UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

MAESTRIA EN INGENIERIA (INGENIERIA ELÉCTRICA)

DESARROLLO DE UN SISTEMA DE IDENTIFICACION DE CANCIONES POR TARAREO PARA DISPOSITIVOS MOVILES.

TESIS
QUE PARA OPTAR POR EL GRADO DE
MAESTRO EN INGENIERIA ELECTRICA ORIENTACION TELECOMUNICACIONES.

PRESENTA:
MARTIN SALGADO ESTRADA

TUTOR:
DR.VICTOR GARCIA GARDUÑO
FACULTAD DE INGENIERIA

MIEMBROS DEL COMITÉ:

PRESIDENTE: DR.JOSÉ MARÍA MATÍAS MARURI
FACULTAD DE INGENIERIA

SECRETARIO: DR. JULIO CÉSAR TINOCO MAGAÑA
FACULTAD DE INGENIERIA

VOCAL: DR. VICTOR GARCÍA GARDUÑO
FACULTAD DE INGENIERIA

1ER.SUPLENTE: DR. JAVIER GOMEZ CASTELLANOS
FACULTAD DE INGENIERIA

2DO.SUPLENTE: DR.FRANCISCO GARCÍA UGALDE
FACULTAD DE INGENIERIA

MÉXICO D.F. ENERO 2012

JURADO ASIGNADO:

Presidente:	Dr. Matías Maruri José María
Secretario:	Dr. Tinoco Magaña Julio Cesar
Vocal:	Dr. García Garduño Víctor
1er.Suplente:	Dr. Gómez Castellanos Javier
2do.Suplente:	Dr. García Ugalde Francisco

México D.F. Ciudad Universitaria, Facultad de Ingeniería.

Tutor de tesis:
Dr. Víctor García Garduño

_____

Firma

Agradecimientos.

Agradezco primeramente a mi familia; que me ha apoyado todo este tiempo incondicionalmente, especialmente a mi padre y madre; que han hecho lo mejor que han podido para sacarme adelante y no rendirme en nada de lo que hago. A mi gemelo Fernando y mi hermana Citlalli; que han sido mis mejores amigos desde pequeños y aun seguimos juntos.

Quiero agradecer a Alejandro Benítez, uno de mis mejores amigos por el apoyo y todos aquellos momentos que hemos pasado desde que empezamos la carrera como ingenieros. A todos los compañeros y amigos que he logrado hacer durante mis estudios de maestría; la he pasado muy bien con ustedes, en especial a Pablo, Senyazen, Eddy y Abril; con que compartido muy buenos momentos y ahora forman parte de mi vida.

Quiero agradecer en especial a mi tutor de tesis el Dr. Víctor García Garduño; por todo el apoyo que me ha brindado a lo largo de maestría y por todo el conocimiento que me ha compartido, la convivencia y la paciencia.

Índice.

Índice de figuras

Pagina

Desarrollo de un sistema de reconocimiento de canciones por tarareo para dispositivos móviles.

Introducción

En los últimos años las aplicaciones en dispositivos móviles han tenido un auge impresionante. Ahora contamos con innumerables servicios, juegos, búsquedas o todo tipo de entretenimientos al alcance de nuestras manos dentro de nuestros teléfonos inteligentes, tabletas o PDA.

Desde los clásicos buscadores que se han usado ya hace algunos años, como Google o Yahoo! que ahora tienen sus propias aplicaciones móviles [1,3]. Aunque el desarrollo de estas nuevas herramientas no solo se centre en búsquedas, estas son muy requeridas por los usuarios. Las nuevas aplicaciones en búsquedas pueden ser llevadas para el análisis de algo en específico; como son búsquedas de video, imágenes o audio.

La identificación de información es una herramienta útil y muy requerida. Existen diversas aplicaciones especializadas en identificación de imágenes o videos, los cuales su principal función es dar información al usuario sobre lo que esta observando. En [4] se cuenta con una aplicación cuya finalidad es identificar texto en japonés para posteriormente ser traducido al idioma que se requiera; siendo una herramienta útil para personas que desconocen el idioma o simplemente para el aprendizaje de este. Otras aplicaciones son empleadas para reconocimiento de otro tipo de caracteres. En [5] los caracteres que se identifican son códigos de barras capturados por las cámaras integradas en los dispositivos, que juegan un papel importante en la búsqueda de ofertas comerciales; así siendo de suma utilidad para encontrar el ítem por el mejor precio.

Dentro del todo el mar existente de herramientas de búsqueda existen por supuesto aquellas especializadas en música. Desde aquellos mercados virtuales de venta de música, hasta la identificación de canciones por diversos métodos como es el caso de este trabajo. Algunos de los métodos más ordinarios son las búsquedas básicas, es decir: por autor, nombre, fechas, etc. Pero existen aquellas que van más allá identificando las canciones por métodos como la captura de audio. Una de ellas es la búsqueda por muestra de audio; [5] este tipo de búsquedas consiste en encontrar un candidato de identificación a base de una captura de audio con un dispositivo móvil. En otras palabras identificar la canción que estoy escuchando. Este tipo de búsqueda es mas sofisticada he interesante, ya que en este proceso se analiza el audio para encontrar parámetros característicos de la canción, que ayuden a su clasificación e identificación.

En la búsqueda que se desarrollo en este trabajo se centra en la identificación de música por tarareo o QBH de sus siglas en ingles. Este consiste en identificar una canción a través de una melodía cantada o tarareada por el usuario. En principio suena algo complicado y lo es; tomando en cuenta de que existen millones de canciones y que cada una de estas canciones

cuentan con sus propias características, las cuales deben de ser identificadas para poder realizar una aplicación de este tipo.

Por otro lado existen diversas investigaciones e incluso aplicaciones funcionales. Aunque aún queda mucho por descubrir en este tema. En este trabajo se proponen dos formas de hacer una identificación. La primera de ellas se basa en muchos de las propuestas ya realizadas; se trata de un análisis que extrae un descriptor de la canción en base a sus frecuencias fundamentales a través del tiempo. En resumen identificar las notas musicales involucradas en la canción. Aunque en esta propuesta la base del análisis solo se trata para un solo instrumento en nuestro caso la voz. Este tipo de análisis lo denominaremos como monofónico; ya que solamente habrá de intervenir un solo instrumento a la vez, el cual se analizará para obtener su descriptor característico. Para este análisis existen distintos modelos para obtención de un descriptor. La mayoría de estos se basan en la obtención del pitch ya que esta será la herramienta principal para la realización de esta aplicación.

En el segundo caso se propone una forma de obtención de un descriptor para hacer la misma identificación. Pero en este caso se involucraran más de un instrumento denominado aquí como polifónico. En el cual se tratan de identificar los elementos más importantes o sobresalientes de la canción original; es decir, se hará un comparativo de la canción original y la voz del usuario. Este segundo es aún mas complejo, ya que para obtener un buen descriptor debemos tener en cuenta cada uno de los sonidos que forman la canción; y discernir dentro de estos, cuales son los elementos mas característicos de ésta. Aunque este sistema se basa principalmente en el pitch. La obtención del mismo esta hecho de otra manera, para poder obtener una huella digital de la canción y con esta adquirir el descriptor adecuado para su identificación.

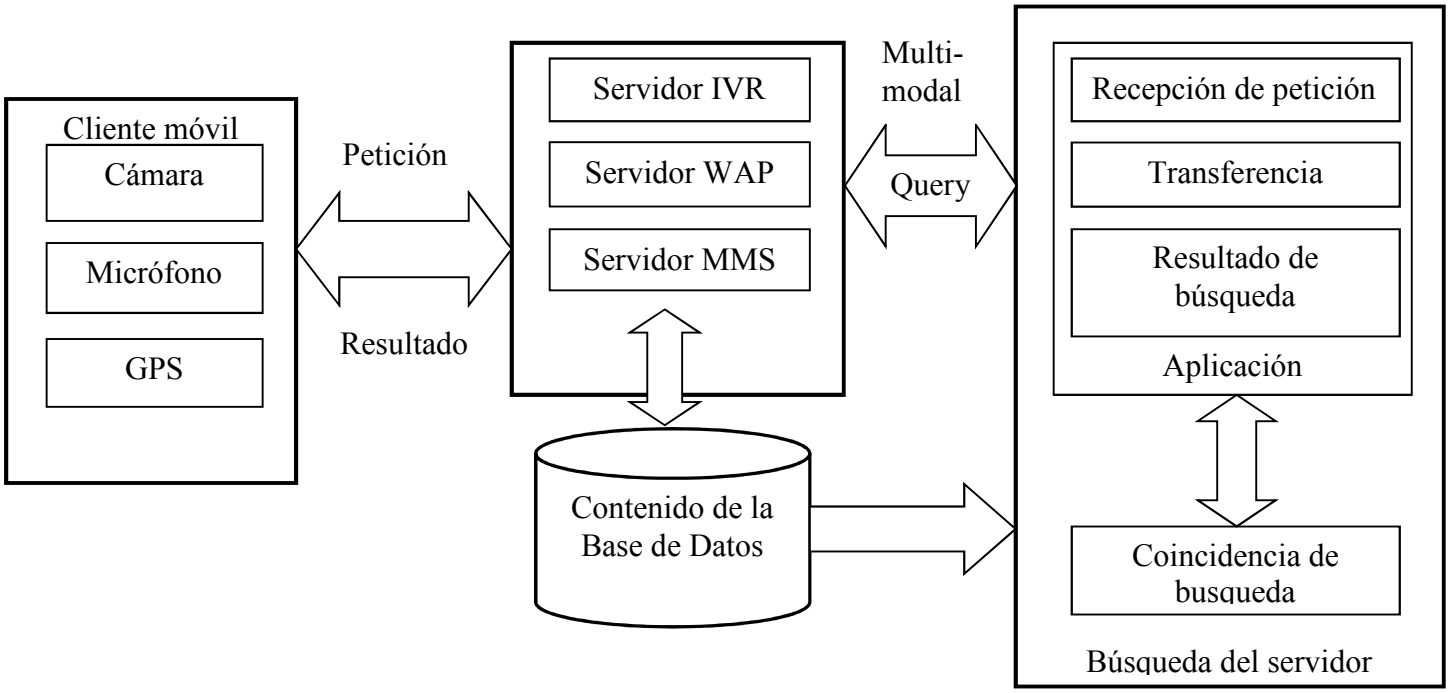


Figura A. El *diseño típico de los sistemas multimodales de búsqueda móvil.*

Basada en la estructura de una búsqueda móvil figura A. Definimos el funcionamiento esta primeramente accediendo al micrófono del dispositivo para hacer una captura del tarareo; para así mandar la petición de búsqueda al servidor. Una vez en nuestro servidor este hará el análisis del archivo de audio mandado por el usuario, para obtener el descriptor con el que se realizará la búsqueda de coincidencia con los datos de todas las canciones almacenadas en nuestra base de datos. Una vez teniendo los mejores candidatos, estos serán devueltos al usuario para que tenga la posibilidad de adquirir la canción deseada.

Este trabajo se centrará principalmente en la obtención del mejor descriptor para la identificación de la canción; en el método de identificación y otras características que hagan más eficiente el sistema, que son los principales elementos de esta aplicación.

Capitulo 1.

Estado del Arte

1.1 Panorama general

El ser humano siempre se ha caracterizado por aprender y adquirir información de la manera más sencilla y práctica posible. Por lo que en la actualidad no es nuevo suponer hacia donde se dirige la tecnología, lo cual es facilitar la vida del ser humano. Es decir comunicarse de inmediato con las personas que él desea, obtener información cuando la desee y hasta entretenerse esté donde esté. Ya sea con sus redes sociales, viendo películas, noticias o cualquier otra forma de entretenimiento que puedan adquirir de inmediato.

Los dispositivos móviles como celulares, PDA, tabletas, etcétera. Son dispositivos que últimamente han ido evolucionando a grandes pasos, debido a la relativa facilidad de desarrollar aplicaciones diversas para éstos. Ya que acceden a varias fuentes de información en un instante. Es por eso, que cada vez hay mas desarrollo de estas que se suelen ofrecer en diversos mercados virtuales. Ya sean gratuititas o por algún precio según sea proveedor que la ofrece.

Una de las fuentes de información mas solicitadas para los usuarios de estos dispositivos es la música. Ya que es una fuente de entretenimiento que siempre ha existido en la vida del ser humano y siempre será por decirlo de alguna manera indispensable. Tal es así que muchas de las aplicaciones o software de computación son dedicados para la música. Sea para la reproducción de archivos de audio o hasta la edición, masterización y mezcla de música.

MPEG-7 es uno de los estándares que se encargan de dar las recomendaciones para archivos multimedia, a grandes rasgos audio y video entre otras cosas. Estas a su vez contienen ciertos descriptores ya sean de imágenes, video o audio. De esta manera pueden ser clasificados para ser obtenidos con relativa facilidad por los usuarios. Para las

aplicaciones que están en desarrollo o están ya al mercado es necesario proponer nuevos descriptores, para que la clasificación y las búsquedas que se obtenga sean satisfactorias.

No obstante, no cualquier descriptor puede ser usado para la clasificación de la música imágenes o video. Esto dependerá del tipo de aplicación que se desea realizar; ya que hay descriptores tan robustos que el proceso de análisis requiere de mucho tiempo y capacidad de procesamiento, que simplemente resulta infructuoso para la aplicación y la búsqueda. A pesar de que los dispositivos móviles son cada vez mas poderosos, estos problemas en ocasiones provocarían conflictos lo cual llevaría a la fallas y errores en la búsqueda.

Las telecomunicaciones es un área tan extensa que difícilmente se puede determinar en cuantas áreas se divide. Este trabajo de desarrollo esta enfocado a telecomunicaciones por lo que se tomaran aspectos representativos de esta. Cabe notar, que el desarrollo y análisis de esta aplicación esta fuertemente relacionado con el procesamiento del audio, así que puede llegarse a profundizarse en este sin dejar de estar relacionado con el fin de mandar información a distancia. Es decir, dar a conocer al usuario esté donde esté el nombre de la canción y el archivo de audio de ser posible. Cumpliendo así las expectativas de dar un servicio de telecomunicaciones.

Existen diversidad de herramientas relacionadas con el audio, que el usuario de una computadora o un dispositivo móvil puede utilizar. En el QBH (Query By Humming) para dispositivos móviles, se tienen que tomar en cuenta cosas muy básicas pero importantes. Las capacidades de un dispositivo móvil, en cuanto a la cantidad de información que puede procesar y a la cantidad de información que puede mandar. Esto influye en que debemos tomar archivos de audio con baja calidad, debido a que la grabación que podemos obtener de un dispositivo móvil esta limitada. Además factores como la información que se puede mandar a través de la red que esta sujeta a ciertas tasas de transmisión.

En el desarrollo de este sistema se tomo en cuenta una baja calidad de audio, asignando un solo canal con una tasa de muestreo de 8000Hz y limitando los fragmentos a grabar a no más de 20 segundos. Así teniendo parámetros fijos para el análisis de la canción; dificultando ésta tarea por la cantidad de ruido o efectos no deseados en el procesado de la misma, tomando en cuenta que se desarrollaran dos posibles caminos para el sistema.

Uno de ellos es tomar diversas muestras de tarareos de un determinado número de personas. Es decir voz contra voz y la segunda y mas interesante alternativa, es la de analizar la música para lograr modelarla, modelar la voz y hacer las respectivas comparaciones para que esta pueda ser clasificada. El segundo sistema es mucho mas elaborado que el primero, ya que se requiere de tener un buen descriptor por ambas partes lo suficientemente preciso para poder compararlas entre si. Así que en el segundo caso se simulará el sistema y del primero se creará un boceto del sistema lo suficientemente grande como para comparar entre una base de datos de por lo menos 50 canciones.

1.2 Descriptores de Audio (MPEG7).

MPEG7 consiste en un estándar de recopilación de metadatos ya sean de imágenes, audio o video. Hoy en día se ha popularizado la búsqueda de diversos servicios y aplicaciones audiovisuales. Dispositivos como son cámaras fotográficas, teléfonos inteligentes, PDA, consolas de video juegos y otros portátiles cuentan con estos servicios. Dado a las diversas necesidades que se han estado adquiriendo entre los usuarios, se deben crear nuevas aplicaciones y servicios para estas plataformas.

Para lo cual el estándar MPEG7 puede ser de gran ayuda para encontrar algunos de los parámetros requeridos para nuestro análisis de audio. Y es en estos descriptores en los cuales nos centraremos. Como es conocido las máquinas de búsqueda existentes como los buscadores de internet utilizan ciertos filtros para llegar a la información requerida por el usuario. Para el análisis digital de un archivo de audio que contiene música, voz, y otras entidades dentro del archivo, pueden realizarse diversos análisis. Para obtener diversos parámetros tales como cuantas voces interpretan la canción, los instrumentos que la ejecutan, el ritmo y/o género que interpretan entre otros.

El objetivo de los descriptores es obtener la mayor información posible de la forma de onda de la señal de audio. Cabe mencionar, que se conoce como audio a cualquier señal audible y no necesariamente se trata de música; ya que está basada en ciertas "reglas" y sus combinaciones.

Los descriptores son compactos y son ideales para aplicaciones de búsqueda. Pero también son eficientes para clasificación, reconocimiento, filtros de datos y navegación. Un descriptor puede ser llamado huella digital o vector característico, ya que estos contienen alguna o algunas de las características de la señal de audio. Estas huellas digitales pueden ser utilizadas para hacer comparaciones, que pueden ser aprovechadas en la búsqueda de información que será la fuente para las implementaciones que se puedan realizar.

Los descriptores audio se han estudiado desde 1970, y a través de las décadas han ido incorporándose diversos análisis acerca de las señales y cada vez hay más demanda de archivos multimedia. Desde entrados los años 90's se consiguieron describir ciertos parámetros de la música; como es la distribución de la energía espectral, el radio armónico o frecuencia fundamental. Los cuales permiten la comparación con otros sonidos que pueden considerarse similares, en especial la frecuencia fundamental es descriptor en el cual se basa la aplicación que se describe en este trabajo.

En el estándar MPEG7 [13] quedan definidos algunos de los descriptores de audio, los cuales sirven para diversos análisis como se explica a continuación:

- Información de almacenamiento y características de contenido: las cuales son el formato de almacenamiento y la decodificación del mismo.
- Información relacionada con el uso del contenido: se abarcan punteros de derechos de autor, historial de uso y el horario de difusión.

- La información que describe los procesos de creación y producción de los contenidos: El director, el autor, el título, etc.
- La información estructural de los componentes temporales de los contenidos: la duración de la canción.
- Información sobre las características de bajo nivel en el contenido: distribución del espectro de energía, timbres del sonido, descripción de la melodía etc.
- Información conceptual sobre la realidad que expresa el contenido: interacción entre los eventos y objetos.
- Información sobre cómo navegar por el contenido de una manera eficiente.
- Información sobre las colecciones de objetos.
- Información sobre la interacción del usuario con el contenido: historial y preferencias de su uso.

Estos descriptores pueden ser usados como un filtro, parámetro o referencia para encontrar las características deseadas del usuario por ejemplo: nombre del artista, año del álbum longitud del archivo etcétera. Entre mejor sean los descriptores, o mayor información contengan la búsqueda será más eficiente y acertada.

1.3 Descriptores de bajo nivel.

Los descriptores de bajo nivel LLDs (Low Level Descriptors), son la capa base del estándar MPEG-7. Estos consisten en la colección de características de audio simples y de baja complejidad, que puedan ser usados para caracterizar cualquier tipo de sonido.

Estos descriptores son de importancia en general para describir el archivo de audio, a su vez estos también pueden ser extraídos del archivo automáticamente y representar las variaciones en tiempo y frecuencia. En base a estos mismos se pueden encontrar similitudes entre diferentes archivos, y esto nos provee de una base de clasificación e identificación del contenido de audio.

Descriptores básicos: los cuales están basados en el dominio del tiempo, es decir en la variación temporal de la señal, como en la forma de onda y la variación de potencia.

Descriptores básicos de espectro: estos describen el espectro en términos del envolvente, centroide, propagación y llanos.

Descriptores de parámetros de señal: que describen frecuencias fundamentales de la señal de audio así como la harmonía de la señal, como el pitch y la intensidad de la señal

Descriptores temporales de timbre: registro de tiempo del ataque y centroide temporal.

Representaciones espectrales básicas: existen dos, para representar los de dimensión baja y dimensión alta. Estos descriptores son usados principalmente con las herramientas de clasificación del sonido y el indexado, aunque pueden ser utilizados para otras aplicaciones de igual manera.

1.4 Esquemas de descripción DSs (Description Schemes) de MPEG7.

Los DSs son escritos en XML en los cuales se describen las instrucciones a seguir para la adquisición del descriptor. Estos especifican los tipos de descriptores que pueden ser usados en una búsqueda dada y la relación entre estos descriptores o entre otros. Además son definidos usando el lenguaje de definición de descripción (DDL), que está basado en el lenguaje esquemático XML. Estos están instanciados como documentos o fuentes. El resultado de los descriptores puede ser expresado en forma textual para ser leídos por el usuario en la búsqueda de información, o ser comprimidos y binarizados para su transmisión o almacenamiento.

Hay 5 conjuntos de herramientas de descripción que robustamente se pueden corresponder en áreas de aplicación que están integradas en el estándar: la firma del audio, timbres de los instrumentos musicales, descripción de la melodía, reconocimiento general del sonido e indexado, y contenido hablado.

1.5 Lenguaje de definición de descripción de MPEG-7 (DDL).

El DDL define reglas sistemáticas para expresar y combinar DSs y descriptores. Que a su vez esto permite que el usuario cree sus propios DSs y sus descriptores, el DDL no es lenguaje de modelación sino un lenguaje de esquema. En el cual es posible expresar relaciones conceptuales de espacio, tiempo y estructurales entre los elementos que hay en un DS, y entre DSs. El cual provee de un modelo más rico para los enlaces y referencias entre uno o más descripciones y los datos que son descritos. También, la plataforma y la aplicación son independientes de la lectura de la maquina o el humano que los esté usando.
El objeto de un esquema es que defina la clase de documentos XML. Esto es logrado por la especificación particular que construye la estructura y el contenido de los documentos. Las posibles restricciones incluyen: elementos y su contenido, atributos y sus valores, cardinalidades y sus tipos de datos.

1.6 Búsquedas Musicales.

Al día de hoy podemos ver una gran cantidad de aplicaciones de búsqueda de música sin tomar en cuenta las aplicaciones clásicas de mercados virtuales tales como iTunes. Muchas de estas se basan en la obtención de pequeños fragmentos de melodías por diversas fuentes tales como el nombre del artista, disco en la que aparece la canción, un silbido, un tarareo, un aplauso o una muestra de la música que se quiere identificar.

