    m='db10';
  case 10
    m='db11';
  case 11
    m='db12';
  case 12
    m='db13';
  case 13
    m='db14';
  case 14
    m='db15';
  case 15
    m='db16';
  case 16
    m='db17';
  case 17
    m='db18';
  case 18
    m='db19';
  case 19
    m='db20';
  case 20
    m='db21';
end
handles.popupmenu1=m;
guidata(hObject,handles);
```

```

function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
%    contents{get(hObject,'Value')} returns selected item from popupmenu2
% --- Executes during object creation, after setting all properties
b=get(hObject,'Value');
switch b
    case 1
        m='15';
    case 2
        m='16';
end
handles.popupmenu2=m;
guidata(hObject,handles);
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'Background', 'white');
end
% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell
array
%     contents{get(hObject,'Value')} returns selected item from popupmenu3
% --- Executes during object creation, after setting all properties.
a=get(hObject,'Value');
switch a
    case 1
        m='eagles - hotel california.wav';
    case 2
        m='apuesta d rock and roll.wav';
    case 3
        m='chichi peralta - amor narcotico.wav';
    case 4
        m='fran sinatra - new york new york.wav';
    case 5
        m='sonata arctica - blank file(2).wav';
    case 6
        m='a la la la la long.wav';
    case 7
        m='09 mas alla del sol.wav';
    case 8
        m='desde mi cielo.wav';
    case 9
        m='a ti.wav';
    case 10
        m='tu amor me hace bien.wav';
    case 11
        m='goodbye my love.wav';
    case 12
        m='Cosa Mas Linda Que Tu.wav';
    case 13
        m='infinity.wav';
    case 14
        m='horoscopos ft gloria trevi - cinco minutos (duranguense).wav';
    case 15
        m='The Beatles - Yesterday.wav';
    case 16
        m='salsa cubana la charanga habanera - la charanga habanera - lola
lola.wav';
    case 17
        m='01-satisfaction45.wav';
    case 18
        m='dj tiesto & maxi jazz - dance4life(2).wav';
    case 19

```

```

        m='04 brick by boring brick.wav';
    case 20
        m='metallica - one.wav';
    case 21
        m='carlos gardel - tango (perfume de mujer soundtrack).wav';
    case 22
        m='edward maya feat alicia - stereo love (original mix).wav';
    case 23
        m='number one feat n dubz.wav';
    case 24
        m='kesha - tik tok469.wav';
end
handles.popupmenu3=m;
guidata(hObject,handles);
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a3=handles.popupmenu1;
a2=handles.popupmenu2;
a1=handles.popupmenu3;
name=[a1, ' ', a3, ' ', a2, ' promedio.csv'];
set(handles.textosalida,'String',name);
%a3=cell2str(a3)
%a4=get(hObject,'String');
%a4=cell2str(a4)
%a5=get(hObject,'String');
%a5=cell2str(a5)
%a6=[a3, ' ', a5, ' ', a4, ' ', 'promedio'];
%name=sprintf('%s.csv',a6);
fid=fopen(name);
A=[];
while true

```

```

m=fgetl(fid);
if m==-1
    break
else
    a=1;
    c=[];
    i=1;
    while i<6
        b=[];
        while true
            if a>size(m,2)
                break
            else
                tr=strcmp(m(1,a),',');
            end
            if tr==1
                break
            else
                b=[b m(1,a)];
            end
            a=a+1;
        end
        out=str2double(b);
        c=[c out];
        i=i+1;
        a=a+1;
    end
    A=[A;c];
end
end
x=A(:,5);
y=A(:,1);
y1=A(:,3);
y2=A(:,4);
plot(y,x,'b',y1,x,'r',y2,x,'g');
q1=handles.edit1;
q2=handles.edit2;
q3=handles.edit3;
q4=handles.edit4;
axis([q1 q2 q3 q4]);
fclose(fid);
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%    str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes during object creation, after setting all properties.
d=get(hObject,'String');
MinX=str2double(d);
handles.edit1=MinX;
guidata(hObject,handles);
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit2 as text
%    str2double(get(hObject,'String')) returns contents of edit2 as a double
% --- Executes during object creation, after setting all properties.
e=get(hObject,'String');
MaxX=str2double(e);
handles.edit2=MaxX;
guidata(hObject,handles);
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit3 as text

```

```

%    str2double(get(hObject,'String')) returns contents of edit3 as a double
% --- Executes during object creation, after setting all properties.
f=get(hObject,'String');
MinY=str2double(f);
handles.edit3=MinY;
guidata(hObject,handles);
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit4 as text
%    str2double(get(hObject,'String')) returns contents of edit4 as a double
% --- Executes during object creation, after setting all properties.
g=get(hObject,'String');
MaxY=str2double(g);
handles.edit4=MaxY;
guidata(hObject,handles);
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

### 3. Programa de elección de tamaño de segmentos

```

function varargout = Segmentos(varargin)
%Programa para elección de tamaño de segmento por tiempos.
%Creado por Kevin Hernández
gui_Singleton = 1;

```

```

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Segmentos_OpeningFcn, ...
                  'gui_OutputFcn', @Segmentos_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before Segmentos is made visible.
function Segmentos_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Segmentos (see VARARGIN)
% Choose default command line output for Segmentos
set(handles.Salida,'String','');
set(handles.Segmento,'String','');
set(handles.tama,'string','');
numseg=(1:1:600);
a=0;
handles.numseg=numseg;
handles.output = hObject;
handles.a=a;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes Segmentos wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = Segmentos_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes on button press in uno.

```

```

function uno_Callback(hObject, eventdata, handles)
% hObject   handle to uno (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
set(handles.Segmentos,'UserData',1);
guidata(hObject, handles);
% --- Executes on button press in cuatro.
function cuatro_Callback(hObject, eventdata, handles)
% hObject   handle to cuatro (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
numcam=4;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',4);
guidata(hObject, handles);
% --- Executes on button press in cinco.
function cinco_Callback(hObject, eventdata, handles)
% hObject   handle to cinco (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
numcam=5;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',5);
guidata(hObject, handles);
% --- Executes on button press in siete.
function siete_Callback(hObject, eventdata, handles)
% hObject   handle to siete (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
numcam=7;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',7);
guidata(hObject, handles);
% --- Executes on button press in seis.
function seis_Callback(hObject, eventdata, handles)
% hObject   handle to seis (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
numcam=6;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',6);
guidata(hObject, handles);
% --- Executes on button press in diez.
function diez_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to diez (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numcam=10;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',10);
guidata(hObject, handles);
% --- Executes on button press in nueve.
function nueve_Callback(hObject, eventdata, handles)
% hObject    handle to nueve (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numcam=9;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',9);
guidata(hObject, handles);
% --- Executes on button press in ocho.
function ocho_Callback(hObject, eventdata, handles)
% hObject    handle to ocho (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numcam=8;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',8);
guidata(hObject, handles);
% --- Executes on button press in tres.
function tres_Callback(hObject, eventdata, handles)
% hObject    handle to tres (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numcam=3;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',3);
guidata(hObject, handles);
% --- Executes on button press in dos.
function dos_Callback(hObject, eventdata, handles)
% hObject    handle to dos (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
numcam=2;
handles.numcam=numcam;
set(handles.Segmentos,'UserData',2);
guidata(hObject, handles);
% --- Executes on button press in Canciones.

```

```

function Canciones_Callback(hObject, eventdata, handles)
set(handles.Salida,'String','');
set(handles.Segmento,'String','');
longi=10*44100;
aux=zeros(longi,2);
for i=1:1:10
    switch i
        case 1
            sx=wavread('a ti.wav');
        case 2
            sx=wavread('Cosa Mas Linda Que Tu.wav');
        case 3
            sx=wavread('fran sinatra - new york new york.wav');
        case 4
            sx=wavread('apuesta d rock and roll.wav');
        case 5
            sx=wavread('a la la la la long.wav');
        case 6
            sx=wavread('infinity.wav');
        case 7
            sx=wavread('goodbye my love.wav');
        case 8
            sx=wavread('The Beatles - Yesterday.wav');
        case 9
            sx=wavread('tu amor me hace bien.wav');
        case 10
            sx=wavread('sonata arctica - blank file(2).wav');
    end
    for j=1:1:longi
        aux(j,:)=[sx(j,:)];
    end
    sound(aux,44100);
end
% --- Executes on button press in Segmentos.
function Segmentos_Callback(hObject, eventdata, handles)
% hObject    handle to Segmentos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Salida,'String','');
set(handles.Segmento,'String','');
set(handles.Segmentos,'UserData',0);
pes=10;
lx=randperm(pes);
pio=round(rand*10);

```

```

if pio==0
    pio=1;
end
maxi=max(handles.numseg);
mini=min(handles.numseg);
cantidad=60/handles.tame;
cue=lx(1,pio);
%set(handles.Salida,'String',cue);
if cue==0
    cue=1;
end
switch cue
    case 1
        per=mini;
        serial=(per:1:(per+cantidad-1));
    case 2
        per=mini+cantidad;
        serial=(per:1:(per+cantidad-1));
    case 3
        per=mini+2*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 4
        per=mini+3*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 5
        per=mini+4*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 6
        per=mini+5*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 7
        per=mini+6*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 8
        per=mini+7*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 9
        per=mini+8*cantidad;
        serial=(per:1:(per+cantidad-1));
    case 10
        per=mini+9*cantidad;
        serial=(per:1:(per+cantidad-1));
end
maxim=max(serial);

```

```

minim=min(serial);
n=maxim-minim+1;
l=randperm(n);
OUT=l+minim-1;
r=length(OUT);
a=1;
while (get(handles.Segmentos,'UserData') ==0)
    if a==r+1
        a=1;
    end
    rock=OUT(1,a);
    sx=wavread(sprintf('segmento%05d.wav',rock));
    sound(sx,44100);
    a=a+1;
    drawnow
    %pause(5);
end
if cue==get(handles.Segmentos,'UserData')
    set(handles.Salida,'String','Correcto');
    moni=1;
else
    set(handles.Salida,'String','Incorrecto');
    moni=0;
end
bit=a-2;
set(handles.Segmento,'String',bit);
guidata(hObject, handles);
% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from listbox1
inf=get(hObject,'Value');
gos=get(hObject,'String');
switch inf
    case 1
        set(handles.tama,'string',['1 segundo']);
        numseg=(1:1:600);
        tame=1;
    case 2
        set(handles.tama,'string',['2 segundos']);

```

```

        numseg=(601:1:900);
        tame=2;
    case 3
        set(handles.tama,'string',['3 segundos']);
        numseg=(901:1:1100);
        tame=3;
    case 4
        set(handles.tama,'string',['4 segundos']);
        numseg=(1101:1:1250);
        tame=4;
    case 5
        set(handles.tama,'string',['5 segundos']);
        numseg=(1251:1:1370);
        tame=5;
    case 6
        set(handles.tama,'string',['6 segundos']);
        numseg=(1371:1:1470);
        tame=6;
    case 7
        set(handles.tama,'string',['8 segundos']);
        numseg=(1471:1:1540);
        tame=8;
    case 8
        set(handles.tama,'string',['10 segundos']);
        numseg=(1541:1:1600);
        tame=10;
end
handles.tame=tame;
handles.numseg=numseg;
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function listBox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listBox controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

#### 4. Programa de creación de tablas para gráficas de energía vrs. componentes notables