Las búsquedas basadas en obtención de ejemplos Query By Example (QBE) han sido utilizadas e investigadas para diversos tipos de información que se almacenan en bases de datos. Tales como datos personales, búsquedas en internet, audio, imágenes o video. Cada una de estas búsquedas están ligadas a los tipos de datos que son requeridos por los usuarios. Para este trabajo nos centraremos en las búsquedas de audio específicamente en música.

Si analizamos algunos de estos tipos de búsquedas cada una de estas se basa en distintos descriptores de audio. Una aplicación existente en el mercado es la llamada TrackID [24] en cuyo caso se basa en una búsqueda por muestra de audio conocida como Query by Example Music Retrieval (QEMR). Estas consisten en grabar un fragmento de la canción que se desea reconocer, esta herramienta de búsqueda será capaz de identificarla; así pues, TrackID es capaz de hacer una comparación en su base de datos para posteriormente proporcionar los datos solicitados.

Figura 1.1 Aplicación de TrackID

En el QEMR se analizan las canciones en base a descriptores acústicos y psicoacústicos, los cuales permiten separar características similares de las canciones en parámetros globales y locales [5]. Estas características solo se basan en modelos de música que utilizan el espectro de frecuencias presentes en esta. Algunos de los descriptores involucrados para el desarrollo de estas aplicaciones utilizan el timbre [9] o bancos de filtros con una distribución de Mel [10].

Una de las características que tiene el QEMR respecto al QBH es que las canciones analizadas son exactamente iguales. Es decir que si analizamos y comparamos una canción la estaremos comparando con ella misma, lo que conlleva a una identificación con menos complicaciones de las que se tendrían con cualquier otro tipo de búsqueda. En un sistema como el TrackID no puede identificar por otra fuente de muestra que no sea la misma canción original, en caso contrario no tendrá éxito la búsqueda con lo que conlleva a realizar otro tipo de aplicaciones que utilicen otra fuente de audio.

Otra herramienta en la que actualmente existen diversas aplicaciones es la búsqueda por tarareo, por silbido o por canto. Estas 3 pueden ser englobadas en una sola, ya que en esencia pueden ser analizadas por los mismos descriptores. El pitch que es una herramienta básica para la obtención de las frecuencias fundamentales presentes en la canción o en la voz.

Paginas de internet como Midomi [6] tienen el servicio de identificación de canciones por tarareo o por canto, teniendo un desempeño medianamente bueno. Este servicio hace comparativos en tiempo real con voces previamente grabadas de otros usuarios, trabajando como un sistema monofónico y actualizando su base de datos constantemente, relaciona los datos similares con las de su base. Cabe mencionar que en ocasiones requiere de ayuda del usuario para relacionar el tarareo con su canción, para así ampliar su base de datos. Esto quiere decir que emplea otros métodos para clasificar la canción y no necesariamente analiza el fragmento que ha tarareado o cantado el usuario.

En los últimos años el QBH es un tema de investigación en la que se han dado muchas propuestas de análisis y de modelación de música. La gran mayoría de estos se basan en la obtención de las frecuencias fundamentales que componen la canción o la voz según el tipo de procesamiento que se proponga. En la mayoría de los casos son para sistemas monofónicos, es decir solo se hacen comparaciones voz con voz, voz con instrumentos, voz con datos MIDI en donde solamente se involucra un instrumento a la vez.

En [11] se obtienen los datos almacenados en un archivo MIDI, los cuales cuentan con la información de toda la canción y básicamente la melodía. En este trabajo de tesis como en la mayoría el pitch de la voz es usado para obtener las frecuencias involucradas, así como su duración y su forma, para después ser comparadas con una elección jerárquica frente a las melodías que existen en el archivo MIDI. En otros trabajos se analiza de manera similar al utilizar los archivos MIDI como base de datos, como en [12] la voz es puesta en el análisis clásico, determinando su pitch; pero, procediendo a una modificación para transformar sus datos a un formato MIDI, para hacer su comparativo posterior con los datos obtenidos y los de su base de datos con métodos similares a los utilizados en nuestro sistema.

Este trabajo de tesis como antes mencionado se dividió en dos propuestas: monofónico y polifónico. Y se decidió de esta manera debido que a pesar de que existen diversos métodos para el primer sistema, para el segundo existen menores resultados expuestos.

Pero dado que para poder comprender mejor el tema aquí exhibido, debemos tomar en cuenta ambas alternativas. Dado que el segundo sistema resulta de mayor complejidad pero en consecuencia se podría encontrar con una solución a diversos problemas que se analizan hoy en día. Las propuestas realizadas pueden derivarse en diferentes descriptores de música.

Uno de los trabajos consultados para sistemas polifónicos [25] trata de separar la melodía principal de la canción. Pero solamente toman en cuenta que la melodía principal es ejecutada por el cantante y que esta se encuentra en el centro de un canal estéreo, dando como resultado un aislamiento de la voz principal para ser analizada y comparada posteriormente. Dejando de lado que mucha música tiene partes características ejecutadas por otros instrumentos y no solamente por la voz, así teniendo un desempeño no muy favorable. Aunque en éste se sugiere la reducción del descriptor a una sola octava que es una característica que se utiliza para nuestro análisis.

1.7 Conclusiones.

Para poder modelar la música necesitamos comprender como está compuesta. Así podremos escoger los parámetros mas adecuados y por ende tendremos un mejor resultado. Si este modelo funciona adecuadamente tanto para una canción (sonido polifónico), como para la voz (sonido monofónico), tendremos la certeza de que será posible tener una identificación buena. Sin embargo existen diversos problemas que limitan la búsqueda y por ende el resultado. Problemas como el ruido existente en la canción, en la voz o la misma interpretación del usuario, son problemas que se tratarán en este trabajo.

Capitulo 2.

Fundamentos para el Análisis de la Música y su Representación.

2. Fundamentos para el Análisis de la Música y su Representación.

Para poder desarrollar un sistema adecuado, se requirió de una previa investigación sobre las principales herramientas para el procesamiento de la señal de voz y el tratado de la señal a enviarse a través de la red celular o en internet. En este capitulo se verán de una forma sintetizada estas herramientas y procesos que se involucraron para el desarrollo de la aplicación expuesta.

2.1 Procesamiento Digital de señales.

El procesamiento digital de señales es un área de la ciencia y la ingeniería que ha crecido en cuanto a desarrollo los últimos 30 años. Por lo cual, el rápido avance de esta área ha ayudado al crecimiento de tecnologías en la computación con la fabricación de mejores circuitos integrados. Así dando a muchísimas áreas la ingeniería facilidades en el manejo de la información, al hacer un mejor utilización de los recursos; entre ellas las telecomunicaciones al dar un mejor uso de los recursos de la comunicación como es en el canal y en el almacenamiento de la información.

¿Que es una señal? [14] una señal esta definida como cualquier cantidad física que varia con el tiempo.

Matemáticamente se puede describir como una función con una o más variables independientes como en (2.1).

$$s1(t) = 3t \tag{2.1}$$

Las señales pueden ser encontradas en diferentes casos y áreas de la vida cotidiana, desde las micro-señales eléctricas que se efectúan a través de nuestro sistema nervioso, nuestra voz al hablar al producir perturbaciones en el aire, la electricidad que llega a nuestros hogares, y dispositivos móviles como teléfonos celulares, PDA, laptops etc.

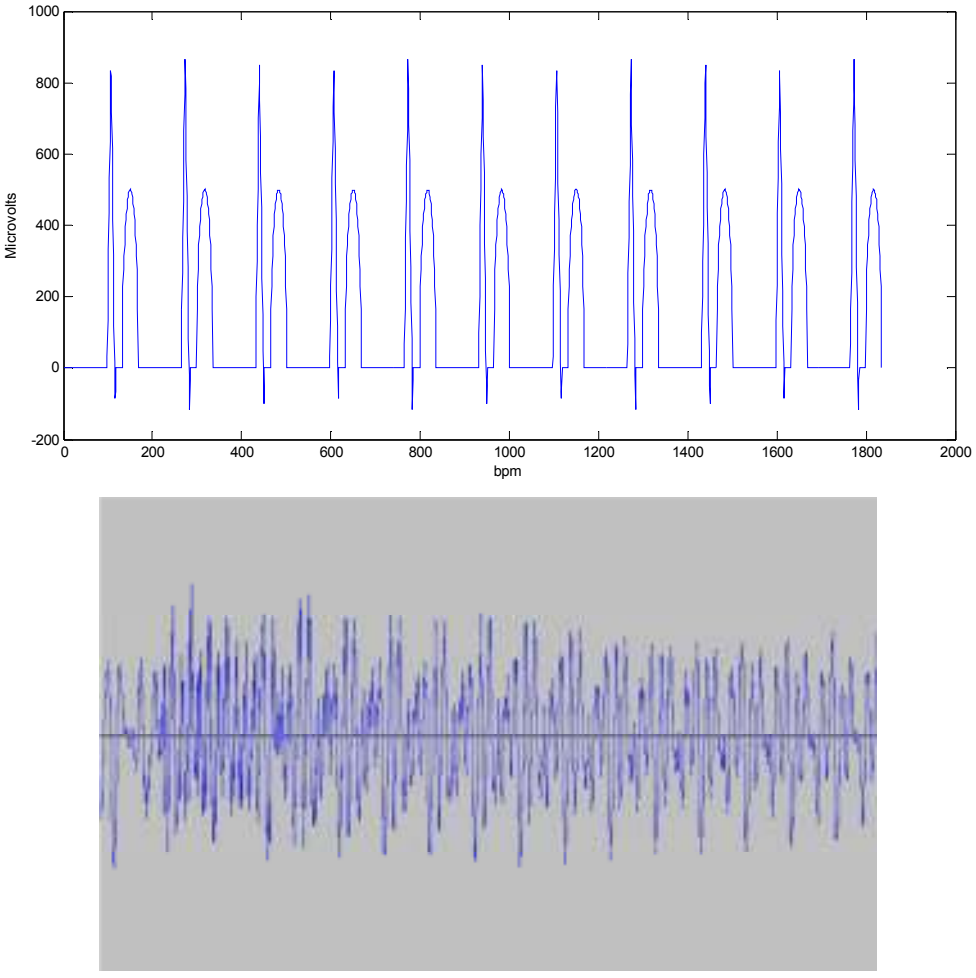


Figura 2.1. Señal de electrocardiograma y señal de voz

En el caso particular de una señal de voz la cual lleva información especifica, como un mensaje o un código o un sonido (nota musical), la señal esta claramente definida por parámetros característicos los cuales se encuentran en otras señales como son amplitud, frecuencias y cambios de fases que esta pueda tener.

Estos parámetros básicos son la base del procesamiento de una señal. Naturalmente aun estamos refiriéndonos a una señal analógica, la cual podemos convertirla en una señal digital siendo ésta una nueva parte del procesamiento de las señales, ahora trabajando con señales del tipo discreto que es un tema que tratará en este capitulo.

Otra señal que es objeto de estudio será una señal de audio específicamente música. A diferencia de una señal de voz (30Hz-3000Hz) el audio en una pieza musical puede estar presente dentro del todo el umbral de audición humana (20Hz-22010Hz).

2.2 Señales discretas en el tiempo

Una señal discreta es una sucesión de números indexados ya sean reales o complejos. Así que una señal discreta en el tiempo es una función de valor entero n, aunque en esta variable n no en todos los casos es necesariamente el tiempo.

Las señales discretas basadas en el tiempo regularmente son derivadas de una señal continua que ha sido muestreada, es decir pudo haber sido tomada por ejemplo de una señal de voz o de audio entre otras, que fue insertada en un convertidor analógico-digital (figura 2.2), así teniendo una señal discreta en el tiempo.

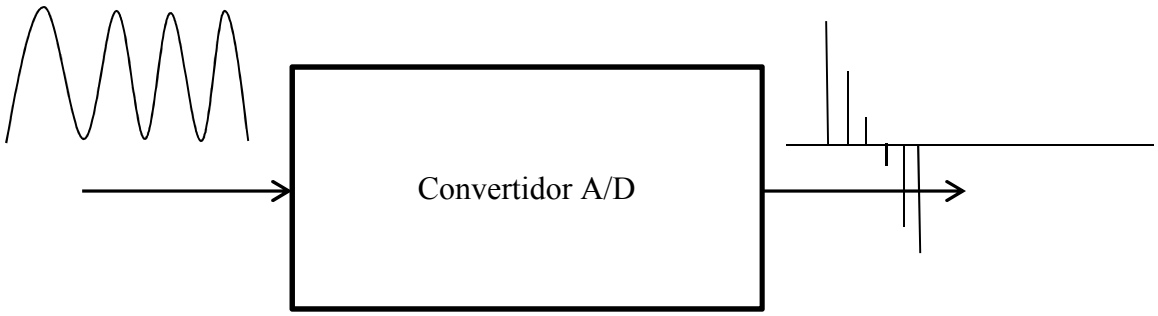


Figura 2.2. Señal digitalizada

Hay muchas razones por las cuales trabajar con señales discretas en lugar de trabajar con la señal analógica. Una de las razones es que un sistema digital programable permite mucha flexibilidad a la hora de reconfigurar la señal digital procesada simplemente con el cambio del programa, reconfigurar una señal analógica usualmente requiere de la modificación del hardware del que esta procede. Otra importante ventaja es que una señal digital puede ser fácilmente almacenada en medios magnéticos y sin perder la fidelidad de éste, y en algunos

casos la implementación de un sistema digital puede llegar a ser mas barato que su contraparte analógica.

Una señal analógica puede ser procesada o representada por diferentes medios ya sean por medio de filtros, mezcladores, sintetizadores, etc. En la contraparte digital de igual manera se pueden realizar todo tipo de análisis y modificaciones a la señal pero estos procesos por obvias razones deben realizarse digitalmente es decir, analizar las muestras de la señal en cuestión.

Las transformaciones que puede sufrir una señal digital pueden ser de gran ayuda para determinar parámetros característicos de esta, ya sea trabajando con ésta en el dominio del tiempo o en el dominio de alguna transformada, (transformada de Fourier, transformada Z, transformada coseno discreta, etc.). De aquí podemos visualizar como es que se pueden obtener los descriptores para la información multimedia enfocada principalmente a audio y video, hablando independientemente del proceso que sigue una señal de estos tipos al ser transmitida por cualquier medio de comunicación.

2.3 Transformada de Fourier

La transformada de Fourier es una herramienta matemática que se podría considerar la más importante para el procesamiento de audio. Ya que asigna una función dependiente del tiempo a una función dependiente de la frecuencia, y esta revela el espectro de frecuencia que componen la función original; es decir, la transformada de Fourier nos muestra dos lados de la misma información.

Cuando una función f depende del tiempo ésta solo muestra la información en el tiempo ocultando la información que podemos obtener en frecuencia. Un ejemplo claro de información que se puede obtener es cuando en una grabación de un instrumento una nota es producida por una guitarra, el ataque de la guitarra se va a ver reflejado a menudo en la información obtenida en el tiempo, pero lo que no muestra la información es que notas han sido ejecutadas.

La transformada de Fourier de una función $\hat{f}$ esconde toda aquella información que se mostraba en el tiempo y en cambio muestra toda la información acerca de las frecuencias. En este caso haciendo referencia al ejemplo de la guitarra solo observaríamos aquellas notas que han sido ejecutadas pero no tendríamos idea de en que tiempo estas fueron tocadas.

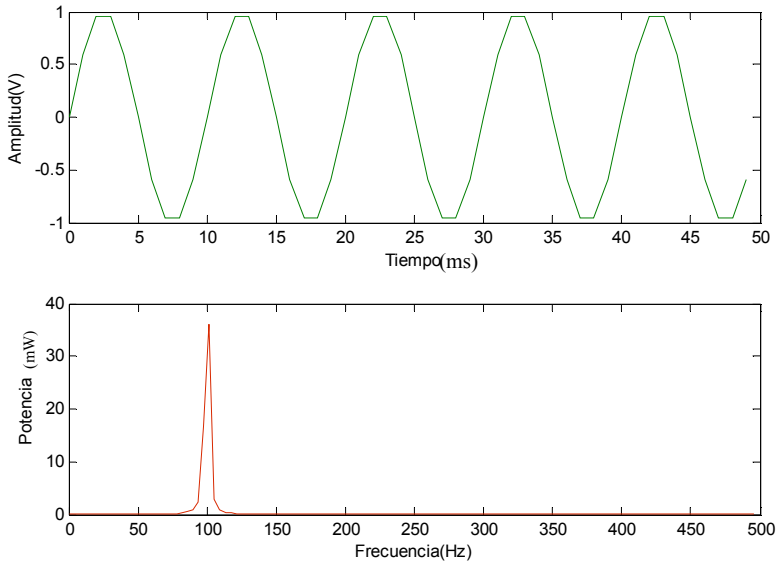


Figura 2.3. Representación en tiempo y en frecuencia

En forma continua es como hemos tratado la transformada de Fourier. Como es sabido, ésta tiene su similar en forma discreta. Tales señales poseen una representación de Fourier que puede ser considerada como una superposición ponderada de señales senoidales de frecuencias diferentes, el cálculo de la transformada de Fourier consiste en la evaluación de integrales o sumas infinitas.

Para realizar el cálculo en la práctica se debe de realizar aproximaciones, para obtener la transformada por medio de sumas infinitas, esto puede realizarse eficientemente por la conocida transformada rápida de Fourier (FFT).

La transformada de Fourier en su forma continua puede ser definida como en (2.2),

$$x(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t).\, e^{-j\omega t} dt \tag{2.2}$$

En la práctica en el procesamiento de señales y otras áreas se utiliza t y ω, que están designadas al tiempo y frecuencia respectivamente. La expresión que nos lleva del dominio de la frecuencia al dominio del tiempo se encuentra descrita en (2.3).

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(\omega).\, e^{j\omega t}\, d\omega \tag{2.3}$$

La idea base de la representación de Fourier, es la de representar una señal como una superposición ponderada de las funciones de frecuencia elementales independientes. Cada

una de las ponderaciones expresa la extensión que le corresponde, su función elemental que contribuye con la señal original, para revelar ciertos aspectos de la señal.

2.3.1 Transformada discreta de Fourier

El análisis de frecuencias de señales discretas en el tiempo, es el más utilizado en procesamiento digital de una señal; para trabajar en el análisis de frecuencias en tiempo discreto de una señal *x(n)* tenemos que transformar la secuencia del dominio del tiempo al dominio de la frecuencia en una representación equivalente como hemos visto anteriormente.

Al tratar de realizar una transformada de Fourier de una señal se requiere realizar la evaluación de integrales o de sumatorias infinitas, que es en general una tarea no viable. Por lo cual la forma en que se emplea esta transformación es considerando la representación de una secuencia *x(n)* de su espectro $X(\omega)$, tal representación en el dominio de la frecuencia nos lleva a la trasformada discreta de Fourier (DFT) que es una herramienta muy útil para el análisis de frecuencia de señales en tiempo discreto.

Con el muestreo en el dominio de la frecuencia de una secuencia finita aperiódica *x(n)*, en general las muestras separadas en frecuencia

$$X\left(\frac{2\pi k}{N}\right), k = 1,2,3, \dots, (N-1) \tag{2.4}$$

No representan únicamente la secuencia original *x(n),* cuando *x(n)* tiene duración infinita. Pero en cambio la frecuencia de las muestras en (2.4), corresponden a una secuencia periódica $x_p(n)$ que tiene periodo *N,* donde $x_p(n)$ es un alias de *x(n)* como se indica en (2.5),

$$x_p = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{2.5}$$

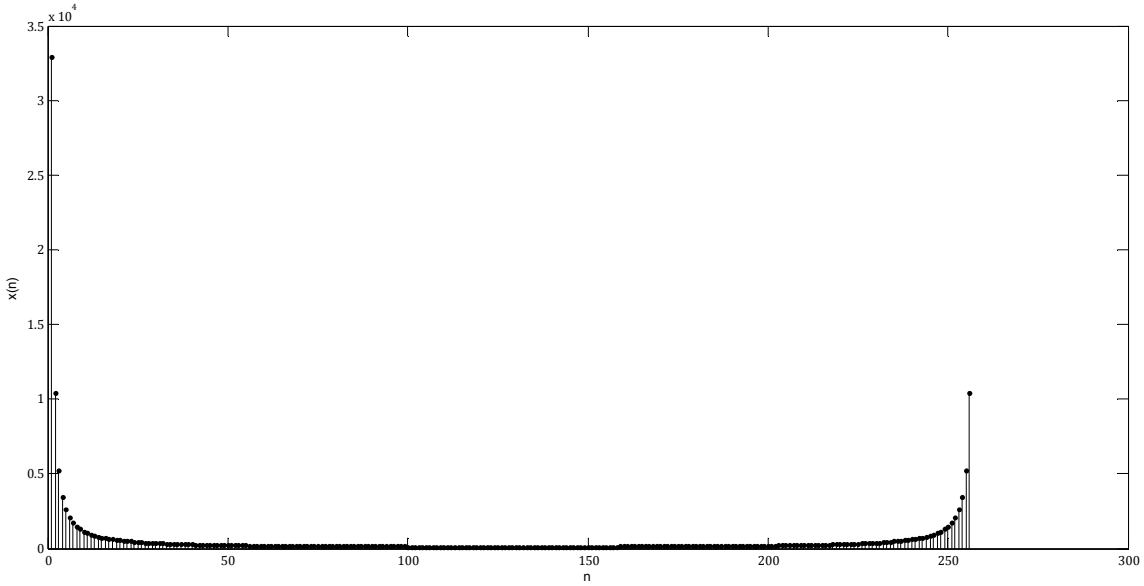


Figura 2.4. Transformada Discreta de Fourier

Como la frecuencia de las muestras es obtenida evaluando la transformada de Fourier $X(\omega)$ de un set de N (igualmente espaciadas) de frecuencias discretas, en la relación (2.6), es llamada transformada discreta de Fourier (DFT) de *x(n)*,

$$X(k) = \sum_{k=0}^{N-1} X(k) e^{-j2\pi kn/N}$$

$$k = 1,2,3, \dots, N-1 \qquad (2.6)$$

Y para recobrar la secuencia de *x(n)* de las muestras de frecuencia, tenemos la expresión en (2.7).