```
function [compnot]=graficasprob2(tamseg, familia,filename)
%Programa de creación de tablas para gráficas de Energía Vrs. Comp. Not.
%Creado por Kevin Hernandez
[h0 h1 h2 h3]=daubechies(familia);
[retraso]=frp(h0, h1, h2, h3, 0.001);
[y]=wavread(filename);
l=length(y);
yx=zeros(l,1);
for ru=1:1:l
    yx(ru,1)=sum(y(ru,:))/2;
end
r=2^tamseg;
x=floor(l/r);
compnot=zeros(24,x);
promedio=zeros(24,1);
desvest=zeros(24,1);
pcn=zeros(r,1);
for tix=1:1:r
    pcn(tix,1)=tix/r;
end
for rox=1:1:x
    y1=zeros(1,r);
    for k=1:1:r
        y1(1,k)=yx(k+(2^15)*(rox-1),1);
    end
    [DWT]=Wavelet(y1,h0,h1,retraso);
    Potencia=DWT.^2;
    DWTm=sort(Potencia,'descend');
    pot=sum(DWTm);
    DWTt=(DWTm./pot).*100;
    DWTa=zeros(1,length(DWTt));
    num=0;
    for mix=1:1:length(DWTt)
        DWTa(1,mix)=num+DWTt(1,mix);
        num=DWTa(1,mix);
    end
    cor=1;
    for mix=1:1:length(DWTa)
        pot1=DWTa(1,mix);
        if (pot1>=5-0.0000001 && cor==1)
            compnot(cor,rox)=pcn(mix,1);
```

```

    cor=2;
end
if (pot1>=10-0.0000001 && cor==2)
    compnot(cor,rox)=pcn(mix,1);
    cor=3;
end
if (pot1>=15-0.0000001 && cor==3)
    compnot(cor,rox)=pcn(mix,1);
    cor=4;
end
if (pot1>=20-0.0000001 && cor==4)
    compnot(cor,rox)=pcn(mix,1);
    cor=5;
end
if (pot1>=25-0.0000001 && cor==5)
    compnot(cor,rox)=pcn(mix,1);
    cor=6;
end
if (pot1>=30-0.0000001 && cor==6)
    compnot(cor,rox)=pcn(mix,1);
    cor=7;
end
if (pot1>=35-0.0000001 && cor==7)
    compnot(cor,rox)=pcn(mix,1);
    cor=8;
end
if (pot1>=40-0.0000001 && cor==8)
    compnot(cor,rox)=pcn(mix,1);
    cor=9;
end
if (pot1>=45-0.0000001 && cor==9)
    compnot(cor,rox)=pcn(mix,1);
    cor=10;
end
if (pot1>=50-0.0000001 && cor==10)
    compnot(cor,rox)=pcn(mix,1);
    cor=11;
end
if (pot1>=55-0.0000001 && cor==11)
    compnot(cor,rox)=pcn(mix,1);
    cor=12;
end
if (pot1>=60-0.0000001 && cor==12)
    compnot(cor,rox)=pcn(mix,1);

```

```

    cor=13;
end
if (pot1>=65-0.0000001 && cor==13)
    compnot(cor,rox)=pcn(mix,1);
    cor=14;
end
if (pot1>=70-0.0000001 && cor==14)
    compnot(cor,rox)=pcn(mix,1);
    cor=15;
end
if (pot1>=75-0.0000001 && cor==15)
    compnot(cor,rox)=pcn(mix,1);
    cor=16;
end
if (pot1>=80-0.0000001 && cor==16)
    compnot(cor,rox)=pcn(mix,1);
    cor=17;
end
if (pot1>=85-0.0000001 && cor==17)
    compnot(cor,rox)=pcn(mix,1);
    cor=18;
end
if (pot1>=90-0.0000001 && cor==18)
    compnot(cor,rox)=pcn(mix,1);
    cor=19;
end
if (pot1>=95-0.0000001 && cor==19)
    compnot(cor,rox)=pcn(mix,1);
    cor=20;
end
if (pot1>=96-0.0000001 && cor==20)
    compnot(cor,rox)=pcn(mix,1);
    cor=21;
end
if (pot1>=97-0.0000001 && cor==21)
    compnot(cor,rox)=pcn(mix,1);
    cor=22;
end
if (pot1>=98-0.0000001 && cor==22)
    compnot(cor,rox)=pcn(mix,1);
    cor=23;
end
if (pot1>=99-0.0000001 && cor==23)
    compnot(cor,rox)=pcn(mix,1);

```

```

        cor=24;
    end
    if (pot1>=100-0.0000001 && cor==24)
        compnot(cor,rox)=pcn(mix,1);
        break
    end
end
end
pote=[5;10;15;20;25;30;35;40;45;50;55;60;65;70;75;80;85;90;95;96;97;98;99;100];
datos=[compnot pote];
datonumerico=num2cell(datos);
alldata1=[datonumerico];
A=[filename,' ',familia,' ',num2str(tamseg)];
name=sprintf('%s.csv',A);
cell2csv(name,alldata1);
numero=0;
for pr=1:1:24
    promedio(pr,1)=sum(compnot(pr,:))/x;
    desvest(pr,1)=std(compnot(pr,:));
end
graf1=promedio+desvest;
graf2=promedio-desvest;
dt=[promedio desvest graf1 graf2 pote];
dt=num2cell(dt);
A1=[filename,' ',familia,' ',num2str(tamseg),' ','promedio'];
name1=sprintf('%s.csv',A1);
cell2csv(name1,dt);
end

```

## 5. Programa para elección de número de componentes notables a almacenar

```

function [can energia med dest]=decimarw(s,cn,familia,k)
%Programa para elección del número de componentes notables a almacenar
%Creado por Kevin Hernandez
[componentes indices energ u r]=compnot(s, k, familia, cn);
[h2 h3]=daubechies(familia);
can=zeros(u,r);
energia=zeros(1,r);
for t=1:1:r
    m=zeros(1,2^k);
    for x=1:1:cn

```

```

        m(1,indices(t,x)+1)=componentes(t,x);
    end
    ui=lwavelet(m,h2,h3);
    can(:,t)=ui';
    c=energ(t,1);
    if c==0
        energia(1,t)=0;
    else
        energia(1,t)=(sum(can(:,t).^2)/energ(t,1));
    end
end
energia=energia.*100;
med=mean(energia);
dest=std(energia);
y=zeros(r*u,1);
for pit=1:1:r
    y(2^k*(pit-1)+1:2^k*(pit),1)=can(:,pit);
end
yx=wavread(s);
if size(yx,2)==2
    yr=zeros(size(yx,1),1);
    for s=1:1:size(yx,1)
        yr(s,1)=sum(yx(s,:))/2;
    end
    yx=yr;
end
pot=sum(yx.^2);
pot1=sum(y.^2);
size(pot)
size(pot1)
norm=pot1/pot;
y=y.*(1/sqrt(norm));
cn1=num2str(cn);
ext='.wav';
sip=sprintf('%s',s,cn1,ext);
wavwrite(y,44100,sip);
end

```

## Programas de ondeletes

### 1. Programa de creación de filtros Daubechies

```

function [h0 h1 h2 h3]=daubechies(p)
%Programa de filtros Daubechies

```

```

%Creado por Kevin Hernandez
t=[1 1];
t1=1;
for s=1:1:p
    t1=conv(t,t1);
end
t2=[1 -1];
t7=0;
for s1=0:1:p-1
    elev=2*s1;
    if elev==0
        pi1=1;
        t3=1;
    else
        pi=factorial(p+s1-1)/(factorial(s1)*factorial(p-1));
        pi1=(-1)^s1*pi/(2^elev);
        t3=1;
        for s2=1:1:elev
            t3=conv(t3,t2);
        end
    end
    t4=zeros(1,p-1-s1);
    t5=[t4 pi1];
    t6=conv(t5,t3);
    yx=zeros(1,length(t6));
    yx(1,1:1:length(t7))=t7;
    t7=t6+yx;
end
v=roots(t7);
m=size(v,1)/2;
t8=t1;
for s3=1:1:m
    roc=[v(s3,1) -1];
    t1=conv(t1,roc);
end
for s4=m+1:1:2*m
    roc1=[v(s4,1) -1];
    t8=conv(t8,roc1);
end
t1=real(t1);
t8=real(t8);
h2=(-1)^(p+1)*(1/sqrt(sum(t8.^2)))*t8;
h0=(1/sqrt(sum(t1.^2)))*t1;
h1=zeros(1,length(h2));

```

```

h3=zeros(1,length(h2));
for s5=1:1:length(h2)
    h1(1,s5)=(-1)^(s5+1)*h2(1,s5);
    h3(1,s5)=(-1)^(s5)*h0(1,s5);
end
end

```

## 2. Programa de obtención de la transformada de ondelete

```

function [DWT, detalle, aproximacion]=Wavelet(xn,h0,h1,retraso)
%Programa de obtencion de transformada Ondelete
%Creado por Kevin Hernandez
N=length(xn);
m=length(h0);
x=log(N)/log(2); %Nivel j
mix=[];
for j=1:1:m
    mix=[xn mix];
end
detalle=[];
for k=1:1:x %Nivel j
    s0=conv(mix,h0); %Convolucion de señal con h1
    s1=conv(mix,h1); %Convolucion de señal con h2
    s0=s0(retraso+1:N+retraso);
    s1=s1(retraso+1:N+retraso);
    s0=reshape(s0,2,N/2);
    s1=reshape(s1,2,N/2);
    c=s0(1,:);
    d=s1(1,:);
    detalle=[d detalle];
    N=length(c);
    mix=[];
    for j=1:1:m
        mix=[c mix];
    end
end
aproximacion=c;
DWT=[aproximacion detalle];
end

```

## 3. Programa de obtención de la transformada de ondelete inversa

```

function [IDWT]=lwavelet(DWT,h2,h3)
%Programa de obtención de la transformada de Ondelete Inversa
%Creado por Kevin Hernandez
aproximacion=DWT(1,1);
N=length(DWT);
m=length(h2);
detalle=DWT(1,2:N);
u=log(N)/log(2);
c=aproximacion;
cont=1;
for k=0:1:u-1
    x=2^k;
    d=detalle(1,cont:cont+x-1);
    n=length(d);
    du=[];
    cu=[];
    for j=1:1:m
        du=[d du];
        cu=[c cu];
    end
    da=[du; zeros(size(du))];
    ca=[cu; zeros(size(cu))];
    db=da(:)';
    cb=ca(:)';
    cc=conv(cb,h2);
    dc=conv(db,h3);
    ce=(cc+dc);
    c=ce(1,(m-2)*n+1:m*n);
    cont=cont+x;
end
IDWT=c;
end

```

#### 4. Programa de obtención de componentes notables

```

function [componentes indices energia]=compnot1(filename, tam, p, cant, ubic)
%Programa de obtención de componentes notables
%Creado por Kevin Hernandez
if nargin<5
    ubic='C:\Documents and Settings\Kevin\Mis
documentos\Kevin\USAC\Electronica\Tesis\Tesis wavelet\Primeras pruebas\';
end
if nargin<4

```