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \qquad (2.7)$$

En la practica es utilizada la llamada transformada rápida de Fourier, matemáticamente la DFT es una aproximación de la transformada de Fourier, donde la DFT de tamaño N es una asignación lineal de $\mathbb{C}^N \rightarrow \mathbb{C}^N$ dado por una matriz de N *x* N.

$$DFT_N = \left(\Omega_N^{kj}\right)_{0\le k,j\le N} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \Omega_N & \dots & \Omega_N^{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \Omega_N^{(N-1)} & \dots & \Omega_N^{(N-1)(N-1)} \end{pmatrix}$$

(2.8)

Donde $\Omega_N = e^{-2\pi i/N}$, para un vector de entrada $v = \left(v(0), v(1), \dots, v(N-1)\right)^T \in \mathbb{C}^N$, la evaluación de la DFT esta dado por el producto matriz-vector $\hat{v} = DFT_N . v$, donde $\hat{v} = (\hat{v}(0), \hat{v}(1), \dots, \hat{v}(N-1))^T \in \mathbb{C}^N$ que denota la salida del vector. Se puede notar el cálculo directo del producto de la matriz-vector requiere $O(N^2)$ sumas y multiplicaciones. Para la mayoría de las aplicaciones esto es demasiado lento en muchos casos se tiene que lidiar con largos de $N \gg 10^5$. El punto importante es que existe un algoritmo eficiente, que es llamado trasformada rápida de Fourier (FFT), el cual solo requiere $O(NlogN)$ sumas y multiplicaciones. La idea de la FFT fue originalmente ideada por Gauss, pero fue redescubierta por Coley and Tukey basándose en la factorización de la matriz de la DFT consistiendo de $O(logN)$ escasas matrices cada una de las cuales se puede evaluar con $O(N)$operaciones. Véase [16].

2.4 Función de autocorrelación.

La correlación es una operación matemática la cual permite cuantificar el grado de similitud entre dos señales, aunque en apariencia no haya señales de eso. Puede ser muy parecida a la convolución, sin embargo las aplicaciones y las propiedades de estas son utilizadas para diferentes propósitos.

Es evidente que la noción de distancia esta unido al parecido y por esa razón el producto escalar será una parte importante en lo que respecta a la evaluación de la distancia entre las señales dadas las señales *x(n)* y *y(n)*, la distancia entre ambas estará definida por (2.9) donde se omite la división entre los sumandos por sencillez y no altera las conclusiones.

$$D_{xy} = \sum_{n=0}^{N-1} |x(n) - y(n)|^2 = (x-y)^H (x-y) = \|x\|^2 + \|y\|^2 - x^H y - y^H x$$

(2.9)

Como puede observarse en la distancia entre los 2 términos es primeramente las energías de las señales comparadas, a mayor energía mayor distancia. En segundo término es el esperado a mayor parecido menor la distancia.

Lo anteriormente expuesto en cuanto a la distancia y el parecido puede ser extendido a ser empleado a una señal consigo misma. El parecido es la unidad (100%) y su auto-producto

escalar es la norma. Lo interesante de esto es calcular el parecido de una señal con una versión desplazada de ella misma es evidente que de este modo se puede extender la noción de auto-parecido, o mejor conocido como autocorrelación de una señal. Esta función obtenida para diversos desplazamientos o lags se denomina función de autocorrelación en la expresión presentada en (2.10) sirve para calcular el parecido de Q - 1 lags sucesivos.

$$r(m) = \frac{1}{M}\sum_{n} x*(n-m)x(n) = \frac{1}{M}\sum_{n} x*(n)x(n+m) \tag{2.10}$$

Donde se ha divido por el número de términos involucrados en la sumatoria. El valor que es calculado, en la autocorrelación de m indicara el parecido entre *x(n) y x(n-m)*.

Esta función de autocorrelación puede ser muy útil ya que nos permite encontrar patrones repetitivos dentro de una señal. Es decir la periodicidad de la señal, aunque esta presente ruido y con esto detectar la frecuencia fundamental de la misma. De esto se deriva que podemos encontrar el *pitch* de una señal de audio; el pitch es un elemento muy importante para la elaboración de este trabajo y se vera mas a detalle en el capitulo 3.

2.5 Filtros Digitales.

Un filtro es un sistema cuya finalidad es modificar el espectro de la señal de entrada ya sea este continuo o discreto, en el audio estos filtros son utilizados para modificar la forma de onda en un modo específico para alterar las propiedades de su espectro. Siendo unos de los mas referidos para este trabajo aquellos que atenúan ciertas partes a un determinado ancho de banda; todos los filtros satisfacen cierta linealidad, estabilidad y varianza, la cual puede ser expresada mediante una convolución, una trasformada discreta de esta señal *s*. La máxima tasa de muestreo de un filtro digital es igual a la mitad de la frecuencia de muestreo del teorema de Shannon.

En general, los filtros digitales se pueden clasificar en dos categorías que son: filtros con respuesta al impulso infinita (IIR) y los filtros con respuesta al impulso finita (FIR) también conocidos como filtros transversales. En el caso de los filtros IIR se aprovechan métodos para diseñar filtros analógicos, y estos se transforman en filtros digitales usando diferentes transformaciones (lineal, bilineal, etc.). Estos filtros son inestables ya que la transformada rápida tiene tanto polos como ceros, y en el caso de los filtros FIR estos siempre son estables. Hay distintos caminos para diseñar este tipo de filtros pero este va a depender de la circunstancia en la que sea empleado.

Algunas ventajas de un filtro digital sobre su símil analógico son: la respuesta en frecuencia es mas cercana a la ideal; no requieren de sintonización, pueden multiplexarse para el procesamiento de mas señales; su rediseño es sencillo ya que solo implica cambiar los

coeficientes de este, sus componentes son independientes de la frecuencia de operación del filtro; tienen un alto grado de integración por lo que se tiene mayor confiabilidad.

Para el diseño de filtros IIR, se requiere de la transformación de un filtro analógico a un digital que satisfaga las necesidades requeridas. Este tipo de diseño es directo, ya que se aplican métodos de diseño ya derivados de los filtros analógicos y así en ocasiones se requerirán de un filtro analógico usando un digital.

Hay una gran variedad de filtros analógicos que pueden ser usados para el diseño de filtros digitales, como son el filtro Butterworth, Chevyshev, Elípticos, Bessel, entre otros.

La transformada de Fourier de la señal se conoce como la respuesta en frecuencia de un filtro que puede referirse a importantes características sobre la selectividad de un filtro subyacente. La respuesta en frecuencia puede ser utilizada para especificar ciertas características de un filtro a como lo requiera la aplicación. Es decir como ejemplo podemos utilizar reducir la frecuencia de muestreo de una señal de audio de 44.1KHz a 4KHz, necesitamos un filtro antialiasing que remueva todas las frecuencias por arriba de 2000Hz y retenga aquellas debajo de esta.

2.5.1 Filtros de convolución

La convolución de dos señales *x* y *y* discretas es una clase de multiplicación que produce una señal del tipo $x * y$.

La convolución puede ser definida para diversos espacios. En particular una convolución circular puede ser definida por funciones periódicas, y una convolución discreta se puede definir para las funciones con números enteros. Estas generalidades de la convolución tienen aplicación en el campo del análisis numérico, el algebra lineal numérica y el diseño e implementación de filtros FIR en el procesamiento de la señal.

Un filtro matemáticamente puede ser representado de una E→S que va ha transformar de una señal $x \in E$ a una nueva señal $y \in S$ donde E y S son espacios que se adecuan a esa señal. Para nuestro caso podemos considerar como prioritarios al caso de las señales discretas si es que no se indica lo contrario, entonces podríamos considerar los espacios de las señales en $E = \ell^2(\mathbb{Z})$ y $S = \ell^2(\mathbb{Z})$ [15].

Así una clase de filtros puede ser descrita con convoluciones, que es un concepto que es una herramienta matemática para el análisis de una señal. En este sentido representa una cantidad de superposición entre *x* y una versión superpuesta y trasladada de *y*. La convolución de *x* y *y* en una posición $n \epsilon \mathbb{Z}$ esta definido por.

$$(x * y)(n) = \sum_{k \epsilon \mathbb{Z}} x(k)y(n-k) \tag{2.11}$$

Nótese que la sumatoria en (2.11) puede ser infinita por lo general en *x* y *y*. Pero la convolución de *x*y* existe bajo las mismas condiciones en las señales digitales. La forma mas fácil de que se cumpla la condición es que *x* y *y* tengan un número finito de entradas diferentes de cero. Para ser mas específicos se define la longitud de $\ell(x)$ tal que x puede ser.

$$\ell(x) := 1 + \max\{n|x(n) \neq 0\} - \min\{n|x(n) \neq 0\}. \tag{2.12}$$

Entonces para dos señales *x* y *y,* de longitud positiva y finita, la convolución de *x* * *y* existe y $\ell(x * y) = \ell(x) + \ell(y) - 1$. Otra condición conocida como desigualdad de Young, dice si $x \in \ell^1(\mathbb{Z})$ y $y \in \ell^p(Z)$, entonces $|(x * y)(n)| < \infty$ para toda $n \in \mathbb{Z}$ y $||x * y||_p \leq ||x||_1 \cdot ||y||_p$.

Siendo de importancia ahora la convolución en el contexto de un filtro se deriva del siguiente hecho: todos los filtros lineales $T : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ que son invariantes en el tiempo y satisface una condición de continuidad puede ser expresada como un filtro de convolución. En efecto la definición $h := T(\delta)$, se puede demostrar que $T = C_h$.

Un filtro $T = C_h$ es llamado filtro FIR si *h* tiene longitud finita y es un filtro IIR en el caso contrario. Si se encuentra que $h \neq 0$ es su respuesta al impulso de algún filtro FIR, entonces $\ell(h)$ es también llamada longitud y $\ell(h) - 1$ el orden del filtro FIR. Por otro lado un filtro es llamado causal si $h(n) = 0$ para $n < 0$. La propiedad de la causalidad se convierte en importante en el contexto en las aplicaciones de procesamiento de la señal en tiempo real, donde no se puede observarlos futuros valores de la señal. Por ejemplo si se filtra una señal *x* con un filtro causal FIR $T = C_h$ de orden N descrito por (2.13)

$$T(x)(n) = \sum_{\ell=0}^{N} h(\ell)x(n-\ell), \tag{2.13}$$

Para coeficientes del filtro $h(0), \ldots, h(N)$ con $h(0) \neq 0$ y $h(N) \neq 0$. La salida de la señal *T(x)* en el punto *n* solo depende de las muestras pasadas de *x*(*n* - 1),. . ., *x*(*N* – n) y de las muestras del presente de *x*(*n*) de la señal de entrada *x*.

2.5.2 Diseño y especificaciones de los filtros.

En el proceso de diseño de un filtro se empieza por las especificaciones del mismo las cuales nos darán las limitaciones en magnitud y/o fase de la frecuencia de respuesta, limitaciones sobre la muestra o unidad de respuesta del paso del filtro, así como el tipo de filtro (FIR o IIR), y el orden del filtro. Una vez que estas especificaciones del filtro se han definido el siguiente paso es encontrar un conjunto de coeficientes en el filtro que produzcan un filtro adecuado. Cuando este terminado el diseño de un filtro por ultimo surge su implementación en el sistema ya sea por medio de hardware o software. De ser

necesario se deben cuantizar los coeficientes del filtro y escoger la estructura adecuada para el filtro.

Si suponemos que queremos diseñar un ideal filtro pasa-bajas con una frecuencia de corte en ω_0. En la teoría podríamos obtener dicho filtro simplemente con invertir la frecuencia de respuesta como se observa en la figura 2.5.

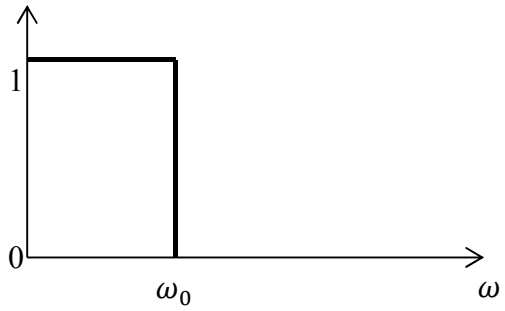


Figura 2.5. Filtro paso-bajas ideal

Si denotamos una función sinc como sigue:

$$sinc(t) = \begin{cases} \frac{sin\pi t}{\pi t} \; para \; t \neq 0, \\ 1 \;\; para \;\; t = 0. \end{cases} \tag{2.14}$$

Y luego mediante un calculo encontramos que los coeficientes para un filtro ideal esta dado por $h(n) = 2\omega_0 sinc(2\omega_0 n) \; para \; n \in \mathbb{Z}$. Sin embargo este filtro tiene muchos inconvenientes dado que tiene un número infinito de coeficientes del filtro diferentes de cero y no es causal ni estable.

En la actualidad realizar un filtro ideal con estas características no es posible, por lo que se tienen que trabajar con aproximaciones que pueden tener fenómenos como los que siguen: la frecuencia de respuesta *H* de un filtro que es realizable presenta rizos en la banda de paso y la banda de parada.

Además de que *H* no tiene ningún punto de corte brusco en la banda de paso y en la banda de no paso véase figura 2.6 tampoco H puede caer de la unidad a cero abruptamente.

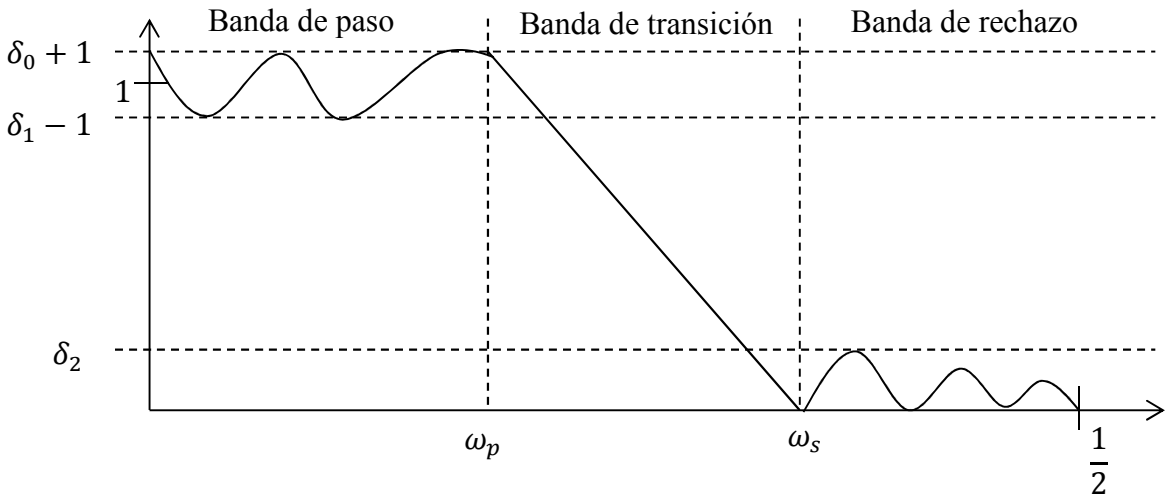


Figura 2.6 Características en magnitud de un filtro realizable.

En las aplicaciones algunas cosas pueden ser toleradas como son los rizos en la banda de paso y en la banda de no paso. La transición de la respuesta en frecuencia de la banda de paso a la banda de no paso figura (2.6), es llamada banda de transición del filtro. El límite de la banda de frecuencia ω_p define el límite de la banda de paso, mientras que el parámetro ω_s va a definir el comienzo de la banda de rechazo. La diferencia que existe entre estas dos $\omega_s - \omega_p$ es conocida como ancho de transición, de manera similar en la parte de la banda de paso se conoce como ancho de banda del filtro.

Si hay rizos en la banda de paso del filtro, la máxima desviación de los rizos por arriba y por debajo de 1 son denotados por δ_0 y δ_1 respectivamente. Si la magnitud de $|H|$ varia dentro del intervalo de $[1 - \delta_1, 1 + \delta_0]$. La máxima magnitud de los rizos en la banda de rechazo del filtro esta denotada por δ_2, y de la misma manera estas características pueden ser definidas para un filtro pasa-altas.

En el caso de un filtro pasa-banda tiene dos bandas de rechazo, así como también tiene dos bandas de transición a la izquierda y a la derecha. En el diseño habitual de un filtro, un filtro *h* es construido de forma que se encuentra dentro de sus especificaciones y clase dentro de lo que es requerido. Y del grado en que *H* se aproxime a las especificaciones del filtro depende del orden del filtro.

Como se había mencionado anteriormente existen otras especificaciones referidas a la respuesta en fase del filtro. Si un filtro *h* es la función de muestra de frecuencia elemental. $n \mapsto e_\omega(n) = e^{2\pi i \omega n}$:

$$(h * e_\omega)(n) = \sum_{k \in \mathbb{Z}} h(k) e^{2\pi i \omega (n-k)} = H(\omega) e_\omega(n) = |H(\omega)| e^{2\pi i (\omega n + \phi_h(\omega))} \tag{2.15}$$

Se puede notar que la fase induce a un desplazamiento en el tiempo en función de la frecuencia elemental, que generalmente depende de ω. Para una señal de entrada en general tal retraso en todos los componentes de frecuencia conduce a un retardo global en el proceso de filtrado que no es considerado como una distorsión lo cual lleva a ciertas definiciones.

Si $\phi_h = c\omega$ modulo 1 para algún $c \in \mathbb{R}$, el filtro h se dice que es de fase lineal, la función $\tau_h: [1,0] \rightarrow \mathbb{R}$ esta definida por

$$\tau_h(\omega) = \frac{d\phi_h}{d\omega}(\omega) \tag{2.16}$$

Es llamado retraso de grupo de un filtro h donde las discontinuidades en la fase son a consecuencia de las ambigüedades que no se consideran. El valor de τ_h puede ser interpretado como el tiempo de retraso que un componente de la señal de frecuencia ω sufre en el proceso de filtrado.

2.6 Representaciones musicales.

Como se menciona en este capitulo la música puede ser representada por signos los cuales pueden ser interpretados por nosotros como notas musicales y en conjunto música. Existen reglas que fueron establecidas para poder ser interpretadas, como por ejemplo cualquier lengua cuenta con sus propios signos, fonemas, gramática etc. En el caso de la música es muy similar cada signo y/o signos representaran una letra, palabra o frase. Hoy en día, las librerías musicales tienen gran diversidad de información como texto, video, audio, etcétera. Lo cual representa un problema dado que la música esta representada en muchos formatos.

En este tema nos centraremos en entender a grandes rasgos la representación musical, la representación de audio por forma de onda utilizado para la grabación de CDs y la representación de interfaz digital de instrumentos musicales (MIDI). Ya que estas representaciones son en las que se basa este trabajo de tesis para el procesamiento de la música.

2.6.1 Representación por signos (Notación Musical).

Como es conocido, las melodías normalmente son representadas por signos. En la música clásica occidental están plasmados sobre una partitura. Esta representación musical permite a un músico tocar la melodía por medio de instrucciones en la figura 2.7 podemos visualizar una pieza musical.

Esta representación esta definida por objetos que quieren decir entonación de cada nota (pitch), el inicio de la melodía, la duración de la nota, la dinámica, la ejecución y los tiempos en los cuales no se debe tocar ninguna nota (silencios). El tiempo de la música es especificado por notas textuales sobre la partitura como *allegro, moderato, andante*, etc. Esto se aplica regularmente para el tiempo local de la melodía o por decir de alguna manera frase. Del mismo modo la intensidad y la dinámica son referidas a notas como *forte, crescendo o diminuendo.* Una partitura describe la música de cierta manera, como es que la pieza musical debe ser ejecutada por el músico en ciertos momentos y con ciertos efectos como la acentuación de las notas. Aunque en la interpretación de una melodía siempre es libre, lo que conlleva a tener múltiples interpretaciones de una misma canción las cuales tendrán variaciones en tiempo; en la articulación y en la dinámica en que es interpretada la música; incluso, puede haber variaciones en las notas ya que existen arreglos como los arpegios o las notas gracia que siguen con la armonía de la música. Sin embargo puede que no estén especificadas en la partitura de la pieza musical.

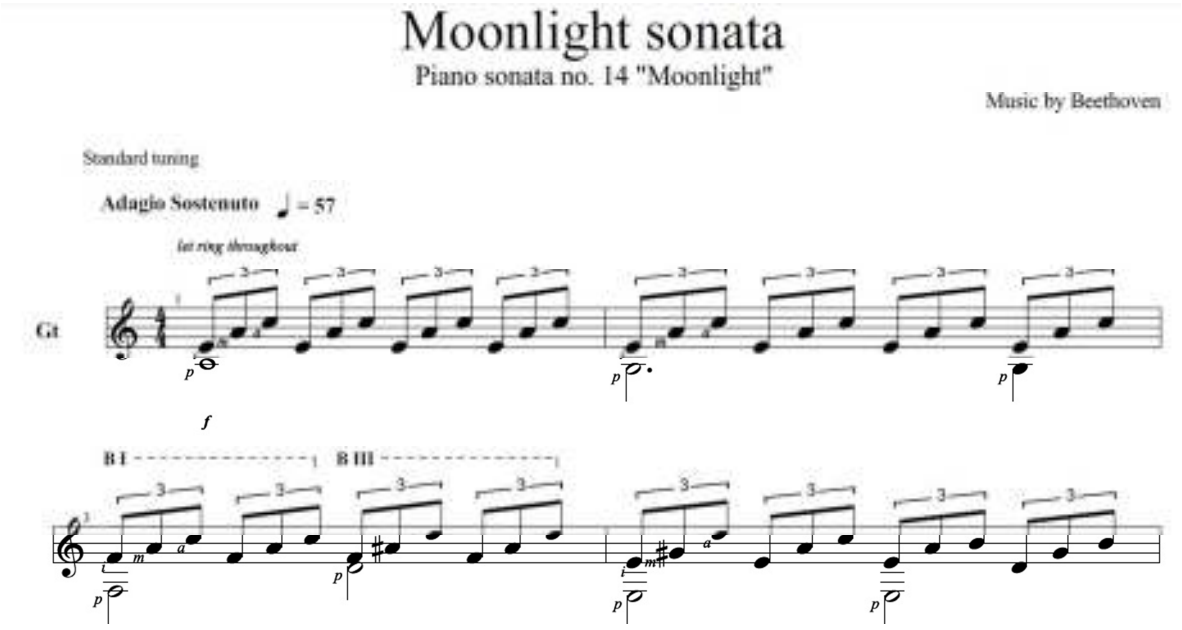


Figura 2.7. Representación en signo (Partitura) fragmento de Moonlight sonata

En la figura 2.7 podemos observar las notaciones pertinentes que se le sugieren al interprete para ejecutar la melodía. En algunos textos se ha sugerido escribir este tipo de códigos basados en la representación de las partituras en un formato llamado *MusicXML* [18]. En el cual se describen de manera muy similar los aspectos de interpretación de la canción. En un archivo XML las notas y medidas son escritas en forma de código, de esta manera una maquina puede almacenar toda aquella información como la armadura y características de una canción ver figura 2.8 como referencia a este descriptor.

```
                <tuning-step>G</tuning-step>
                <tuning-octave>3</tuning-octave>
            </staff-tuning>
          - <staff-tuning line="5">
                <tuning-step>B</tuning-step>
                <tuning-octave>3</tuning-octave>
            </staff-tuning>
          - <staff-tuning line="6">
                <tuning-step>E</tuning-step>
                <tuning-octave>4</tuning-octave>
            </staff-tuning>
        </staff-details>
    </attributes>
  - <note>
      - <pitch>
            <step>C</step>
            <octave>3</octave>
        </pitch>
        <duration>4096</duration>
        <voice>0</voice>
        <type>whole</type>
      - <notations>
          - <technical>
                <fret>3</fret>
                <string>5</string>
            </technical>
            <articulations/>
          - <dynamics>
                <mf/>
            </dynamics>
        </notations>
    </note>
  - <backup>
        <duration>4096</duration>
    </backup>
  </measure>
</part>
</score-partwise>
```

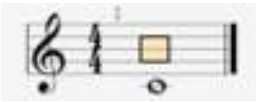


Figura 2.8. Fragmento de código de Music XML generado por Guitar Pro 6 y su equivalente en notación clásica.

Hay muchas formas de generar la representación de partituras digitalmente, una de ellas seria copiar la partitura manualmente y representarla en el código XML. Lo implicaría una cierta dificultad a la hora de hacer este tipo de trabajo y con ello acarear errores en la notación musical aunque. Actualmente existen una diversidad de software capaz de editar estas partituras en un formato entendible por el usuario y posteriormente el programa es capaz de pasar la música a un formato XML. En la figura 2.8 se genero el código con la

ayuda de un editor de este tipo llamado *Guitar Pro 6* aunque hay una diversidad de estos software como *Sibelius* y *Finale.*

Estos softwares ofrecen una gran diversidad de herramientas para el compositor, tales como poder controlar el tiempo dinámicamente; o la opción de poder capturar la ejecución de su instrumento MIDI a la computadora, consultar [19] y [20] para mayor referencia de estas herramientas.

2.6.2 Representación por forma de onda.

Esta representación puede resultar familiar para ciertas personas y más aun cuando se dedican a la edición de audio y masterización de grabaciones. Hablando físicamente el sonido es producido por vibraciones generadas por una fuente ya sea la voz de un cantante, las cuerdas de una guitarra o el aire que en un clarinete. Estas perturbaciones causaran en las partículas en el medio que se transporten por medio del aire alternando su nivel de presión. Cuya consecuencia será que tenga una determinada forma de onda, que a su vez será interpretado por nuestro oído como sonido o si este es captado por un micrófono será convertido en una señal eléctrica.

Gráficamente los cambios de presión pueden ser representados en *presión-tiempo* al que se va a referir como forma de onda el cual muestra las variaciones de la presión del aire. Como es sabido los puntos se alternan a un cierto tiempo para ser llamada periódica. En este caso el periodo puede ser definido por el tiempo en que se repiten los puntos de presión más altos de estos periodos. Es evidente que podemos obtener la frecuencia que es medida en Hz ya que es reciproca al periodo, en un ejemplo claro podemos considerar una onda sonora senoidal como se muestra en la figura 2.9.

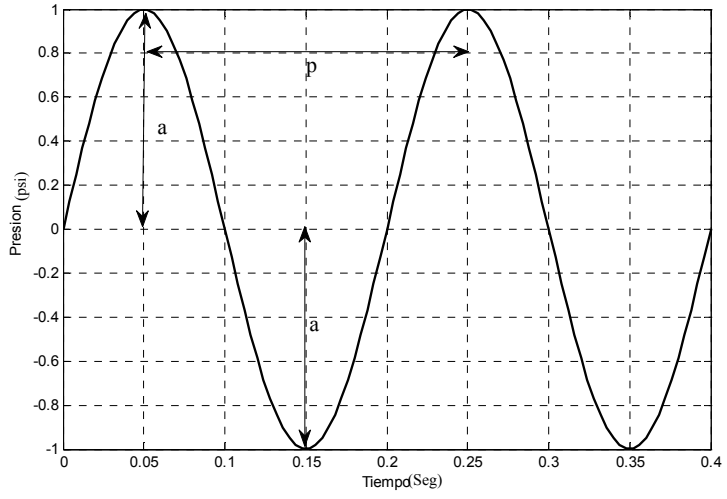


Figura 2.9. Representación de forma de onda de una señal periódica con un frecuencia de 5Hz donde a es la amplitud de la presión del aire y p es el periodo de la onda.

En el ejemplo podemos ver claramente los parámetros de una onda con 5Hz de frecuencia esto en el sonido es considerado como el armónico puro de una nota. La propiedad que relaciona esta frecuencia que es percibida es conocida como *pitch,* como se ha mencionado ya anteriormente. Un ligero cambio de frecuencia en el pitch no representa mucho a la hora de percibirlo, por lo que regularmente se asocia un determinado rango de frecuencias correspondientes a un pitch (nota).

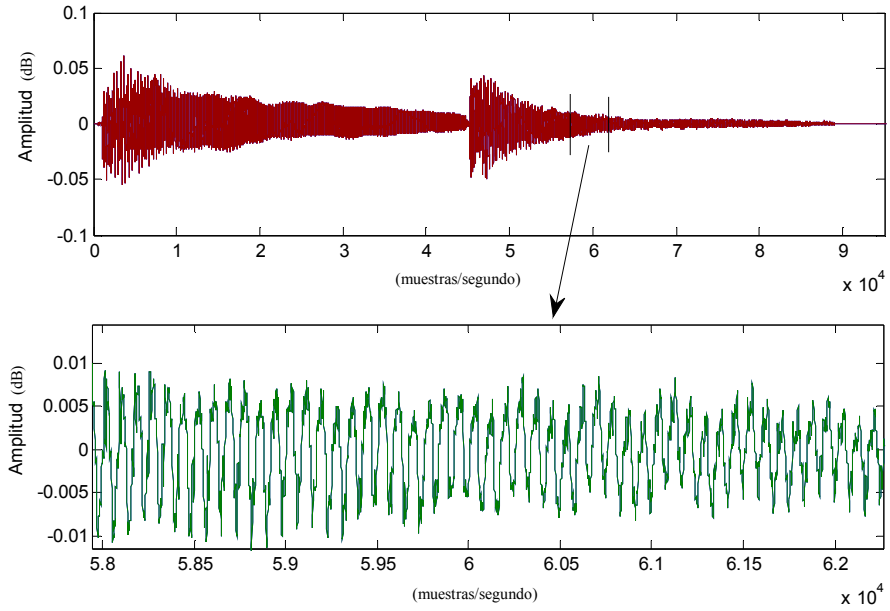


Figura.2.10. Octavas y periodicidad de la onda

Si es tocada en piano la nota C4 en el cual su pitch concierne a una frecuencia de 261.6 Hz figura 2.10. Es imperceptible la variaciones en un rango de 15.58Hz que es donde se encuentra la banda inferior de la nota sucesiva C4#. Es posible percibir la distancia que existe entre C4 y C5 ya que los pitches de estas se encuentran 261.6Hz y 523.25Hz sucesivamente como puede observarse estas notas son armónicas ya que C5 tiene el doble de frecuencia que C4, esta relación es conocida como octava. Es decir C4 y C5 son octavas lo que en estas notas da a nuestro oído una sensación de harmonía. En la música occidental principalmente, es utilizada una escala cromática la cual se tomara como referencia para este trabajo de tesis.

2.6.3 Representación Interfaz Digital de Instrumentos Musicales (MIDI).

La representación de la música en MIDI es un protocolo de información de comunicación para dispositivos electrónicos como computadoras, sintetizadores, secuenciadores, etcétera para la generación de tonos musicales. Este protocolo esta basado en representar la música de manera similar a una partitura. Así como su forma de onda, es decir un híbrido de estas

dos brindando información importante tal como la dinámica de la música para una interpretación más específica de una melodía.

La aparición de los sintetizadores digitales en la música exige tener un orden debido a los diversos problemas que existen en compatibilidad entre los dispositivos, exigiéndose forzosamente que existiera un lenguaje en común. Para el cual por encima de las características de un fabricante fuera compatible con diversos dispositivos involucrados. En 1983 es cuando surge la primera especificación del MIDI que se publica [21].

El MIDI permite que un músico pueda controlar un instrumento musical electrónico remotamente; si consideramos un sintetizador en el cual se pulsa una tecla del instrumento este tendrá una respuesta en forma de sonido. El cual tiene las características esenciales como el tiempo que dura la nota, la tonalidad y la intensidad con la cual se toca. Este evento puede ser guardado como un mensaje y/o mensajes MIDI; el cual puede ser disparado por el músico y el instrumento ejecutara las instrucciones que se han guardado en la memoria en forma de mensajes MIDI los cuales tienen codificados. El comienzo de la nota, la velocidad y el final de esta. Cabe aclarar que el MIDI no transmite señales de audio, solo tiene los mensajes de eventos que se deben de interpretar por el instrumento que los esta recibiendo.

El MIDI, es formato que contiene en valores numéricos de 0 – 127 asignadas las frecuencias de las notas (pitch), que serán generadas por el instrumentos. Esto es muy similar a la asignación de notas que tenemos en un piano acústico. El cual tiene 88 teclas, en la figura 2.11 se observa una porción de un piano añadiendo nota y numero MIDI. El MIDI codifica estas notas asignando un número; por ejemplo, si escogemos el número 45 este número corresponde a A2. Así de esta manera los tonos son asignados a un número específico. La escala del MIDI comienza desde el C0 a G#9, en cuanto a la velocidad de la nota; de manera similar, esta definida por un número entero entre 0 y 127. Esta básicamente controla la intensidad del sonido en caso de que inicie una nota, y controla la decadencia de la nota cuando esta finalizando; así como tiempos de activación física de ciertas notas y su duración. Aunque esto dependerá del instrumento o sintetizador que se use.

El Canal MIDI esta dado por un valor entero de entre 0 y 15. Lo cual indica intuitivamente el instrumento que tiene el respectivo número de canal. Sin contar que cada canal es capaz de producir sonidos polifónicos; es decir, múltiples notas al mismo tiempo. Y por ultimo, la marca de tiempo es representada por cuantos pulsos de reloj deben de esperarse respecto a la siguiente nota a ser ejecutada.

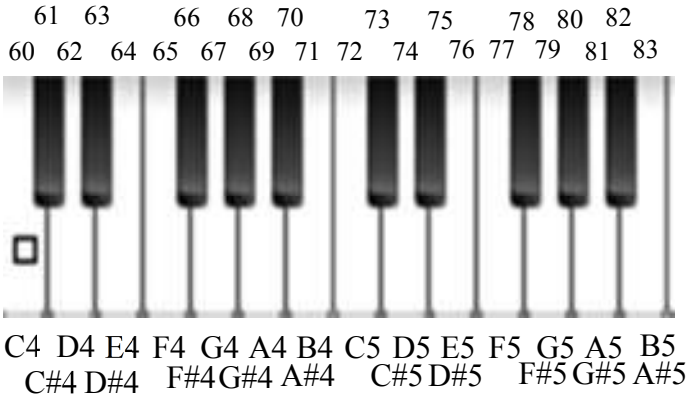


Figura. 2.11 Asignación de números MIDI de dos octavas (C4-B5).

Una característica importante de la música MIDI es la versatilidad y la poca memoria que ocupan estos archivos. Como en la notación de una partitura, el MIDI puede medir el tiempo de duración de una nota y cada *cuarto* será medido por un cierto número de repeticiones de reloj (PPQN). Por lo que la precisión de duración esta definido en microsegundos. Poniendo un ejemplo 600000μs por cuarto, corresponde a una velocidad de 100 bpm. El tiempo siempre esta especificado en la cabecera del archivo MIDI.

Generalmente un archivo MIDI puede ser generado por un instrumento MIDI. Como se ha mencionado, aunque existen diversas formas de generar las instrucciones. Programas de edición de audio comúnmente utilizados como *Adobe Audition, Cubase* [22,23] entre otros, que permiten crear secuencias utilizando una interfaz conocida como piano roll. La cual permite escribir de una manera grafica las instrucciones MIDI al ejecutarse, así generando una gran variedad de sonidos sintezados o utilizando bancos de sonidos. En la figura 2.12 se puede apreciar una interfaz de este tipo en el cual se determinan las características con las que se ejecutaran las notas introducidas en el mensaje MIDI.

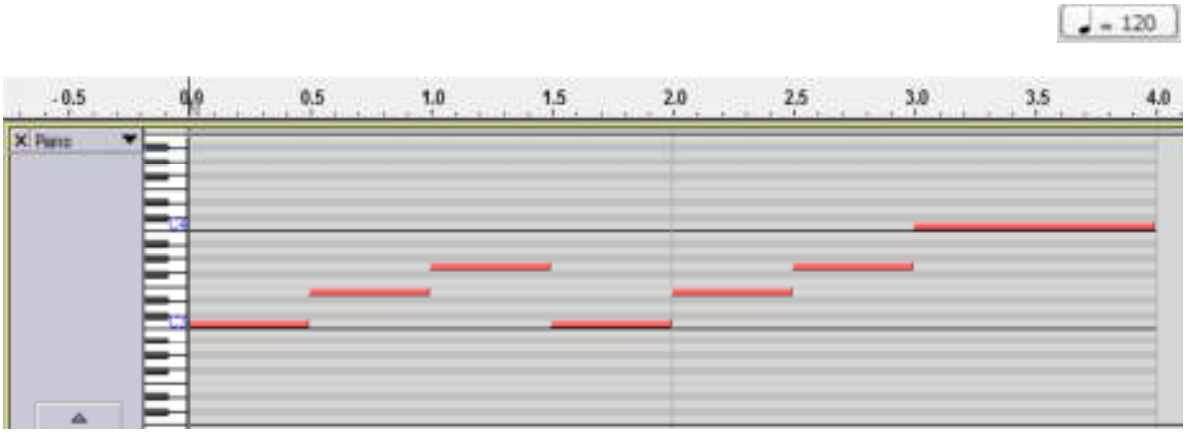


Figura. 2.12 Representación en Piano Roll.

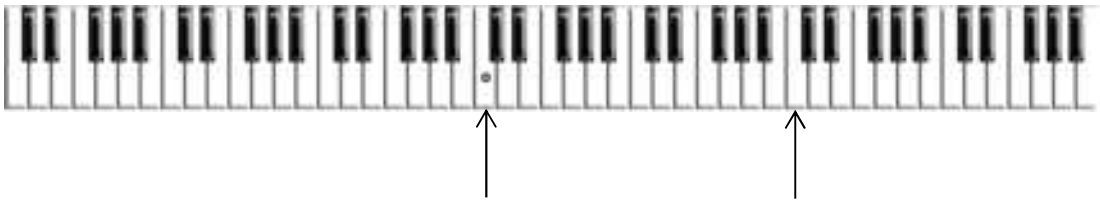
El MIDI tiene algunas limitantes respecto a una partitura; esta solo puede representar algunas eventos en la melodía como un interruptor encender nota - apagar nota. Definir tiempo de duración y el número de la nota; lo que carece de una interpretación precisa. El MIDI no tiene distinción entre bemoles y sostenidos. Para un MIDI estos son lo mismo ya que este tiene el mismo valor asignado. Ejemplo si escogemos la nota G5# y A5b para la interpretación MIDI los dos valen 80, por este motivo se crearon herramientas como el XML music para poder dar una especificación mas correcta y sin perdidas de información a la hora de interpretar la música. Ya que un ser humano puede apreciar la música y un MIDI puede parecerle monótono.

2.7 La voz en la música (tonalidades).

Dentro de las tesituras en las cantantes profesionales, y dentro de la voz de cualquier persona, se encuentran en distintos rangos de octavas. En promedio 2 octavas, son las cotas alcanzando su cero fónico y su límite fónico de cada individuo.

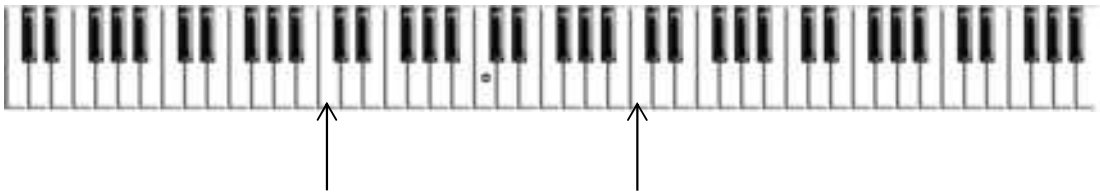
Para distinguir los tipos de voces en el uso coral se pueden distinguir cuatro principales grupos. Cuya tesitura es menor de 2 octavas para así incluir a las voces menos experimentadas.

- soprano: de do_4 a do_6



- contralto: de mi_3 a mi_5

- tenor: de do_3 a do_5



- bajo: de mi_2 a mi_4

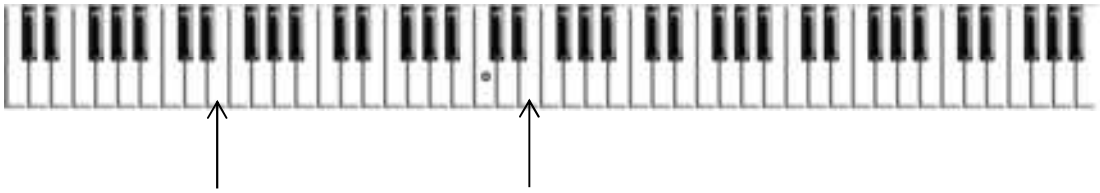


Figura 2.13 Formación de los tipos de voces

Haciendo referencia a la frecuencia, esta es la responsable de las tonalidades que utilizamos en la música. En consecuencia que utilizamos al hablar o al cantar, y cada uno de estos tonos y semitonos se asocia un rango de frecuencias en el cual se le asigna el nombre de su nota figura.2.14

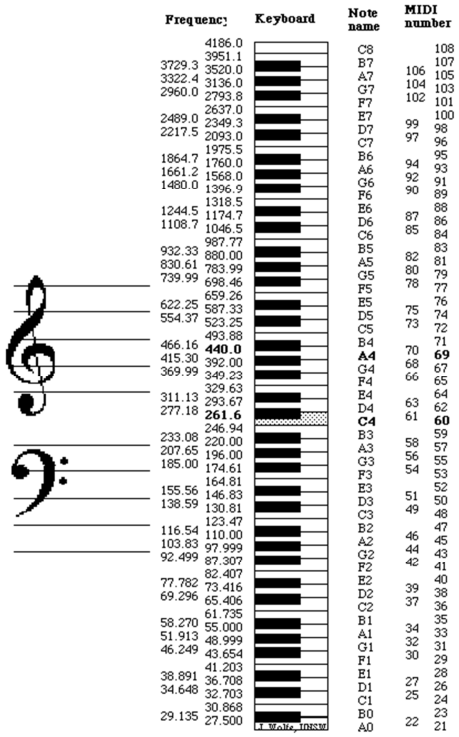


Figura 2.14 Asignación de tonalidad de acuerdo a frecuencia

Aunque la música no solo se compone de notas tocadas aleatoriamente. Existen una diversidad de “reglas” las cuales se toman en cuenta para poder llamar música al efecto de esta secuencia de notas. Entre los requisitos más importantes para esto se encuentra el ritmo, que es la frecuencia de repetición de sonidos y silencios; los cuales pueden tomar diversos intervalos de tiempo, ya sean de mayor o menor potencia. En otras palabras acentos o notas suaves figura 2.15.

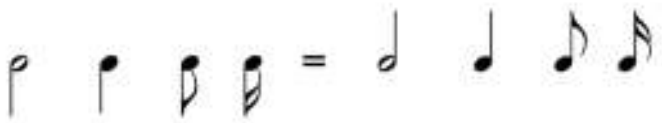


Figura 2.15. Signos de temporización musical

Existe una gran diversidad de simbología musical, a la cual no es imprescindible hacer referencia para este estudio solo se tomara en cuenta aquella realmente necesaria (básica).

Otro de los elementos de la música al que podemos referirnos es el compás, el cual es la entidad de métrica musical compuesta por unidades de tiempo. En una obra musical escrita, las notas y los silencios que están comprendidos entre dos líneas divisorias que componen un compás. Un fragmento musical estará compuesto por el conjunto de compases que lo conforman, los cuales tendrán la misma duración hasta que se cambie el tipo de compás.

Los tres tipos de compás de subdivisión binaria son: 2/4, 3/4 y 4/4.

Los tres tipos de compás de subdivisión ternaria son: 6/8, 9/8 y 12/8.

2.8 Escalas.

Las escalas son una parte importante para la composición de una canción. Una escala en la música es una sucesión ordenada consecutivamente de un entorno sonoro en particular. Refiriéndose que son el conjunto de notas ejecutadas ya sea en forma ascendente o descendente. Una nota es ascendente si el sonido se torna más agudo y descendente si el sonido se vuelve mas grave. Dentro los cuales se consideran los llamados grados, que son las notas que se deben interpretar para poder conformar una melodía.

La utilización de una escala musical es meramente didáctica para poder sintetizar una melodía en particular. En ocasiones una canción es compuesta por una sucesión de notas que pertenecen a una clásica distribución de tonos, estas notas normalmente pertenecen a una misma escala musical; por lo cual estas pueden ser resumidas en la armadura de la canción al inicio de la partitura.

En consecuencia podemos considerar que una escala musical es un ordenamiento de ciertos sonidos característicos de una melodía. Lo que nos proporciona como resultado una gran diversidad de estilos musicales. Esto es algo que puede apreciarse a través de la época y la

región donde se desarrolle la escala. Un claro ejemplo de esto es la diferencia existente entre la música oriental y occidental, que cuentan con diferencias musicales claramente marcadas en la percepción y ejecución de los instrumentos. Sin mencionar las diferencias que existen entre los instrumentos.

2.9 Métodos de clasificación.

Los métodos de clasificación son una herramienta útil, regularmente utilizada para poder hacer un ordenamiento de *n* elementos y poder discernir entre si un elemento nuevo pertenece a cierto conjunto. En otras palabras, sirve para distinguir si un elemento de estudio puede ser relacionado con otro elemento o conjunto que pertenecen a una base de datos, con la finalidad de realizar la identificación del elemento en cuestión.

Existen distintos tipos de métodos de clasificación, como la clasificación por el método de Shell. En este se aplica el método de inserción para separar entre elementos más cercanos y más lejanos, para finalmente tener una sucesión de elementos por las cercanías entre sus elementos. Otro de los métodos del que podemos hablar es la clasificación rápida. El cual solo se dedica a dividir los elementos para ordenarlos: divide en grupos cada vez más pequeños hasta lograr una identificación.