```

    cant=2^tam;
end
[h0 h1 h2 h3]=daubechies(p);
[retraso]=frp(h0, h1, h2, h3, 0.001);
%filename1=filename;
%Se vectoriza el archivo a leer convirtiendolo de estereo a mono
nombre=sprintf('%s',ubic,filename);
[y]=wavread(nombre);
l=length(y);
yx=zeros(l,1);
if size(y,2)==2
for ru=1:1:l
    yx(ru,1)=sum(y(ru,:))/2;
end
else
    yx=y;
end
%Se crean las matrices a utilizar
componentes=zeros(1,cant);
energia=zeros(1,1);
indices=zeros(1,cant);
[DWT]=Wavelet(yx',h0,h1,retraso);
Potencia=abs(DWT);
DWTm=zeros(1,cant);
pot=sum(DWT.^2);
for t=1:1:cant
    [c,i]=max(Potencia);
    DWTm(1,t)=DWT(1,i);
    Potencia(1,i)=0;
    indices(1,t)=i-1;
end
energia(1,1)=pot;
componentes(1,:)=DWTm;
potm=sum(componentes(1,:).^2);
x=1/potm;
componentes(1,:)=sqrt(x).*componentes(1,:);
end

```

## 5. Programa de revisión de condiciones de reconstrucción perfecta

```

function [retraso,decision,normalizacion]=frp(h0, h1, h2, h3, incertidumbre)
%Programa de revision de condiciones de reconstrucción perfecta
%Creado por Kevin Hernandez

```

```

a=0;
p0=conv(h0,h2);
p1=conv(h1,h3);
y=p0+p1;
longi=length(y);
for i=1:1:longi
    pru=y(1,i);
    if (pru>incertidumbre || pru<-incertidumbre)
        a=a+1;
        mx=pru;
        r=i;
    end
end
if (a>1 || a<1)
    disp('El banco de filtros no es de reconstruccion perfecta');
    decision=0;
    retraso=0;
    normalizacion=0;
end
if a==1
    if (mx<(2+incertidumbre) && mx>(2-incertidumbre))
        disp('El filtro es de reconstruccion perfecta');
        decision=1;
        retraso=r-1;
        normalizacion=1;
    end
    if (mx>2.001 || mx<1.999)
        disp('El banco de filtros no es de reconstruccion perfecta se necesita dividir
por el factor de normalizacion: ');
        retraso=0;
        decision=0;
        normalizacion=sqrt(mx/2);
    end
end
end
end

```

## Programas extras

### 1. Programa para búsqueda en carpetas de canciones

```

function [energia segmento cancion]=elecbs(idsegme,datos,u)
%Programa de busqueda de canciones y segmentos
%Creado por Kevin Hernandez

```