En la actualidad existen clasificadores mas avanzados los cuales su propósito es brindar respuestas mas inteligentes y no solo basarse en ordenamientos por distintos métodos. Algunos de estos métodos son las cada vez mas usadas redes neuronales, las cuales hoy en día tienen cada vez mas auge de diversas áreas de la ingeniería, informática, medicina entre otras. En trabajos como el expuesto en [27] la red neuronal esta especialmente hecha para hacer un análisis técnico del mercado de valores; y definir si es conveniente comprar o vender en un sistema de predicción de tiempo. Este tipo de aplicaciones en las cuales incluso se involucran cuestiones como el mercado de valores para hacer predicciones, tienen un gran peso ya que su buen funcionamiento deriva en ganancias o perdidas millonarias.

Las redes neuronales pueden ser utilizadas incluso en temas delicados como es el diagnostico de pacientes. En [28] en el cual se propone una red neuronal de algoritmo constructivo, que ayuda a generar patrones de reconocimiento para detectar las posibles causas de los síntomas del paciente incluyendo problemas como el cáncer.

Otro método de clasificación de relevancia en la actualidad son las maquinas de soporte vectorial. Que al igual que en las redes neuronales tratan de clasificar de manera optima para su mejor identificación. En estas se tienen que realizar ciertas etapas para que realice la clasificación. Dado que esta será entrenada con diversas muestras, las cuales se etiquetaran dentro de una cierta clase; serán trasladados a un hiperplano y finalmente se generaran puntos que pertenecen a dicha clase. Una vez entrenada la maquina de soporte, podemos ingresar un vector para realizar una clasificación para poder determinar su clase.

En [29] se utilizan para la clasificación digital de música, especialmente para la clasificación de canciones por artista tomando las principales características de las canciones.

Sin embargo estos métodos de clasificación requieren de un uso en cálculos computacionales excesivamente grandes. Por lo que para nuestra aplicación utilizar alguno de estos métodos para clasificar las canciones involucradas podría derivarse en un tiempo de búsqueda alto. Al ser alto el tiempo de respuesta de la aplicación se torna impráctica. Por lo que se opto por usar un sistema de identificación sencillo pero que refleja eficiencia en resultados y tiempo de cálculos que será visto en el siguiente capitulo.

2.10 Conclusiones.

Para realización de esta aplicación es de importancia tener en cuenta las diversas formas de obtención del pitch. Así como las características que podemos obtener del archivo de audio. Estas nos ayudaran a tener un descriptor que nos ayude a la identificación de la canción deseada. La comprensión de como esta compuesta la música es un elemento básico para nuestra aplicación por lo que al conjuntar ambas, tendremos mayor probabilidad de tener un buen resultado.

Captitulo.3

ANALISIS DEL DESARROLLO DEL SISTEMA

3. Obtención de parámetros para la representación de la canción.

Como se ha estado hablando en capítulos anteriores, la importancia de obtener las características de una canción es fundamental para el reconocimiento de la canción. En este trabajo de tesis se proponen dos maneras de representación de una canción. El primero se basara en hacer una búsqueda de la canción en base al reconocimiento de notas en la voz humana (voz vs voz); es decir, el sistema analizara diversos tarareos de una misma canción para obtener una base de datos robusta, y un sistema no muy complejo.

El segundo se trata de una descomposición de las notas musicales tanto en tiempo como en energía. Haciendo posible un reconocimiento entre distintos instrumentos musicales. En pocas palabras comparar un sonido polifónico con la voz (música vs voz). Este sistema representa un reto mayor que el anterior, ya que considerando todos los sonidos involucrados de una canción puede tornarse problemático. Y es en este caso donde se ha centrado las actuales investigaciones en el área. Aquí se propondrá un método de representación musical para este caso.

3.1 Planteamiento Sistema 1 (voz vs voz).

En primera instancia, se ha planteado que la voz tiene diferentes tonalidades según la persona que este hablando. Y encontrar la frecuencia fundamental en esta es una tarea necesaria para una comparación exitosa.

Basándose el descriptor de MPEG-7 de Frecuencia Fundamental de Audio o Audio Fundamental Frequency (AFF) [13]. El cual provee de una estimación de una frecuencia

fundamental f_0 en segmentos donde se asume la señal como periódica, puede ser una herramienta útil dado que se obtienen candidatos de frecuencias fundamentales los cual se resume a una obtención del pitch.

En el estándar de MPEG-7 dado que la estimación no se normaliza, debemos proporcionar parámetros adicionales que deben ser definidos junto con un valor estimado de f_0 para poder formar el descriptor AFF. Estos son los siguientes parámetros:

- *lolimit:* el límite inferior del rango de frecuencias en el cual esta siendo buscado f_0.
- *hilimit:* el límite superior del rango de frecuencias en el cual esta siendo buscado f_0.
- Una medida de confianza en la presencia de periodicidad en la parte que esta siendo analizada, contenida en un valor entre 0 y 1.

La medida de confianza permite denotar el respectivo grado de periodicidad de la señal. Si este número es 0, eso quiere decir que el intervalo analizado no es periódico. Si es 1 quiere decir que es periódico.

El descriptor puede ser usado junto con otros descriptores; como harmonías de audio AH, para proporcionar información mas detallada sobre la estructura de los *armónicos* del sonido para construir un descriptor de un nivel superior.

En la figura 3.1 podemos visualizar el análisis de un fragmento de canción muestreada a 22050Hz, analizada con el descriptor utilizando una lolimit = 50 y un hilimit = 550 calculada en Matlab.

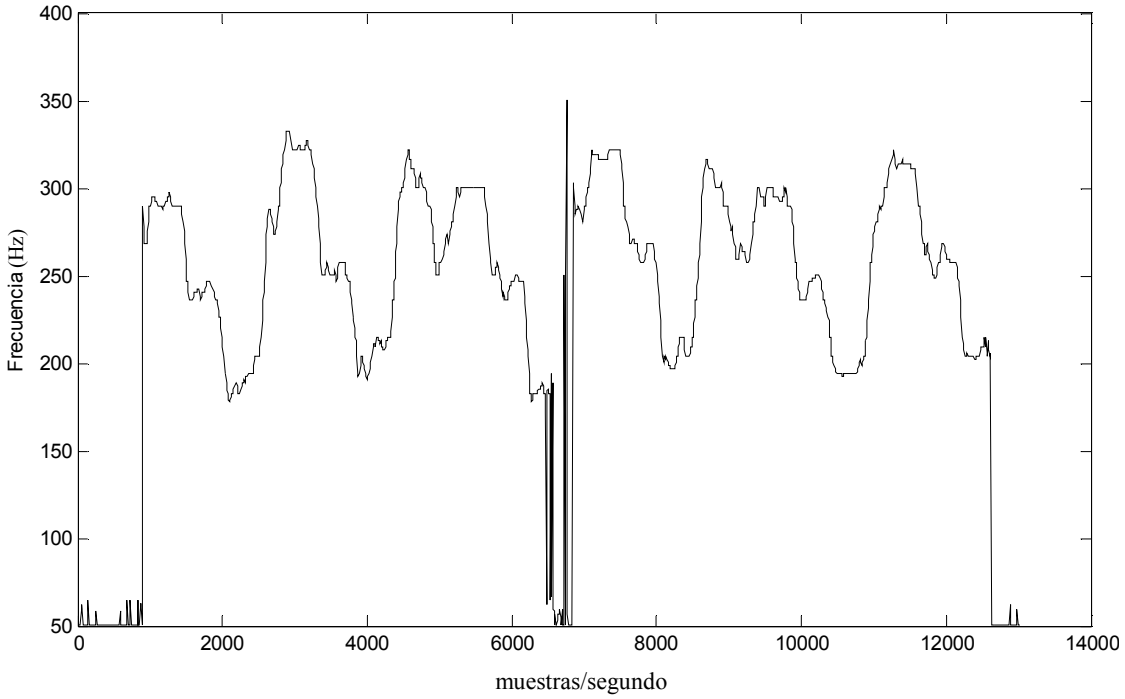


Figura.3.1.AFF aplicado a una señal de voz.

En la figura 3.1 podemos apreciar los distintos cambios de frecuencia que se encuentran en el tarareo de una persona. Observamos las variaciones de frecuencia que describe en el tiempo. Sin embargo, podemos notar que es un poco difícil precisar de qué nota se trata, ya que este descriptor solo nos proporciona frecuencias fundamentales sin tomar en cuentas las características ya antes mencionadas como: tonalidad, tiempo, ritmo, etc.

Una de las técnicas mas usadas para la obtención del pitch es la autocorrelación (cap.2.4). En nuestro caso hemos optado por utilizar este medio para obtener la frecuencia fundamental.

Basándose en el descriptor, determinamos los diferentes aspectos los cuales se tomaron en cuenta para la representación de la música que se enumeran a continuación:

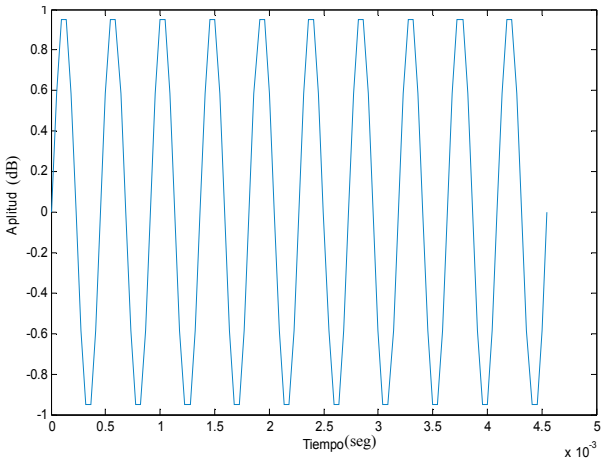
- Pitch
- Procesado del pitch.
- Obtención de la escala cromática (Chroma).

Estos son los puntos más relevantes para el análisis de la música (tarareos) en este primer sistema.

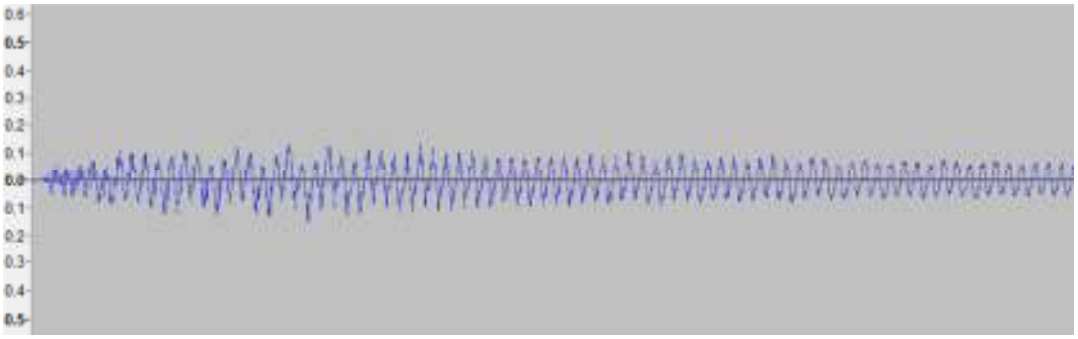
3.1.2 Pitch.

El pitch, es una propiedad del sonido que puede ser percibida en una escala de frecuencia determinada y puede ser utilizado para interpretar una melodía. El pitch de un sonido cualquiera puede ser medido en Hertz y a manera de ejemplo es posible representarlo con una señal senoidal (figura 3.2).

El pitch esta ligado a otra característica del sonido llamada timbre [24]. Lo cual para la percepción del ser humano es muy importante; ejemplo, supongamos que percibimos el canto de un ave, de cierta manera podemos percibir la tonalidad del canto del ave. ¿Pero como sabemos que es un ave la cual esta emitiendo el sonido?, el timbre es lo que nos va a brindar esto, dado que el timbre es la mezcla de frecuencias acompañadas con la frecuencia fundamental (pitch) que otorgan las características al sonido que oimos, como es: la posición, la intensidad y otra serie de atributos que puede no se hayan definido aún. Y es de esta manera como el ser humano percibe la música. Hay notas musicales que podemos percibir, y con el timbre podemos definir si es un guitarra, un piano, un violín etc.



a)



b)

Figura 3.2. a) Representación senoidal del pitch de A4 (440Hz), b) A4 en piano.

El pitch es una propiedad subjetiva, la cual implica problemas involucrados en psicoacústica. En el cual se realizan los estudios de procesamiento y de la percepción auditiva. Para el tema de estudio expuesto en este trabajo de tesis le veremos como la frecuencia fundamental de una tonalidad.

Al percibir un sonido, existe de por medio como es conocido, una frecuencia fundamental. Pero esto implica que esta va tener presentes armónicos los cuales van a influenciar en la representación musical. Este evento puede ser percibido de una mejor forma en la representación de sonidos polifónicos que veremos mas adelante.

En la forma onda de una señal de audio, no puede percibir ninguna periodicidad aparente, como en la representación del pitch con una onda senoidal. Hay una gran variedad de instrumentos y timbres de voz. Que pueden ser analizados, los cuales tendrán formas de onda muy complejas y en adición en ciertos intervalos de tiempo habrá similitudes. Es por esto que existen diversos procesos para obtener el pitch de una onda de audio. Dos de estos

serán expuestos aquí, como son el método de autocorrelación y la obtención del pitch por medio de bancos de filtros.

3.1.3 Detección del pitch por el método de la autocorrelación.

Hay muchos métodos con los cuales calcular el pitch y en general todos tienen existo en la obtención del pitch. Entre ellos se encuentra el método de la autocorrelación. Que aunque en este se despliegan algunos picos en ciertos periodos de tiempo, su cálculo obtiene resultados favorables para los propósitos de nuestra aplicación.

Una de las características para el análisis de la obtención del pitch, es que el cálculo debe ser obtenido por medio del uso de ventanas en periodos cortos de tiempo dado que si tomáramos un periodo muy grande el pitch no significa nada ya existirían diversas variaciones de frecuencias. Por lo cual se debe de tomar en cuenta un tamaño de ventana ideal para el análisis. La ventana de análisis debe contener de 2 a 3 periodos de pitch esto se traduce que para el caso de la obtención del pitch para frecuencias relativamente altas la ventana de análisis debe ser corta (5-20ms) y para frecuencias mas bajas ventanas mas largas (20-50ms).

Dada una señal discreta *x(n)*, definida para toda *n*, la función de autocorrelación esta descrita de la siguiente manera:

$$\phi_x(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)x(n+m) \tag{3.1}$$

Como se mencionó la autocorrelación es una transformación de una señal. Que hará visible su estructura en la forma de onda. Para una detección del pitch si asumimos que *x(n)* es periódica y cuenta con un periodo de onda P es decir *x(n)= x(n + P)* para toda *n*, entonces podemos ver que.

$$\phi_x(m) = \phi_x(m+p), \tag{3.2}$$

La autocorrelación, también tiene el mismo periodo y si hay periodicidad en la función de autocorrelación quiere decir que hay periodicidad en la señal.

Pero considerando una señal no estacionaria como la voz, el concepto de una autocorrelación en periodo de tiempo largo no significara nada. Por lo que suena razonable definir una función de autocorrelación de tiempo corto, la cual operara en segmentos cortos de la señal como sigue:

$$\phi_\ell(\mathrm{m}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=0}^{\mathrm{N}'-1} [\mathrm{x}(\mathrm{n}+\ell)\mathrm{w}(\mathrm{n})][\mathrm{x}(\mathrm{n}+\ell+\mathrm{m})], \quad 0 \leq \mathrm{m} \leq \mathrm{M}_0 - 1 \tag{3.3}$$

Donde *w(n)* es una ventana apropiada de análisis, N es la longitud de la sección que esta siendo analizada *N'* es el número de muestras que esta siendo utilizado para el cálculo de $\phi_\ell(m)$ M_0 es el numero de puntos de autocorrelación calculados, ℓ es el índice de la muestra de inicio de cada trama.

Para aplicaciones de detección del pitch se fija el valor de $N' = N - m$, así que únicamente las muestras $(x(\ell), x(\ell+1), \cdots, x(\ell+N-1))$ son usados para el cálculo de la autocorrelación.

Los valores que generalmente son usados para M_0 y N son de 200 y 300 respectivamente que corresponden a un máximo periodo de pitch de 20ms y 30ms para un análisis de la trama (200 muestras con una tasa de muestreo de 10KHz).

Tomando en cuenta que para nuestra aplicación de considera tomar una tasa de muestreo de 8KHz, dado que esta velocidad de muestreo es utilizada por los teléfonos móviles y otros dispositivos. El análisis se lleva a cabo en estas condiciones tomando una ventana de análisis de la señal de 20 milisegundos que es una ventana que se considera apropiada para el análisis de la voz.

En la figura 3.3 podemos referenciarnos al procedimiento que se lleva a cabo gráficamente. Como es tomada la ventana, obteniendo la autocorrelación para conseguir los puntos de esta en frecuencia para un determinado tiempo, que será definido por el desplazamiento de nuestra ventana. De otra manera, el traslape que tenga esta con respecto a las muestras anteriores que ya han sido analizadas.

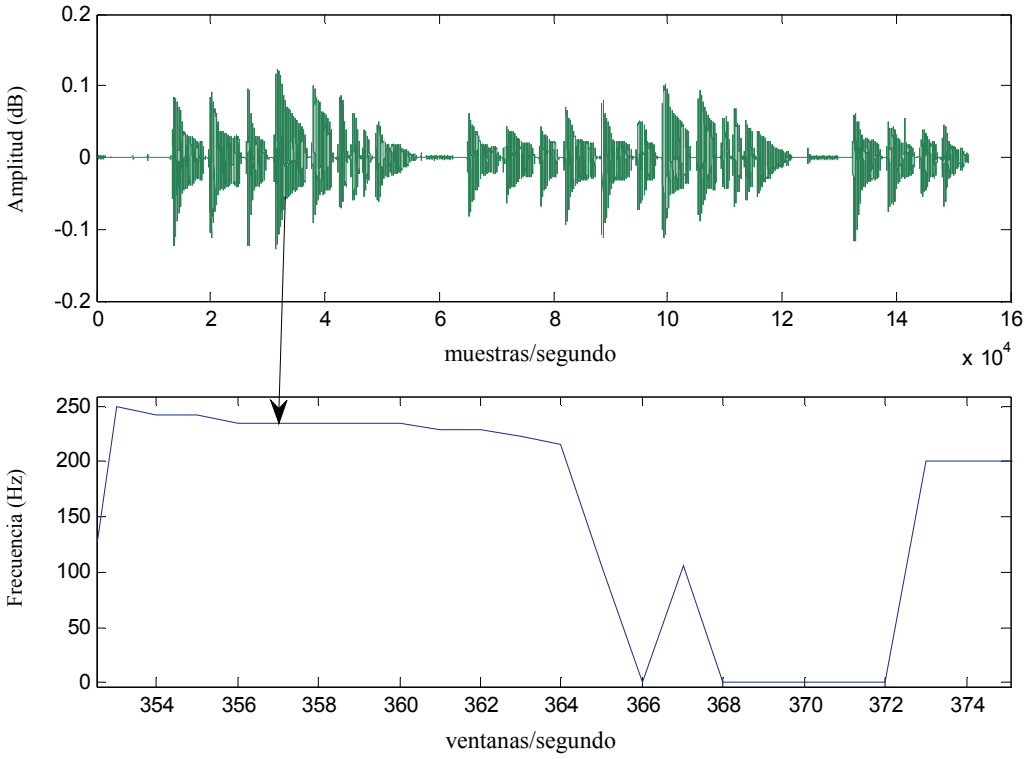


Figura 3.3 Fragmento de análisis audio a pitch.

Tomando en cuenta, que cada ventana que hemos elegido ha sido de 20ms y se ha realizado un avance de la ventana de Hamming de la mitad, que es una señal para evitar la discontinuidad al principio y al final de los bloques. Es decir 240 muestras se traslapan en la ventana, para tener una mejor resolución en el análisis del pitch. Encontramos un problema, en la figura 2.3, podemos llegar a apreciar que en la muestra 366 llega a 0 (lo cual representa silencio), inmediatamente vuelve a subir. Este es un error común que se observa en el descriptor AFF que afectan al análisis. Ya que en donde debería haber silencio existe una súbita subida de frecuencia. Principalmente estos errores afectan a la búsqueda ya que esto dependerá una buena clasificación de la canción.

Si tenemos una buena cantidad de errores en nuestra representación de la música, se afectara directamente con el resultado de nuestra búsqueda. Esto ha hecho que se buscara una solución a este problema en particular, dado que la señal presenta ciertas irregularidades en determinados puntos. Se opto por aplicar filtros, ya que estos se encargaron eliminan estos picos.

De igual manera que en una señal se eliminan esas señales no deseadas, en nuestro descriptor del pitch eliminamos lo que no es necesario o lo indeseado. Así logrando una buena visualización con la cual podemos comparar más adecuadamente.

3.2 Obtención de escala cromática con el pitch

La obtención del pitch y la variación de este en el tiempo, es una herramienta bastante útil para la caracterización de las canciones. Aunque en la música, solo existen un determinado número de notas musicales lo cual facilita la identificación de un tono.

Con el pitch encontramos *n* variaciones dentro de un fragmento de melodía. Así que para asemejar mejor una representación basada en la variación de la notas, con un análisis del pitch asignaremos a un cierto ancho de banda la nota referida en la escala cromática.

En la figura 3.4 observamos la asignación de la frecuencia central y el numero MIDI correspondiente a cada nota de acuerdo a las teclas de un piano estándar.

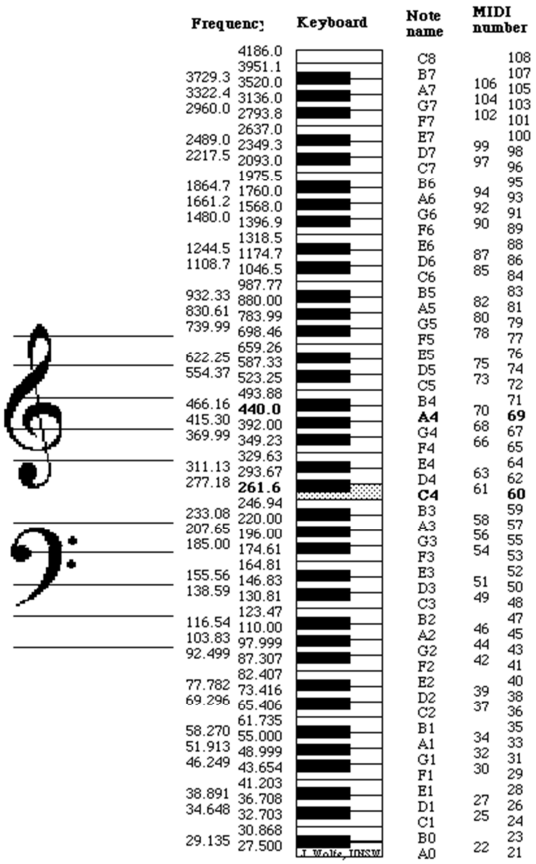


Figura. 3.4. Asignación de frecuencia central de las notas en el piano estándar

Dado que tenemos las frecuencias centrales acumuladas, en la tabla de la figura 3.4 podemos construir la escala en base a esta asignación de frecuencias centrales.

Las notas, como puede ser observado en la escala entre más alta es la nota, más grande es el ancho de banda que tiene respecto a su predecesora. Llevando un comportamiento exponencial, por lo cual la asignación del ancho de banda se ha llevado a cabo de la siguiente manera:

$$BW_1 = f_c - f_{c-1} \tag{3.4}$$

$$BW_2 = f_c - f_{c+1} \tag{3.5}$$

Donde BW_1, es el ancho de banda comprendido entre la frecuencia central de la nota actual y la frecuencia central de la nota anterior. f_c es la frecuencia central de la nota actual y f_{c-1} es la frecuencia central de la nota anterior.

De igual manera tenemos que BW_2, es el ancho de banda entre la frecuencia central de la nota actual y la frecuencia central de la nota posterior y f_{c+1} es la frecuencia central de la nota posterior.

Así, la cota inferior como la cota superior quedarían de la siguiente manera, respectivamente.

$$f_i = f_c - \frac{BW_1}{2} \tag{3.6}$$

$$f_s = f_c + \frac{BW_2}{2} \tag{3.7}$$

Por ejemplo, para la nota E2 tenemos una frecuencia central de 82.407Hz, para la nota D2# que es una nota anterior a esta tenemos una frecuencia central de 77.782Hz y para la nota posterior a E2 es decir F2; tenemos una frecuencia central del 87.307. Para tomar en cuenta que cada nota tiene un ancho de banda se decidió partir el ancho de banda que existe entre estas frecuencias centrales para así tener una distribución de estas sin que ninguna frecuencia quede fuera de la escala.

Para el primer ancho de banda tenemos que:

$BW_1 = 82.407 - 77.782 = 4.62Hz$

Y

$BW_2 = 87.307 - 82.407 = 4.9Hz$

El ancho de banda asignado para E2 será comprendido de la siguiente manera,

$$f_i = 82.407 - \frac{4.62}{2} = 80.094Hz$$

$$f_s = 82.407 + \frac{4.9}{2} = 84.857$$

Es decir que el ancho de banda designado para la nota E2 comprende de

80.097Hz a 84.857Hz $\quad BW = 4.45Hz$

En la tabla 3.1 vemos más ejemplos de la relación aplicada.

Nota	*Fc(Hz)*	*BW1(Hz)*	*Bw2(Hz)*	*Fi(Hz)*	*Fs(Hz)*	*BW(Hz)*
E2	82.4	4.6	4.9	80.0	84.8	4.7
F2	87.3	4.9	5.1	84.8	89.9	5.0
F2#	92.4	5.1	5.5	89.9	95.2	5.3
G2	97.9	5.5	5.8	95.2	100.9	5.6
G2#	103.8	5.8	6.1	100.9	106.9	6.0
A2	110.0	6.1	6.5	106.9	113.2	6.3
A2#	116.5	6.5	6.9	113.2	120.0	6.7
B2	123.4	6.9	7.3	120.0	127.1	7.1
C3	130.8	7.3	7.7	127.1	134.7	7.5
C3#	138.5	7.7	8.2	134.7	142.7	8.0
D3	146.8	8.2	8.7	142.7	151.1	8.4
D3#	155.5	8.7	9.2	151.1	160.1	8.9
E3	164.8	9.2	9.8	160.1	169.7	9.5
F3	174.6	9.8	10.3	169.7	179.8	10.0
F3#	185.0	10.3	11.0	179.8	190.5	10.6
G3	196.0	11.0	11.6	190.5	201.8	11.3
G3#	207.6	11.6	12.3	201.8	213.8	12.0
A3	220.0	12.3	13.0	213.8	226.5	12.7
A3#	233.0	13.0	13.8	226.5	240.0	13.4
B3	246.9	13.8	14.6	240.0	254.2	14.2
C4	261.6	14.6	15.5	254.2	269.3	15.1
C4#	277.1	15.5	16.4	269.3	285.4	16.0
D4	293.6	16.4	17.4	285.4	302.4	16.9
D4#	311.1	17.4	18.5	302.4	320.3	17.9
E4	329.6	18.5	19.6	320.3	339.4	19.0
F4	349.2	19.6	20.7	339.4	359.6	20.1
F4#	369.9	20.7	22.0	359.6	380.9	21.3
G4	392.0	22.0	23.3	380.9	403.6	22.6

G4#	415.3	23.3	24.7	403.6	427.6	24.0
A4	440.0	24.7	26.1	427.6	453.0	25.4
A4#	466.1	26.1	27.7	453.0	480.0	26.9
B4	493.8	27.7	29.3	480.0	508.5	28.5

Tabla 3.1. Asignación de bandas

Así que tomando como referencia la tabla 3.1. Asignamos las notas correspondientes de acuerdo a cada ancho de banda para obtener lo visto en la figura 3.5.

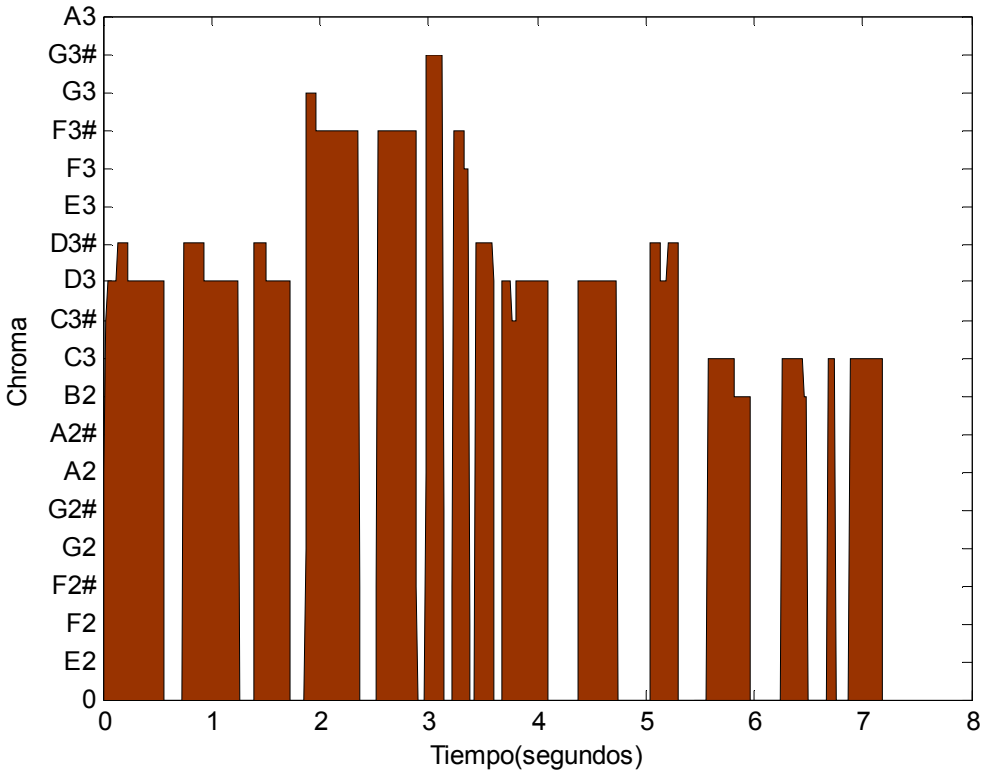


Figura 3.5. Representación de un tarareo en escala cromática.

En la figura 3.5 observamos un fragmento de una canción que ha sido tarareada y procesada para mostrarse como una representación tonal en escala cromática. **N**ótese que los picos de error excesivos han desaparecido, dado al empleo de los filtros. Sin embargo prevalece una variación pequeña de medio tono, en la mayoría de los casos esto es debido a que uno tiende a variar la tonalidad de su voz al cantar o en este caso al tararear. Entre menos experimentada sea la persona en el canto mas variaciones tendremos y el cambio tonal será mas grande. Este cambio, es algo que se debe tomar en cuenta para poder llevar a cabo el reconocimiento de la canción, el cual se tratara posteriormente.

Haciendo una comparación con su contraparte escrita podemos hallar similitudes a simple vista. En la figura 3.6 esta escrito el fragmento de canción que ha sido tarareado por un hombre con voz media. En la partitura comienza con E3 (Mi3) y en el fragmento interpretado con la voz a empezado con D3. Podría parecer que se ha errado en la interpretación ya que los tonos no son los mismos, pero si se observa la siguiente variación tonal en la partitura es A3 y en fragmento de voz es F3#.

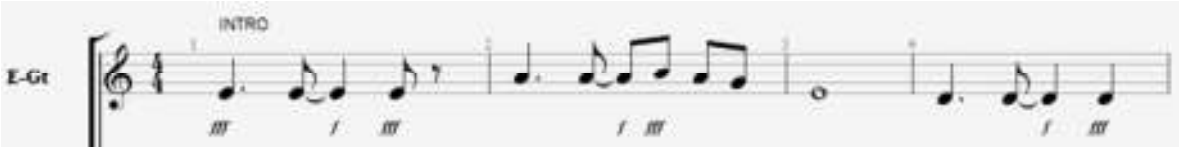


Figura 3.6 Fragmento de partitura de la canción tarareada

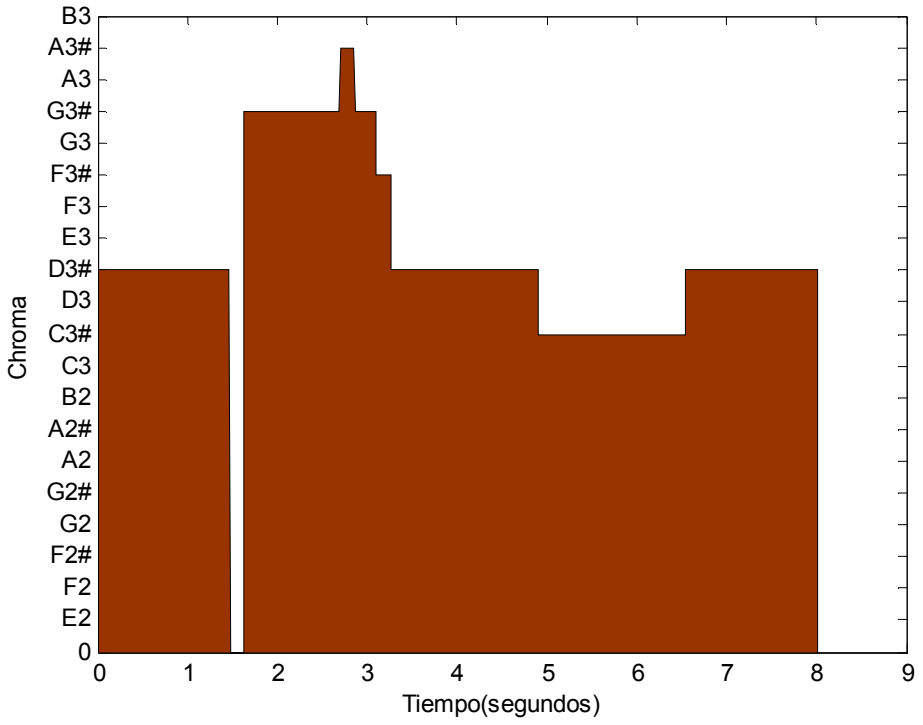
Si contamos el número de variación de tonalidades de la partitura de D3 a A3 tenemos que: son 6 medio tonos = 3 tonos, y la variación de la voz es de D3 a F3# es de 6 a 5 medios tonos contando las variaciones que se provocan normalmente en la voz humana. Así consecutivamente si seguimos la secuencia de notas que están escritas en la partitura son claramente comparables con lo obtenido en el análisis

Esto quiere decir, que con el modelo mostrado podemos representar la música que esta siendo generada por una única fuente en este caso la voz. Tenemos un punto de referencia que puede ser comparado con esta representación grafica tonal a la representación que se tiene escrita en una partitura.

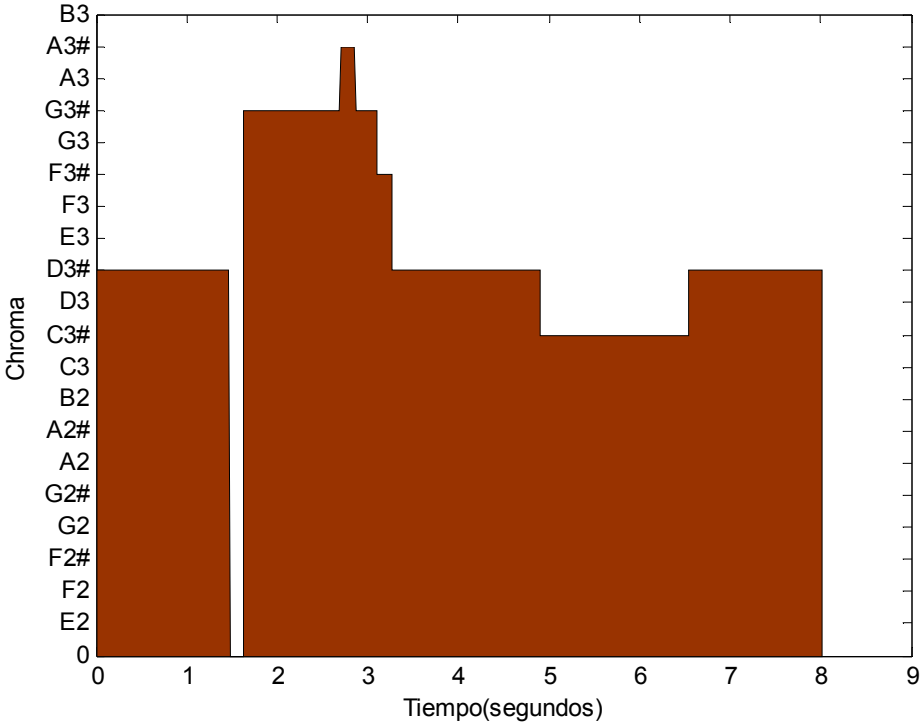
Una vez que se ha comparado con otra fuente de representación musical, introducimos otra fuente de sonido diferente a la voz humana; para observar si esta representación funciona de igual manera con otros tipos de instrumentos empleados en la música occidental.

Para la comparación empleamos 2 de los instrumentos musicales más usados en la música occidental de la actualidad: guitarra y piano.

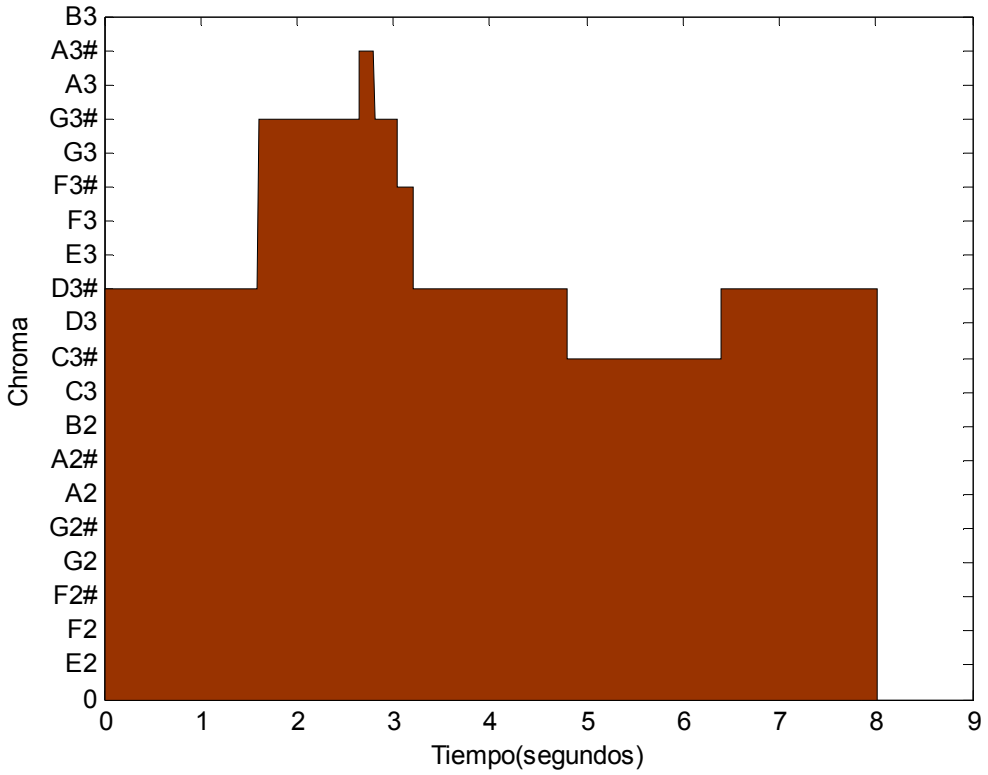
Para tener una referencia más clara se uso la misma partitura que de la figura 3.6 para los instrumentos mencionados.


B3
A3#
A3
G3#
G3
F3#
F3
E3
D3#
D3
C3#
C3
B2
A2#
A2
G2#
G2
F2#
F2
E2
0
Chroma
0 1 2 3 4 5 6 7 8 9
Tiempo(segundos)

a)


B3
A3#
A3
G3#
G3
F3#
F3
E3
D3#
D3
C3#
C3
B2
A2#
A2
G2#
G2
F2#
F2
E2
0
Chroma
0 1 2 3 4 5 6 7 8 9
Tiempo(segundos)

b)



c)

Figura 3.7. a) Representación del fragmento de la partitura interpretado en guitarra acústica.
b) Representación del fragmento de la partitura interpretado en piano.
c) Representación del fragmento de la partitura interpretado en guitarra eléctrica con distorsión

En las figuras anteriores, se puede observar que las interpretaciones en los instrumentos musicales no tienen mucha diferencia. Se podrían considerar prácticamente idénticos. Debido que las notas ejecutadas son las mismas y estas tienen el mismo pitch; lo que en realidad diferencia a los instrumentos unos de otros, es el timbre como ya se había mencionado con anterioridad.

Ahora al hacer una comparación entre el instrumento musical y la voz humana. Podemos notar que existe cierta discontinuidad de las notas que se han interpretado. Dado que en el instrumento musical el sonido, a pesar de no ser constante debido al ataque de las cuerdas el pitch no varía salvo que se cambie de tono; en el caso de la voz dependerá de la forma de cantar de la persona. Si un individuo tararea la canción con una silaba con t o p tendrá un *ataque* mas marcado. Sin embargo si la canción es tarareado con una silaba mas suave como l o n, el ataque no será tan marcado por lo que en consecuencia la representación se

vera muy similar a la de los instrumentos musicales. En caso anterior observa el ataque figura 3.4. En la figura 3.8 se observa este evento que se menciona.

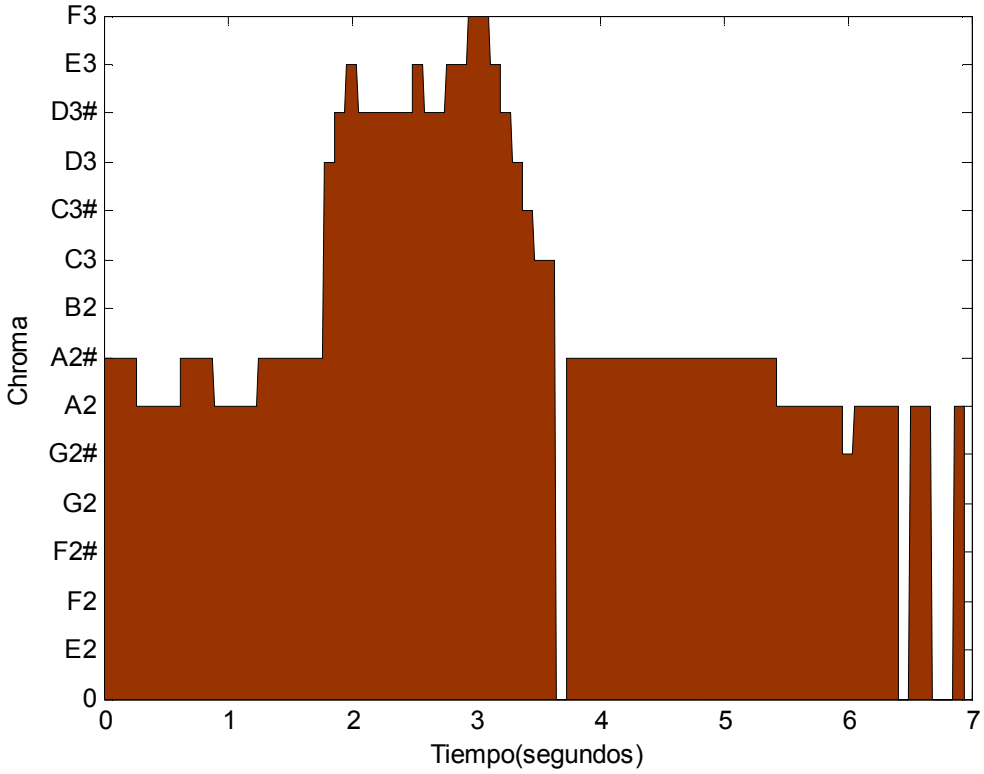


Figura 3.8. Tarareo con silaba na

Haciendo una comparación a primera vista existen similitudes evidentes entre todas. Pero considerando que en el caso de la voz y los instrumentos, existen variaciones tonales considerables las cuales en la música son conocidas como translaciones tonales o un cambio de clave. Es decir puede percibirse que se trata de una misma melodía, pero también es evidente que esta no tiene las mismas notas. Como por ejemplo si una canción empieza originalmente en C3, este puede que sea un tono muy bajo para el caso de una mujer lo cual no es un problema. Ya que una persona es capaz de tararear o cantar la canción aun sin que las tonalidades son las mismas. En pocas palabras la mujer tal vez comience a tararear la canción en D3 que es un tono más cómodo para su voz.

Considerando este problema podemos observar en la figura 3.8 existe un problema de *transposición* entre las melodías. Lo cual puede significar un problema a la hora de querer hacer una comparación entre la canción original y la voz del usuario.