```
sl=size(idsegme,1);
idcancion=zeros(sl,1);
energia=zeros(sl,1);
segmento=zeros(sl,1);
cancion={};
for s=1:1:sl
    idcancion(s,1)=datos(idsegme(s,1)+1,2);
    energia(s,1)=datos(idsegme(s,1)+1,3);
    segmento(s,1)=datos(idsegme(s,1)+1,4);
end
for si=1:1:sl
    u1=u(idcancion(si,1)+1,2);
    cancion=[cancion;u1];
end
end
```

**ANEXO 1**  
**TABLAS DE COEFICIENTES PARA FILTROS DAUBECHIES**



### Filtro DB2

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$
F0	0,483	0,8365	0,2241	-0,1294
F1	-0,1294	-0,2241	0,8365	-0,483

### Filtro DB3

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$
F0	0,3327	0,8069	0,4599	-0,135	-0,0854	0,0352
F1	0,0352	0,0854	-0,135	-0,4599	0,8069	-0,3327

### Filtro DB4

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$	$z^{-6}$	$z^{-7}$
F0	0,2304	0,7148	0,6309	-0,028	-0,187	0,0308	0,0329	-0,0106
F1	-0,0106	-0,0329	0,0308	0,187	-0,028	-0,6309	0,7148	-0,2304

### Filtro DB5

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$
F0	0,1601	0,6038	0,7243	0,1384	-0,2423
	$z^{-5}$	$z^{-6}$	$z^{-7}$	$z^{-8}$	$z^{-9}$
	-0,0322	0,0776	-0,0062	-0,0126	0,0033
Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$
F1	0,0033	0,0126	-0,006	-0,0776	-0,0322
	$z^{-5}$	$z^{-6}$	$z^{-7}$	$z^{-8}$	$z^{-9}$
	0,2423	0,1384	-0,7243	0,6038	-0,1601

### Filtro DB6

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$
F0	0,1115	0,4946	0,7511	0,3153	-0,2263	-0,1298
	$z^{-6}$	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$
	0,0975	0,0275	-0,0316	0,0006	0,0048	-0,0011
Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$
F1	-0,0011	-0,0048	0,0006	0,0316	0,0275	-0,0975
	$z^{-6}$	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$
	-0,13	0,2263	0,3153	-0,7511	0,4946	-0,1115

### Filtro DB7

Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F0	0,0779	0,3965	0,7291	0,4698	-0,1439	-0,224	0,0713
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	0,0806	-0,038	-0,0166	0,0126	0,0004	-0,0018	0,0004
Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F1	0,0004	0,0018	0,0004	-0,0126	-0,0166	0,038	0,0806
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	-0,0713	-0,224	0,1439	0,4698	-0,7291	0,3965	-0,0779

### Filtro DB8

Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$	$z^7$
F0	0,0544	0,3129	0,6756	0,5854	-0,0158	-0,284	0,0005	0,1287
	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$	$z^{14}$	$z^{15}$
	-0,0174	-0,0441	0,014	0,0087	-0,0049	-0,0004	0,0007	-0,0001
Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$	$z^7$
F1	-0,0001	-0,0007	-4E-04	0,0049	0,0087	-0,014	-0,044	0,0174
	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$	$z^{14}$	$z^{15}$
	0,1287	-0,0005	-0,284	0,0158	0,5854	-0,6756	0,3129	-0,0544

### Filtro DB9

Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F0	0,0381	0,2438	0,6048	0,6573	0,1332	-0,2933	-0,097
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	0,1485	0,0307	-0,0676	0,0003	0,0224	-0,0047	-0,0043
	$z^{14}$	$z^{15}$	$z^{16}$	$z^{17}$			
	0,0018	0,0002	-0,0003	0			
Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F1	0	0,0003	0,0002	-0,0018	-0,0043	0,0047	0,0224
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	-0,0003	-0,0676	-0,0307	0,1485	0,0968	-0,2933	-0,1332
	$z^{14}$	$z^{15}$	$z^{16}$	$z^{17}$			
	0,6573	-0,6048	0,2438	-0,0381			

### Filtro DB10

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F0	0,0267	0,1882	0,5272	0,6885	0,2812	-0,2498	-0,196
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	0,1274	0,0931	-0,0714	-0,03	0,0332	0,0036	-0,0107
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	
	0,0014	0,002	-0,0007	-0,0001	0,0001	0	
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F1	0	-0,0001	-1E-04	0,0007	0,002	-0,0014	-0,011
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	-0,0036	0,0332	0,0295	-0,071	-0,0931	0,1274	0,1959
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	
	-0,2498	-0,2812	0,6885	-0,5272	0,1882	-0,0267	

### Filtro DB11

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>
F0	0,0187	0,1441	0,4499	0,6857	0,412	-0,1623	-0,274	0,066
	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>
	0,1498	-0,0465	-0,066	0,0313	0,0208	-0,0154	-0,0033	0,0049
	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>		
	-0,0003	-0,0009	0,0002	0,0001	0	0		
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>
F1	0	0	0,0001	-0,0002	-0,0009	0,0003	0,0049	0,0033
	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>
	-0,0154	-0,0208	0,0313	0,0664	-0,0465	-0,1498	0,066	0,2742
	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>		
	-0,1623	-0,412	0,6857	-0,4499	0,1441	-0,0187		

### Filtro DB12

Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$	$z^7$
F0	0,0131	0,1096	0,3774	0,6572	0,5159	-0,0448	-0,316	-0,0238
	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$	$z^{14}$	$z^{15}$
	0,1825	0,0054	-0,096	0,0108	0,0415	-0,0122	-0,0128	0,0067
	$z^{16}$	$z^{17}$	$z^{18}$	$z^{19}$	$z^{20}$	$z^{21}$	$z^{22}$	$z^{23}$
	0,0022	-0,0022	0	0,0004	-0,0001	0	0	0
Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$	$z^7$
F1	0	0	0	0,0001	0,0004	0	-0,002	-0,0022
	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$	$z^{14}$	$z^{15}$