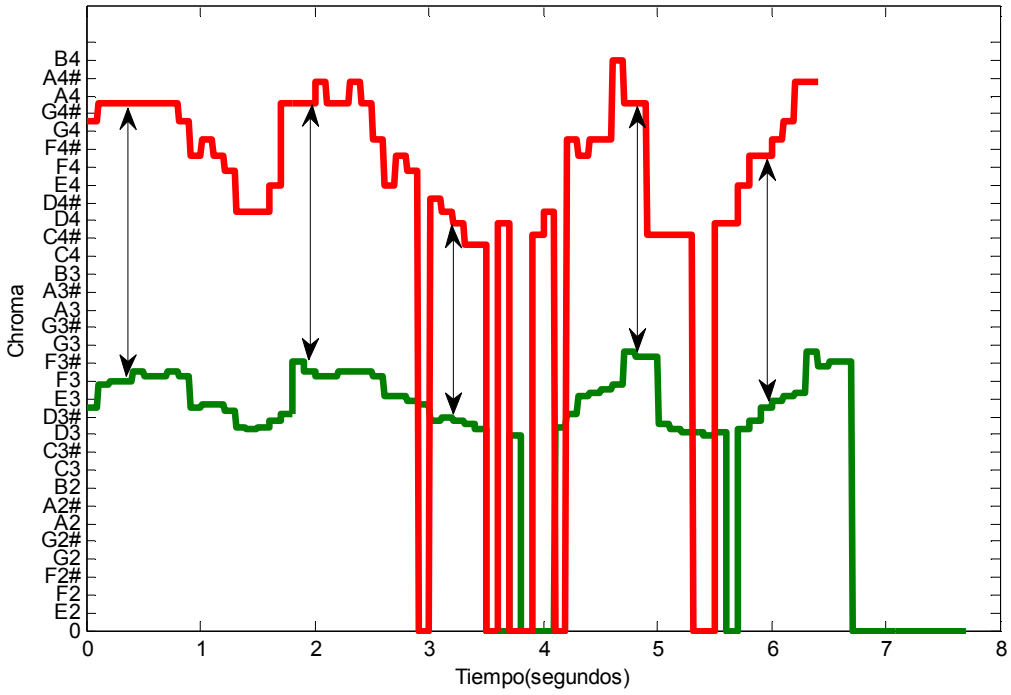


Figura 3.9 Transposición entre dos canciones.

La transposición es un problema que se puede resolver de una manera relativamente fácil. Ya que podemos auxiliarnos de la escala cromática, para lograr una interpretación valida para una comparación entre diversos cambios de clave.

Cuando se considera un cambio de clave en la música, existen algunas reglas que deben de seguirse. Pero para no profundizar en la teoría musical trataremos estas variaciones tonales como distancias entre notas.

En la escala cromática existen 12 notas diferentes las cuales comprenden las conocidas C, D, E, F, G, A, y B. Las cuales solo son 7, estas contaran con una alteración lo cual en música es representada por el signo # o por ♭ dependiendo de la alteración y significara sostenido o bemol correspondientemente. Estas alteraciones están comprendidas en la música como ½ tonos; y cada una de las notas de la escala anterior tendrá una variación de este tipo a excepción de las notas E y B, como ya se ha observado en las figuras anteriores.

Esta estructuración de ½ tonos o semitonos, se ha tomado de referencia para calcular las secuencias que sigue una canción y pueda ser referenciada para una búsqueda.

La separación que hay entre una nota y otra es de un tono a excepción de las notas E y B. Estas solo tienen un semitono de distancia. Esta distancia entre tonos es con la que se tiene que trabajar, para evitar el problema de la trasposición que se presenta en las distintas canciones.

3.3 Transposición.

Como se ha observado anteriormente, la transposición de una canción es un problema para nuestro sistema. Ya que al existir variaciones tonales es difícil hacer una comparación exitosa. Por lo que se tiene que lidiar con este problema para tener un mejor marco de comparación entre las canciones analizadas, y almacenadas en la base de datos con la canción analizada que ha sido enviada por el usuario.

El primer paso que se ha tomado en cuenta, es que al inicio de una canción en la primera nota existe un ataque lo que hace en ocasiones que el pitch en ese pequeño fragmento de nota varíe.

Para normalizar el registro de las notas de las canciones y librarse de la transposición se opto por lo siguiente:

Tomando en cuenta, que una canción o un fragmento de canción regularmente sigue una secuencia que se repite varias veces en una misma melodía. Podemos tomar como referencia la primer nota que es interpretada como nuestro cero o punto de referencia. Para así tener distancias semejantes en cualquiera que sea el tono que este la canción.

Por ejemplo, si una canción empieza por A4 y tenemos esta secuencia, A4#, A4, A4, A4, A4. Observamos que nuestra secuencia empieza con una variación de un semitono. Así que, si tomáramos como referencia dicha nota estaría incorrecta nuestra información, por lo que debemos tomar una ventana de los primeros valores de la secuencia de la canción.

A4#, A4, A4, A4, A4. Para nuestro caso hemos tomado las primeras 5 muestras, ya que al tomar estas tendremos la certeza de que se empieza con el valor adecuado. Así que si asignamos un valor arbitrario a A4 =1. Entonces nuestra secuencia seria la siguiente:

$1\frac{1}{2}$, 1, 1, 1, 1, recordando que la distancia que existe entre A4 y A4# es de $\frac{1}{2}$ .Para tomar la referencia adecuada tomamos la moda de los valores que se exponen en la secuencia, dado que es valor que tiene la mayor frecuencia absoluta al inicio de nuestra secuencia; y el que deseamos tener como marco de referencia de nuestra canción.

Una vez aplicada la moda a nuestra secuencia queda de la siguiente manera:

1,1,1,1,1 : A4,A4,A4,A4,A4 como se observa ya la secuencia de inicio es uniforme, podemos tomarla como referencia así que ampliemos la secuencia de la siguiente manera tomando para este ejemplo solamente 1 octava (C4 - C5). Asignando valores como se observa en la tabla 3.2.

Nota	Valor
C4	1
C4#	2
D4	3
D4#	4
E4	5
F4	6
F4#	7
G4	8
G4#	9
A4	10
A4#	11
B4	12
C5	13

Tabla 3.2. Ejemplo de asignación de valores para la normalización

Nótese que los valores son arbitrarios, pero respetan la distancia en tonalidad que existe entre ellos. Para poder visualizar el marco de referencia se pondrá en la figura 3.10 un ejemplo visual para su mejor comprensión.

Siguiendo con el ejemplo anterior ampliamos la secuencia como sigue.

A4,A4,A4,A4,A4, B4,B4,B4,B4,F4#,F4#,C5 de manera que utilizando los valores de la tabla 3.2. Tenemos lo siguiente:

10, 10, 10, 10, 10, 12, 12, 12, 12, 7, 7, 13

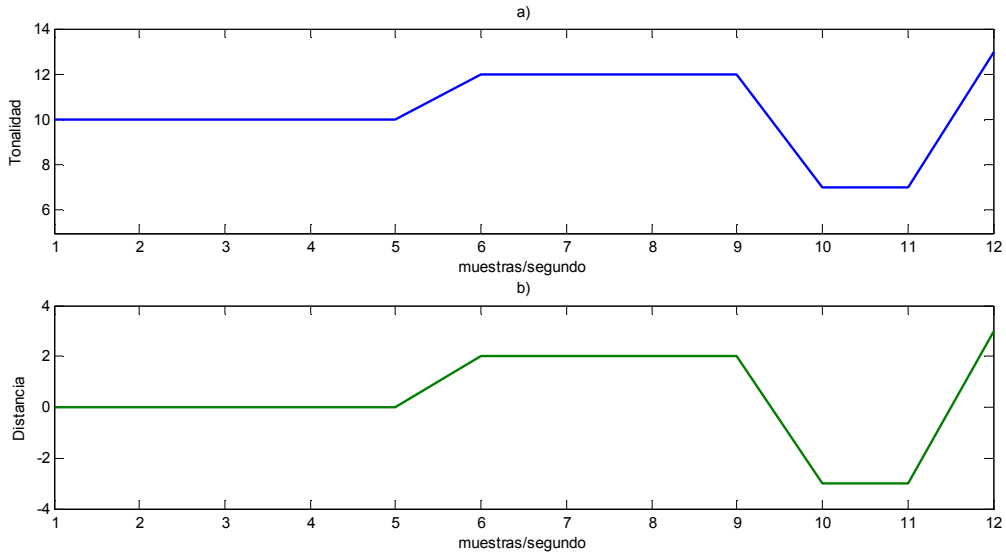


Figura 3.10 a) Distancias en valores arbitrarios sin normalizar, b) Valores de distancias tonales normalizadas.

Para obtener la normalización realizamos solo la siguiente operación.

$$Dn = (v_i, v_{i+1}, v_{i+2}, v_{i+3}, \ldots, v_{i+n}) - v_{im} \tag{3.8}$$

Donde *Dn* es el vector de distancias normalizadas, v_i es el valor inicial del vector y v_{im} es el valor de la moda obtenida de las primeras muestras.

Tomando en cuenta la operación para nuestra secuencia tenemos que.

$$Dn = (10-10), (10-10), (10-10), (10-10), (10-10), (12-10),$$

$$, (12-10), (12-10), (12-10), (7-10), (7-10), (13-10)$$

$$Dn = 0,0,0,0,0,2,2,2,2,-3,-3,3$$

Como consecuencia de esta operación, la secuencia para esta canción se a resumido a distancias en semitonos como puede observarse en la figura 3.10 b). Tiene exactamente la misma forma que la anterior, pero con el detalle que esta esta centrada en 0 esto quiere decir que no importa en que tonalidad se encuentre la música o el tarareo. Si la melodía es la misma tendrán un vector de distancias muy similar por lo que la transposición se ha anulado o en el peor de los casos se ha reducido.

Así para nuestro ejemplo el vector de distancias a quedado resumido a $Dn = 0,0,0,0,0,2,2,2,2,-3,-3,3$ esto es que en la primera parte existe una subida de tonalidad de 2 semitonos desde A4 a B4 posteriormente hay una bajada de 5 semitonos desde B4 a F4# y por ultimo una subida de 6 semitonos de F4# a C5 nótese que no se considera el medio tono de la nota en cuestión por lo que para un músico estos números

podrían estar incorrectos o confusos, pero solamente son una referencia para evitar la transposición.

3.4 Cálculo de distancias para reconocimiento de las canciones.

Hasta ahora, se ha obtenido una representación de las variaciones de una melodía dentro de una canción. Por medio de una normalización y cálculo de distancias basadas en la escala cromática, para evitar la transposición de las melodías. Ahora resta ver el método con el cual se encontrara las coincidencias dentro de la base de datos.

Los elementos dentro de la base de datos estarán previamente analizados, teniendo dentro de esta solamente los vectores de distancia tonal. Tomando en cuenta que el usuario mandara a este sistema una grabación de su voz tarareando la canción deseada. Será procesada para obtener su vector de descripción y se hará la comparación con elementos de la base de datos.

Para tener una comparación dentro de la base de datos encontraremos de nuevo distancias. Pero en esta ocasión entre los dos vectores que están involucrados en el proceso; es decir, el usuario y los datos almacenados. Como puede figurarse, los vectores de distancia de una melodía ya sea que tenga variación tonal o no, se verán de manera similar.

Para hacer el cálculo de distancias se ha tomado en cuenta dos procesos. Esto es debido al tiempo de computación que lleva realizar uno de estos. El primero consiste en un cálculo de distancia, sin tomar en cuenta las variaciones que puede existir en tiempo de las canciones. Por ejemplo puede que una canción tenga las mismas notas, pero la duración de sus notas no sea la misma entre si.

El segundo proceso que se ha tomado en cuenta es el llamado Alineamiento Temporal Dinámico (Dynamic Time Warping) DTW de sus siglas en ingles. Que básicamente encuentra la distancia entre dos secuencias (vectores), encontrando las similitudes entre estas y calculando la distancia mas corta que existe entre estas dos. Con este proceso podemos tratar los casos en que las variaciones de tiempo o ritmo son considerables. Dado que la mayoría de los usuarios no serán cantantes experimentados o músicos; habrá muchos casos en los que este tipo de problema aparezca, claro que se pago con un tiempo de computación mucho mas largo que con el primero, pero en ciertas ocasiones mas confiable.

3.4.1 Cálculo de distancia directa entre secuencias.

Dado que siempre existe la posibilidad de que la melodía que canta o tararea un usuario puede ser aproximadamente igual, a la que existe en la base de datos figura 3.11. Ya sea en tonalidades y en tiempo, se a recurrido a este proceso de calculo directo. Que salva mucho el tiempo de cálculo que se tiene que realizar con cualquier otro método debido a la sencillez del mismo.

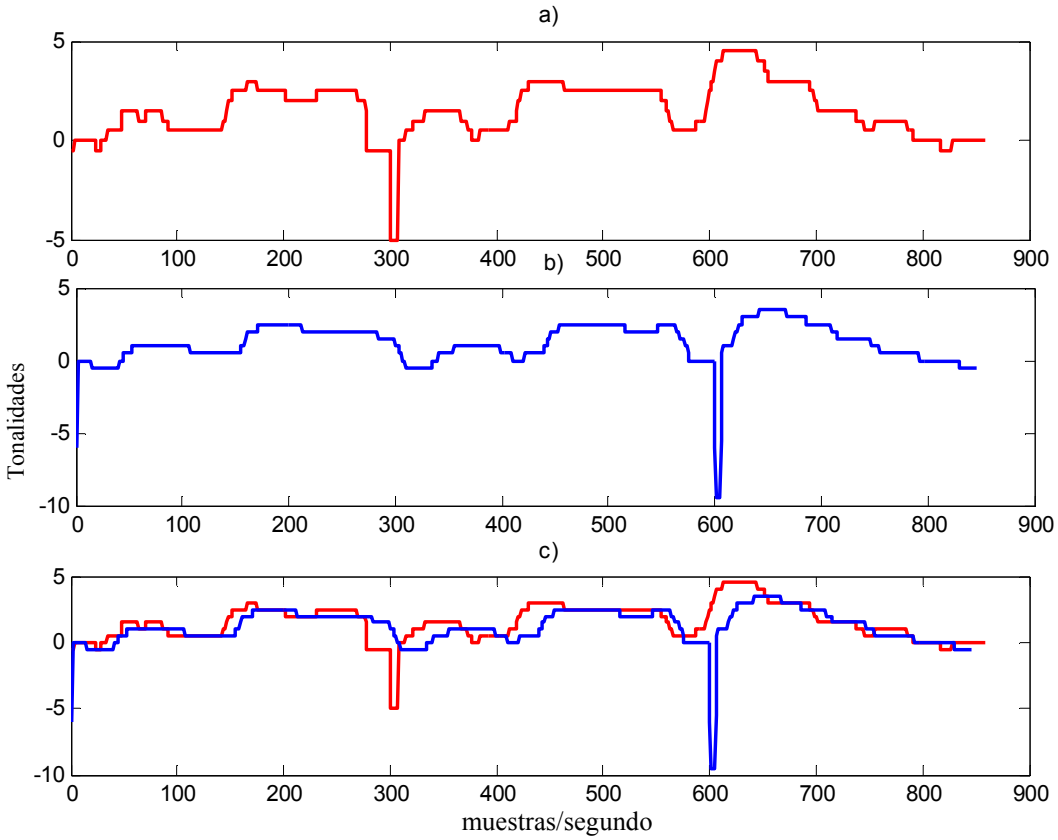


Figura 3.11.a) Secuencia en base de datos; b) Secuencia tarareada por el usuario; c) Comparación de las dos secuencias para visualizar sus distancias.

Existen un número indefinidamente grande de canciones en el mundo. Por lo que una base de datos para este tipo de aplicaciones se extiende a millones. Aunque siempre la música puede ser dividida en géneros, si es interpretada por una mujer o un hombre, o hasta por la región geográfica de la cual es el artista. De esta manera ser más específico el tipo de canción a buscar. Aun de esta manera, existen un número muy grande de canciones a tomar en consideración.

Es de mucha importancia que el sistema no demore demasiado en dar una respuesta al usuario. Ya que normalmente quiere esa información en un momento determinado, y la respuesta no puede tomar días; sino el propósito de la aplicación será insignificante. Por lo que entre menor tiempo de respuesta el sistema mayor interés tendrá usuario en utilizarlo.

Por lo tanto, el método consiste en dado dos secuencias A(n) y B(n), encontraremos sus diferencias punto a punto. Es decir elemento a elemento, para obtener un nuevo vector que indicara las distancias de cada uno de los puntos (nótese que no se utiliza la distancia euclidiana) entonces:

$$D(n) = |A(n) - B(n)|; \tag{3.9}$$

Donde D es el nuevo vector de distancias directas figura 3.12, A es una secuencia de notas normalizada de la base de datos, B es el vector normalizado del usuario y $n \in \mathbb{R}$.

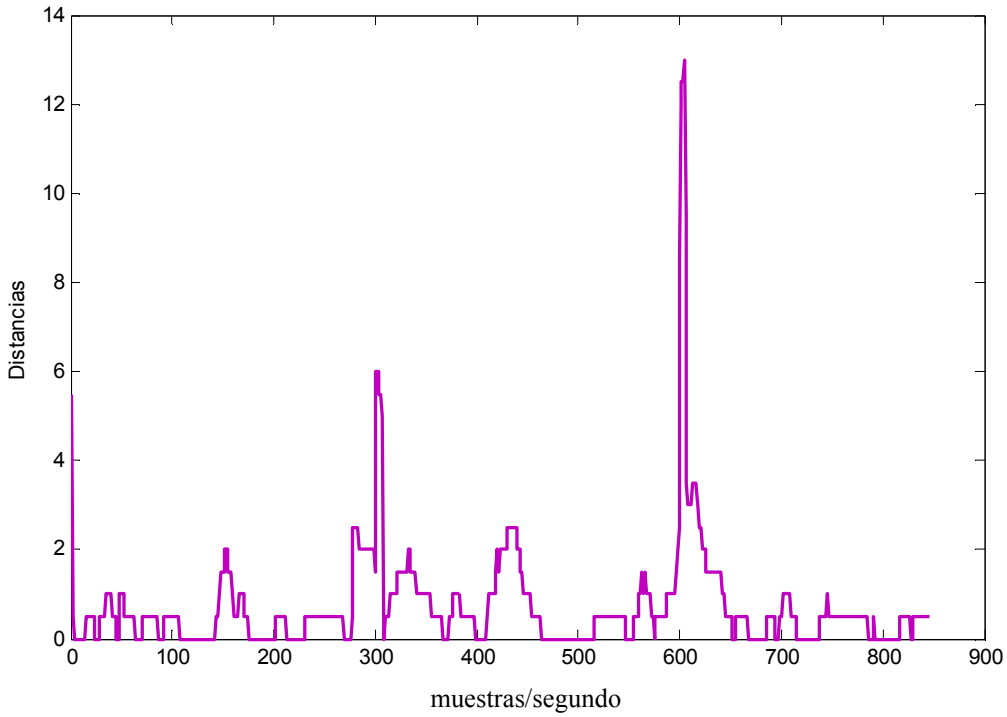


Figura 3.12 Vector de distancias punto a punto

Al tener este nuevo vector de distancias, podemos calcular un promedio de los elementos obtenidos. Si el promedio de las distancias es cercano a cero, quiere decir que las secuencias coinciden y por ende esa es la canción que el usuario esta buscando. En caso contrario, en que la distancia sea demasiado grande; quiere decir, que la canción no es la misma, o que las notas no corresponden a la misma canción. Es evidente que puede darse el caso de error, que en realidad se trate de la canción pero tiene diferencias muy grandes en el tiempo de sus notas.

Así que para calcular el promedio de distancias *Dp* tenemos que:

$$Dp = \frac{1}{n}\sum_{i=1}^{n} D(n)$$

(3.10)

En la figura 3.12 se muestra el vector resultante de la operación. Así que para este ejemplo, dado que se calcula el promedio de las distancias para obtener un factor de similitud entre ambas secuencias. Dando como resultado:

$$Dp = \frac{1}{n}\sum_{i=1}^{n} D(n) = 0.7429$$

Como se observa podría decirse que la melodía tiene variaciones de 0.7 semitonos que se traduciría a una variación de menos de un semitono aproximadamente. Por lo que es muy cercana la melodía, esto quiere decir que las canciones son muy similares. Por lo tanto se podría considerar como un candidato a ser la búsqueda requerida, ahora si observamos el comparativo realizando la misma operación para canciones distintas observe la figura 3.13.

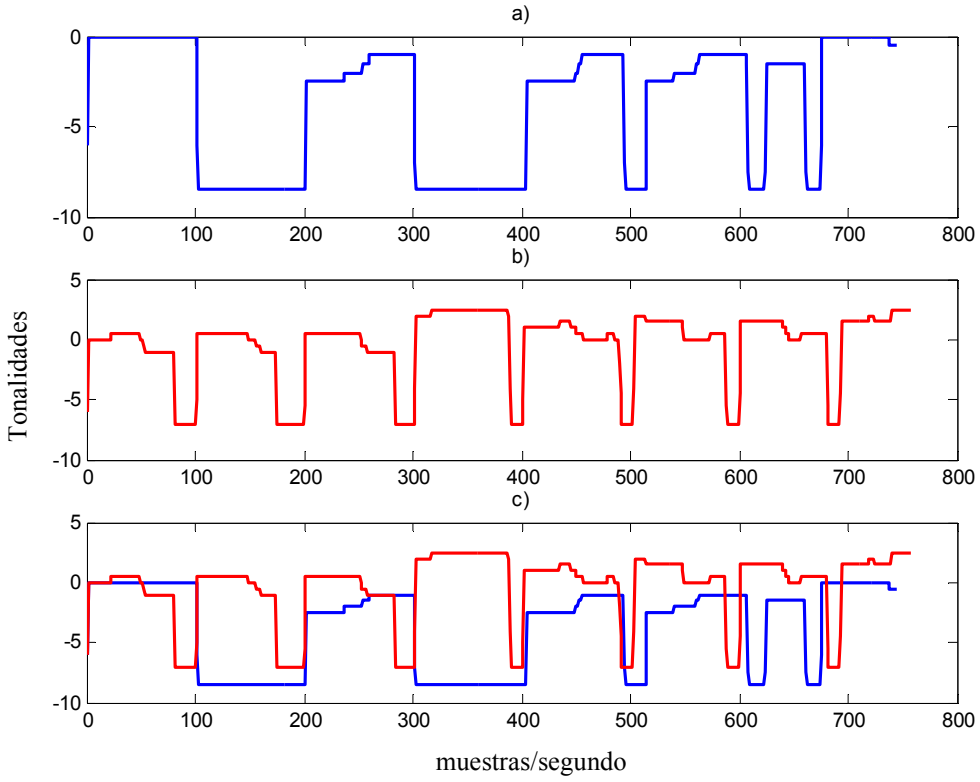


Figura 3.13. a) Tarareo del usuario, b) Canción BD que no coincide, c) Visualización de coincidencia

Entonces aplicando de la misma manera a estos vectores, la ecuación 3.9 obtenemos el siguiente vector de distancias.