	0,0067	0,0128	-0,012	-0,0415	0,0108	0,0964	0,0054	-0,1825
	$z^{16}$	$z^{17}$	$z^{18}$	$z^{19}$	$z^{20}$	$z^{21}$	$z^{22}$	$z^{23}$
	-0,0238	0,3162	-0,0448	-0,5159	0,6572	-0,3774	0,1096	-0,0131

### Filtro DB13

Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F0	0,0092	0,0829	0,3120	0,6111	0,5889	0,0870	-0,3150
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	-0,1246	0,1795	0,0729	-0,1058	-0,0265	0,0561	0,0024
	$z^{14}$	$z^{15}$	$z^{16}$	$z^{17}$	$z^{18}$	$z^{19}$	$z^{20}$
	-0,0238	0,0039	0,0073	-0,0028	-0,0013	0,0009	0,0000
	$z^{21}$	$z^{22}$	$z^{23}$	$z^{24}$	$z^{25}$		
-0,0002	0	0	-0	0			
Filtro	$z^0$	$z^1$	$z^2$	$z^3$	$z^4$	$z^5$	$z^6$
F1	0	0	0	-0	-0,0002	-0	0,0009
	$z^7$	$z^8$	$z^9$	$z^{10}$	$z^{11}$	$z^{12}$	$z^{13}$
	0,0013	-0,0028	-0,0073	0,0039	0,0238	0,0024	-0,0561
	$z^{14}$	$z^{15}$	$z^{16}$	$z^{17}$	$z^{18}$	$z^{19}$	$z^{20}$
	-0,0265	0,1058	0,0729	-0,1795	-0,1246	0,3150	0,0870
	$z^{21}$	$z^{22}$	$z^{23}$	$z^{24}$	$z^{25}$		
-0,5889	0,6111	-0,3120	0,0829	-0,0092			

### Filtro DB14

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$	$z^{-6}$
F0	0,0065	0,0624	0,2539	0,5543	0,6312	0,2187	-0,2717
	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$	$z^{-12}$	$z^{-13}$
	-0,2180	0,1384	0,1400	-0,0867	-0,0715	0,0552	0,0270
	$z^{-14}$	$z^{-15}$	$z^{-16}$	$z^{-17}$	$z^{-18}$	$z^{-19}$	$z^{-20}$
	-0,0302	-0,0056	0,0128	-0,0007	-0,0038	0,0011	0,0007
	$z^{-21}$	$z^{-22}$	$z^{-23}$	$z^{-24}$	$z^{-25}$	$z^{-26}$	$z^{-27}$
	-0,0004	-0	0,0001	-0	-0	0	-0
Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$	$z^{-6}$
F1	-0	-0	-0	0	0,0001	0	-0,0004
	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$	$z^{-12}$	$z^{-13}$
	-0,0007	0,0011	0,0038	-0,0007	-0,0128	-0,0056	0,0302
	$z^{-14}$	$z^{-15}$	$z^{-16}$	$z^{-17}$	$z^{-18}$	$z^{-19}$	$z^{-20}$
	0,0270	-0,0552	-0,0715	0,0867	0,1400	-0,1384	-0,2180
	$z^{-21}$	$z^{-22}$	$z^{-23}$	$z^{-24}$	$z^{-25}$	$z^{-26}$	$z^{-27}$
	0,2717	0,2187	-0,6312	0,5543	-0,2539	0,0624	-0,0065

### Filtro DB15

Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$	$z^{-6}$
F0	0,0045	0,0467	0,2060	0,4926	0,6458	0,3390	-0,1932
	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$	$z^{-12}$	$z^{-13}$
	-0,2889	0,0653	0,1901	-0,0397	-0,1111	0,0339	0,0548
	$z^{-14}$	$z^{-15}$	$z^{-16}$	$z^{-17}$	$z^{-18}$	$z^{-19}$	$z^{-20}$
	-0,0258	-0,0208	0,0151	0,0051	-0,0065	-0,0002	0,0019
	$z^{-21}$	$z^{-22}$	$z^{-23}$	$z^{-24}$	$z^{-25}$	$z^{-26}$	$z^{-27}$
	-0,0004	-0,0004	0,0002	0	-0	0	0
	$z^{-28}$	$z^{-29}$					
	-0	0					
Filtro	$z^0$	$z^{-1}$	$z^{-2}$	$z^{-3}$	$z^{-4}$	$z^{-5}$	$z^{-6}$
F1	0	0	0	-0	-0	-0	0,0002
	$z^{-7}$	$z^{-8}$	$z^{-9}$	$z^{-10}$	$z^{-11}$	$z^{-12}$	$z^{-13}$
	0,0004	-0,0004	-0,0019	-0,0002	0,0065	0,0051	-0,0151
	$z^{-14}$	$z^{-15}$	$z^{-16}$	$z^{-17}$	$z^{-18}$	$z^{-19}$	$z^{-20}$
	-0,0208	0,0258	0,0548	-0,0339	-0,1111	0,0397	0,1901
	$z^{-21}$	$z^{-22}$	$z^{-23}$	$z^{-24}$	$z^{-25}$	$z^{-26}$	$z^{-27}$
	-0,0653	-0,2889	0,1932	0,3390	-0,6458	0,4926	-0,2060
	$z^{-28}$	$z^{-29}$					
	0,0467	-0,0045					

### Filtro DB16

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F0	0,0032	0,0349	0,1651	0,4304	0,6374	0,4403	-0,0898
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	-0,3271	-0,0279	0,2112	0,0273	-0,1324	-0,0062	0,0759
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>
	-0,0076	-0,0369	0,0103	0,0140	-0,0070	-0,0036	0,0031
	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>
	0,0004	-0,0009	0,0001	0,0002	-0,0001	-0	0
	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>			
-0	-0	0	-0				
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F1	-0	-0	-0	0	0	0	-0,0001
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	-0,0002	0,0001	0,0009	0,0004	-0,0031	-0,0036	0,0070
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>
	0,0140	-0,0103	-0,0369	0,0076	0,0759	0,0062	-0,1324
	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>
	-0,0273	0,2112	0,0279	-0,3271	0,0898	0,4403	-0,6374
	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>			
0,4304	-0,1651	0,0349	-0,0032				

### Filtro DB17

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F0	0,0022	0,0260	0,1312	0,3704	0,6110	0,5183	0,0273
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	-0,3283	-0,1266	0,1973	0,1011	-0,1268	-0,0571	0,0811
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>
	0,0223	-0,0469	-0,0033	0,0227	-0,0030	-0,0086	0,0030
	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>
	0,0023	-0,0014	-0,0003	0,0004	-0	-0,0001	0
	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>	Z <sup>-32</sup>	Z <sup>-33</sup>	
0	-0	0	0	-0	0		
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>
F1	0	0	0	-0	-0	-0	0
	z <sup>-7</sup>	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>
	0,0001	-0	-0,0004	-0,0003	0,0014	0,0023	-0,0030
	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>
	-0,0086	0,0030	0,0227	0,0033	-0,0469	-0,0223	0,0811
	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>
	0,0571	-0,1268	-0,1011	0,1973	0,1266	-0,3283	-0,0273
	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>	Z <sup>-32</sup>	Z <sup>-33</sup>	
0,5183	-0,6110	0,3704	-0,1312	0,0260	-0,0022		

### Filtro DB18

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F0	0,0016	0,0193	0,1036	0,3147	0,5718	0,5718	0,1472	-0,2937	-0,2165
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	0,1495	0,1671	-0,0923	-0,1068	0,0649	0,0571	-0,0445	-0,0237	0,0267
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>
	0,0063	-0,0131	0,0001		0,0049	-0,0011	-0,0013	0,0006	0,0002
	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>	Z <sup>-32</sup>	Z <sup>-33</sup>	Z <sup>-34</sup>	Z <sup>-35</sup>	
	0	0	0	0	0	0	0	0	
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F1	0	0	0	0	0	0	0	0	0
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	0,0002	0,0002	-0,0006	-0,0013	0,0011	0,0049	-0,0001	-0,0131	-0,0063
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>
	0,0267	0,0237	-0,0445	-0,0571	0,0649	0,1068	-0,0923	-0,1671	0,1495
	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>	Z <sup>-32</sup>	Z <sup>-33</sup>	Z <sup>-34</sup>	Z <sup>-35</sup>
	0,2165	-0,2937	-0,1472	0,5718	-0,5718	0,3147	-0,1036	0,0193	-0,0016

### Filtro DB19

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>
F0	0,0011	0,0143	0,0813	0,2644	0,5244	0,6017	0,2609	-0,2281
	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>
	-0,2858	0,0747	0,2123	-0,0335	-0,1428	0,0276	0,0869	-0,0265
	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>
	-0,0457	0,0216	0,0194	-0,014	-0,0059	0,007	0,0008	-0,0027
	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>
	0,0003	0,0007	-0,0003	-0,0001	0,0001	0	0	0
Z <sup>-32</sup>	Z <sup>-33</sup>	Z <sup>-34</sup>	Z <sup>-35</sup>	Z <sup>-36</sup>	Z <sup>-37</sup>			
0	0	0	0	0	0			
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>
F1	0	0	0	0	0	0	0	0
	z <sup>-8</sup>	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>
	0	-0,0001	-0,0001	0,0003	0,0007	-0,0003	-0,0027	-0,0008
	z <sup>-16</sup>	z <sup>-17</sup>	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>
	0,007	0,0059	-0,014	-0,0194	0,0216	0,0457	-0,0265	-0,0869
	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	Z <sup>-30</sup>	Z <sup>-31</sup>
	0,0276	0,1428	-0,0335	-0,2123	0,0747	0,2858	-0,2281	-0,2609
Z <sup>-32</sup>	Z <sup>-33</sup>	Z <sup>-34</sup>	Z <sup>-35</sup>	Z <sup>-36</sup>	Z <sup>-37</sup>			
0,6017	-0,5244	0,2644	-0,0813	0,0143	-0,0011			

### Filtro DB20

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F0	0,0008	0,0105	0,0634	0,2199	0,4727	0,6105	0,3615	-0,1392	-0,3268
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	-0,0167	0,2283	0,0399	-0,1555	-0,0247	0,1023	0,0056	-0,0617	0,0059
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>
	0,0323	-0,0088	-0,0138	0,0067	0,0044	-0,0036	-0,0008	0,0014	-0,0001
	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	z <sup>-30</sup>	z <sup>-31</sup>	z <sup>-32</sup>	z <sup>-33</sup>	z <sup>-34</sup>	z <sup>-35</sup>
	-0,0004	0,0001	0,0001	0	0	0	0	0	0
	z <sup>-36</sup>	z <sup>-37</sup>	z <sup>-38</sup>	z <sup>-39</sup>					
0	0	0	0						
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F1	0	0	0	0	0	0	0	0	0
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	0	0,0001	-0,0001	-0,0004	0,0001	0,0014	0,0008	-0,0036	-0,0044
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>
	0,0067	0,0138	-0,0088	-0,0323	0,0059	0,0617	0,0056	-0,1023	-0,0247
	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	z <sup>-30</sup>	z <sup>-31</sup>	z <sup>-32</sup>	z <sup>-33</sup>	z <sup>-34</sup>	z <sup>-35</sup>
	0,1555	0,0399	-0,2283	-0,0167	0,3268	-0,1392	-0,3615	0,6105	-0,4727
	z <sup>-36</sup>	z <sup>-37</sup>	z <sup>-38</sup>	z <sup>-39</sup>					
0,2199	-0,0634	0,0105	-0,0008						

### Filtro DB21

Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F0	0,0005	0,0078	0,0492	0,1814	0,4197	0,6015	0,4446	-0,0357	-0,3357
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	-0,1124	0,2116	0,1152	-0,1399	-0,0818	0,0966	0,0457	-0,065	-0,0187
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>
	0,0397	0,0034	-0,0209	0,0024	0,009	-0,0029	-0,003	0,0017	0,0006
	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	z <sup>-30</sup>	z <sup>-31</sup>	z <sup>-32</sup>	z <sup>-33</sup>	z <sup>-34</sup>	z <sup>-35</sup>
	-0,0007	0	0,0002	0	0	0	0	0	0
	z <sup>-36</sup>	z <sup>-37</sup>	z <sup>-38</sup>	z <sup>-39</sup>	z <sup>-40</sup>	z <sup>-41</sup>			
0	0	0	0	0	0				
Filtro	z <sup>0</sup>	z <sup>-1</sup>	z <sup>-2</sup>	z <sup>-3</sup>	z <sup>-4</sup>	z <sup>-5</sup>	z <sup>-6</sup>	z <sup>-7</sup>	z <sup>-8</sup>
F1	0	0	0	0	0	0	0	0	0
	z <sup>-9</sup>	z <sup>-10</sup>	z <sup>-11</sup>	z <sup>-12</sup>	z <sup>-13</sup>	z <sup>-14</sup>	z <sup>-15</sup>	z <sup>-16</sup>	z <sup>-17</sup>
	0	0	0	0,0002	0	-0,0007	-0,0006	0,0017	0,003
	z <sup>-18</sup>	z <sup>-19</sup>	z <sup>-20</sup>	z <sup>-21</sup>	z <sup>-22</sup>	z <sup>-23</sup>	z <sup>-24</sup>	z <sup>-25</sup>	z <sup>-26</sup>
	-0,0029	-0,009	0,0024	0,0209	0,0034	-0,0397	-0,0187	0,065	0,0457
	z <sup>-27</sup>	z <sup>-28</sup>	z <sup>-29</sup>	z <sup>-30</sup>	z <sup>-31</sup>	z <sup>-32</sup>	z <sup>-33</sup>	z <sup>-34</sup>	z <sup>-35</sup>
	-0,0966	-0,0818	0,1399	0,1152	-0,2116	-0,1124	0,3357	-0,0357	-0,4446
	z <sup>-36</sup>	z <sup>-37</sup>	z <sup>-38</sup>	z <sup>-39</sup>	z <sup>-40</sup>	z <sup>-41</sup>			
0,6015	-0,4197	0,1814	-0,0492	0,0078	-0,0005				