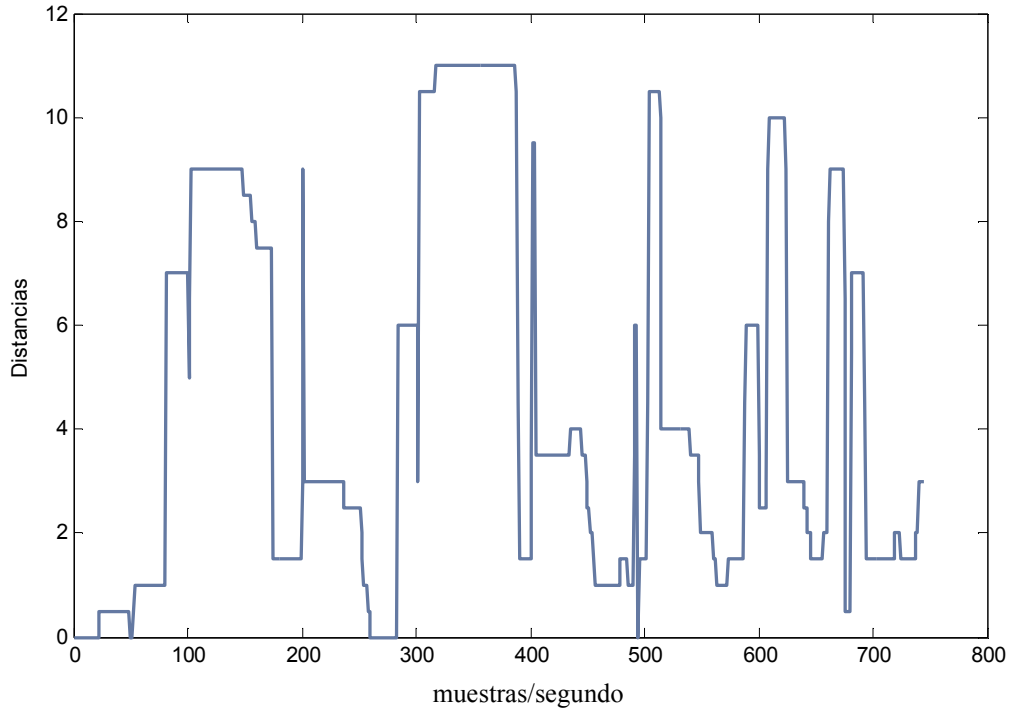


Figura 3.14 Vector de distancias de canciones diferentes

Como se observa en la figura 3.14, el vector resultante tiene muchas alteraciones es decir las distancias son muy grandes en diversos puntos, por cual al aplicar (3.10) tenemos los siguiente.

$$Dp = \frac{1}{n}\sum_{i=1}^{n} D(n) = 4.4556$$

Como se observa, la variación es muy grande ya que en promedio esta entre mas 4 semitonos. Lo que quiere decir que el nivel de coincidencia es muy bajo, por lo tanto se estaría descartando en la base de datos como candidato a la canción en búsqueda.

Por lo tanto este método de comparación es muy efectivo, tomando en cuenta un caso ideal en la cual las variaciones en tiempo son pequeñas. Pero cuando existen demasiadas variaciones en tiempo puede haber diversos errores, por lo cual se debe de tomar un algoritmo más robusto para la búsqueda de la canción.

3.4.2 Calculo de distancia por Dynamic Time Warping (DTW).

El algoritmo conocido como DTW, [15] es utilizado para diversos propósitos y la idea de este algoritmo es simple. Ya que este se encarga de encontrar la secuencia más similar dentro de una búsqueda en una base de datos. La secuencia en la búsqueda, puede ser ligeramente diferente de la secuencia original ubicada en la base de datos. Y es especialmente útil para este propósito, ya que los usuarios tendrán variaciones significativas al interpretar las melodías; será normal tener errores de tiempo y ritmo.

Como observamos en la figura 3.15, existen dos secuencias de canciones muy similares entre si. Pero con el detalle de tener ciertas variaciones en ritmo y tiempo, observamos que si efectuamos nuestro procedimiento anterior, obtendremos distancias largas debido a que al empezar de diferente manera las secuencias apenas coinciden.

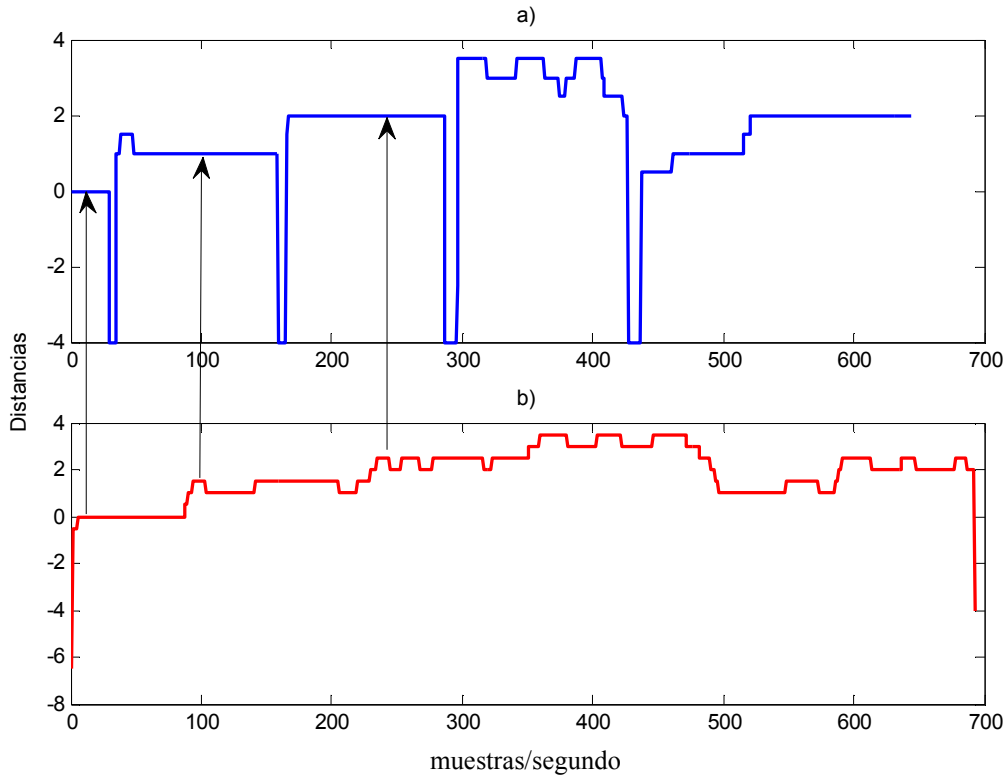


Figura 3.15 Secuencias la misma canción con variaciones en tempo y ritmo.

Las variaciones en ritmo y tiempo usando el método de la distancia directa, van acumulando errores conforme la secuencia va avanzando. Haciendo imposible un reconocimiento exitoso, y desechando este candidato como posible respuesta aunque en realidad se trate de la misma canción.

Usando el algoritmo DTW, podemos evitar que estas variaciones que van apareciendo se vayan minimizando para llegar a tener una coincidencia entre las dos secuencias.

Dado que las secuencias involucradas tienen similitudes, están llegaran a tener una distancia corta de relación de una con otra. Sin embargo si presentamos secuencias de canciones distintas, estas tendrán una distancia muy grande entre si descartándolas como candidatos en la búsqueda.

Como hemos visto en la figura 3.15, las dos secuencias tienen similitudes claras aunque no son iguales. La idea principal de DTW es encontrar el camino entre las dos secuencias minimizando la distancia. La elección de la ruta corresponde, a la elección de que elemento de la primera secuencia será comparado con el elemento de la segunda. Lo ideal de este método, es que los elementos de las secuencias sean comparados entre si, de tal manera que nos dirijamos a una distancia de similitud de cero. Aunque estas comparaciones, no pueden hacerse al azar deben de haber reglas, o restricciones en cuanto se refiere a cual conjunto de elementos puedes comparar con los otros de la otra secuencia.

En un principio, se tienen que usar dos matrices. La primera guardara los valores de distancias y la segunda se utiliza para almacenar la distancia de similitud presente hasta ahora.

Primero calculamos la matriz de distancias *d*, cuyos elementos *d [i, j]* es la distancia entre los elementos de *i* correspondientes a la primera secuencia y *j* correspondientes a la segunda. Como en el cálculo de la distancia anterior no emplearemos el uso de la distancia euclidiana, sino la diferencia en valor absoluto de los dos elementos como en la ecuación 3.9.

Considerando dos secuencias basados en la tabla 3.2.

secuencia1 = 4 4 4 6 6 8 8 8, consideraremos esta secuencia como la secuencia original almacenada en la base de datos.

secuencia2 = 2 2 4 4 4 6 6 8, esta secuencia será la que ha sido generado a partir del tarareo del usuario tomaremos estas matrices pequeños para mostrar un ejemplo claro.

Entonces calculando la matriz *d,* para poder obtener la matriz D. La primera fila y la primera columna son la secuencia1 y secuencia2 respectivamente.

$$d = \begin{bmatrix} - & \mathbf{4} & \mathbf{4} & \mathbf{4} & \mathbf{6} & \mathbf{6} & \mathbf{8} & \mathbf{8} & \mathbf{8} \\ \mathbf{2} & 2 & 2 & 2 & 4 & 4 & 6 & 6 & 6 \\ \mathbf{2} & 2 & 2 & 2 & 4 & 4 & 6 & 6 & 6 \\ \mathbf{4} & 0 & 0 & 0 & 2 & 2 & 4 & 4 & 4 \\ \mathbf{4} & 0 & 0 & 0 & 2 & 2 & 4 & 4 & 4 \\ \mathbf{4} & 0 & 0 & 0 & 2 & 2 & 4 & 4 & 4 \\ \mathbf{6} & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 \\ \mathbf{6} & 2 & 2 & 2 & 0 & 0 & 2 & 2 & 2 \\ \mathbf{8} & 4 & 4 & 4 & 2 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Hay diferentes maneras de inicializar la matriz D de DTW. Esta matriz tiene la misma forma y tamaño que la primera, la forma más básica para inicializar es establecer la primera columna y la primera fila como la suma acumulativa de los elementos de la matriz de distancia, dándonos.

$$D = \begin{bmatrix} & 4 & 4 & 4 & 6 & 6 & 8 & 8 & 8 \\ 2 & 2 & 4 & 6 & 10 & 14 & 20 & 26 & 32 \\ 2 & 4 & & & & & & & \\ 4 & 4 & & & & & & & \\ 4 & 4 & & & & & & & \\ 4 & 4 & & & & & & & \\ 6 & 6 & & & & & & & \\ 6 & 8 & & & & & & & \\ 8 & 12 & & & & & & & \end{bmatrix}$$

Para llenar el resto de la matriz, puede resultar un poco parecido al algoritmo de Viterbi para calcular la mejor distancia hasta que la matriz completa. Después de esto, mostrar el último elemento. Para poder calcular la distancia se tiene que definir el posible conjunto de opciones cuando consideramos un elemento.

Si consideramos que n_1 sea longitud de la secuencia 1 y n_2 sea la longitud de la secuencia 2. Si observamos el algoritmo tenemos que.

Para i desde 2 hasta n_1

- para j desde 2 hasta n_2

1.- Escoger $(k, l) \in conjunto_{(i,j)}$ tal que $D(k, l)$ es mínimo

2.- $D(i, j) = D(k, l) + d(i, j)$

El conjunto de $conjunto_{(i,j)}$ depende de *(i, j)* y puede ser por ejemplo:

$S_1 = \{(i-1, j-1); (i-1, j); (i, j-1)\}$ ó

$S_2 = \{(i-2, j-1); (i-1, j-2); (i-1, j-1); (i-1, j); (i, j-1)\}$

En el proceso del algoritmo, escogemos el predecesor que tenga el mínimo valor. En la figura 3.16. Podemos visualizar el funcionamiento del algoritmo.

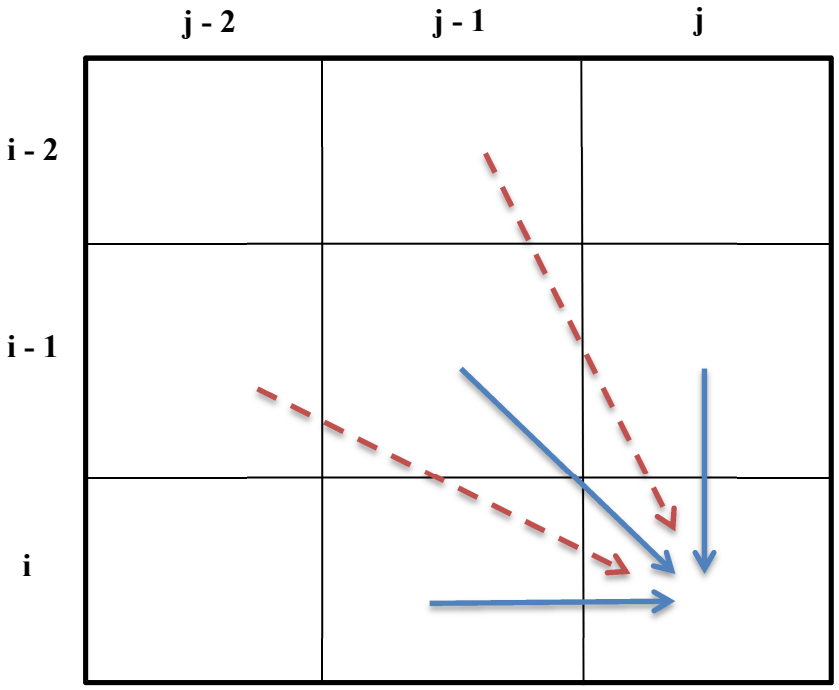


Figura 3.16 Conjuntos del algoritmo DTW, en las flechas azul continuo: S_1 y en las flechas rojo discontinuo: S_2

Hay que tener en cuenta que si $(k, l) \in conjunto_{(i,j)}$, entonces debemos tener $k \leq i$ y $l \leq j$, y (i, j) no esta en el $conjunto_{(i.j)}$ esto con finalidad de no ir hacia atrás.

Entonces de lo visto podemos visualizar todo el algoritmo:

1.- Calculamos la distancia de la matriz d: $d(i,j) = d\big(sec_1(i), sec_2(j)\big), para\ (i,j) \in [1, n_1]x[1, n_2]$

2.- Inicializamos la matriz D:

$$D(1,1) = d(1,1)$$

$$para\ i \in [2, n_1]\ D(i,1) = D(i-1,1) + d(i,1)$$

$$para\ j \in [2, n_2], D(1,j) = D(1,j-1) + d(1,j)$$

3.-Llenamos la matriz D:

Para i desde 2 hasta n_1

Para j desde 2 hasta n_2

(a) escoger $(k,l) \in conjunto_{(i,j)}\ \ tal\ que\ D(k,l) sea\ minimo$
(b) $D(i,j) = D(k,l) + d(i,j)$

4. Regresar $D(n_1, n_2)$

Entonces al regresar a nuestro ejemplo podemos llenar la matriz como a continuación.

$$D = \begin{bmatrix} & 4 & 4 & 4 & 6 & 6 & 8 & 8 & 8 \\ 2 & 2 & 4 & 6 & 10 & 14 & 20 & 26 & 32 \\ 2 & 4 & 4 & 6 & 10 & 14 & 20 & 26 & 32 \\ 4 & 4 & 4 & 4 & 6 & 8 & 12 & 16 & 20 \\ 4 & 4 & 4 & 4 & 6 & 8 & 12 & 16 & 20 \\ 4 & 4 & 4 & 4 & 6 & 8 & 12 & 16 & 20 \\ 6 & 6 & 6 & 6 & 4 & 4 & 6 & 8 & 10 \\ 6 & 8 & 8 & 8 & 4 & 4 & 6 & 8 & 10 \\ 8 & 10 & 12 & 12 & 6 & 6 & 4 & 4 & 4 \end{bmatrix}$$

La distancia que se encuentra en el extremo inferior derecho de la matriz, será nuestra distancia calculada. Es decir que la distancia que existe entre las 2 secuencias es de 4. Para ver una comparación, pondremos un caso en la cual las secuencias son casi iguales. Si tenemos una secuencia s_3 = 4 4 4 6 8 8 8 8 tendremos que.

$$D = \begin{bmatrix} & 4 & 4 & 4 & 6 & 6 & 8 & 8 & 8 \\ 4 & 0 & 0 & 0 & 2 & 4 & 8 & 12 & 16 \\ 4 & 0 & 0 & 0 & 2 & 4 & 8 & 12 & 16 \\ 4 & 0 & 0 & 0 & 2 & 4 & 8 & 12 & 16 \\ 6 & 2 & 2 & 2 & 0 & 0 & 2 & 4 & 6 \\ 8 & 6 & 6 & 6 & 2 & 2 & 0 & 0 & 0 \\ 8 & 10 & 10 & 10 & 4 & 4 & 0 & 0 & 0 \\ 8 & 14 & 14 & 14 & 6 & 6 & 0 & 0 & 0 \\ 8 & 18 & 18 & 18 & 8 & 8 & 0 & 0 & 0 \end{bmatrix}$$

La distancia para este caso es cero, esto significa que para nuestra búsqueda la secuencia 1 y la secuencia 3 son mas similares que la secuencia 1 y 2; dado que su distancia es menor entre si.

Esta es la manera en que DTW funciona, en términos de tiempo de cálculo puede resultar más tardado que el anterior método. Pero puede ser conveniente tener en cuenta el resultado de ambos sistemas para mejor referencia en nuestra identificación de canciones.

3.5 Sistema de identificación polifónico (voz vs música).

El sistema de reconocimiento de canciones polifónico consiste en el análisis del audio de una canción en particular; poder contener sus parámetros más importantes y almacenarlos en vectores que servirán como descriptores de la canción para su posterior clasificación.

La idea de tener un sistema de este tipo se forma en base, a la necesidad de mejores descriptores para la música. En este caso la estructura musical misma de la canción. Así como en especial aquellas partes más significativas, que pueden llevar a un usuario a tararear esa parte en específico de la canción; y tener en cuenta la estructura completa de la misma. Ya sean notas que se encuentran, en una escala baja o notas que se encuentran, en una escala alta.

Como se ha estudiado en los capítulos anteriores, la voz humana cubre un cierto rango de frecuencias. Que en su normalidad una persona promedio, cubriría un rango de frecuencias aproximadamente dentro de los 110 Hz y los 440 Hz como frecuencia central. Que serian correspondientes a las notas A2 a A4 respectivamente, esto para resaltar que un usuario tararea normalmente aquellas partes de la canción que están dentro de sus posibilidades; con excepciones que serán tomadas en cuenta posteriormente.

Si tenemos un descriptor o descriptores más robustos, que puedan proporcionarnos diversas características de la canción, podemos tener una mejor probabilidad de éxito en la identificación. Y por ende nuestro sistema será mas efectivo, y a su vez será mas requerido por el usuario. Una canción en la cultura occidental esta compuestas por partes especificas o partes características de la canción. En la cultura popular, podemos describir una estructura genérica para diversas canciones como es: la introducción, verso, coro, verso, coro, coro etc. Esta claro que esto dependerá del artista; sin embargo, estas partes son las que regularmente recuerda una persona. Existen otras variaciones para darle un énfasis a la canción como son los puentes, los cambios de clave, los silencios, solos, etcétera.

Estas son partes importantes de la canción, pero regularmente las partes comúnmente recordadas o tarareadas son las primeramente mencionadas. Por lo cual son las partes que deben de tenerse en cuenta con mayor prioridad.

La estructura que adopta una canción es repetitiva, lo que hace que sea fácil de memorizar su melodía, e inclusive poder interpretar la canción con la voz de una manera muy cercana a la versión original. Tomando en cuenta que una canción tiene esta estructura y también que es necesario guardar el tiempo de procesamiento. La canción será partida en aquellas partes más importantes, para tener una base de datos más consistente para la identificación.

En el sistema anterior se planteó un descriptor en base a la obtención del pitch por el método de la autocorrelación. Lo cual funciona bien, cuando la música analizada es principalmente un solo instrumento, a lo que se refiere este trabajo como monofónico. Cuando una canción es simplemente interpretada de este modo es mucho más fácil de detectar sus partes importantes y tener la extracción de descriptores más fácilmente. Normalmente en una canción interactúan más de un instrumento, sin contar si la canción lleva voz; lo cual implicaría en si otro instrumento. Este conjunción de instrumentos en armonía figura 3.17 lo denominamos música y dado que esta conformado por diversos instrumentos (sonidos) nos referiremos a el como polifónico.

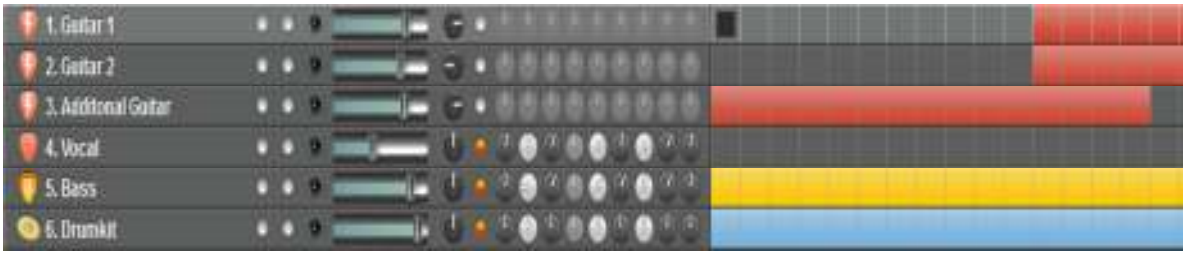


Figura 3.17 Ejemplo pista polifónica

Para realizar un barrido de la canción, extrayendo aquellas notas que la conforman como es evidente es extraer el pitch. Pero es complicado obtener el pitch simplemente con autocorrelación. Ya que esta, obtendrá un pitch calculado en la periodicidad de la onda generada por la música. Pero al tratarse de diversos instrumentos, el pitch obtenido simplemente será el resultado de la mezcla de frecuencias de los instrumentos que son ejecutados al mismo tiempo. Por lo que no resulta de mucha utilidad, obtener esto dado que al tratar de comparar la voz del usuario (monofónica), con la música (polifónica) estas simplemente no serán lo mismo en tiempo y/o en frecuencia; aun cuando se considere transposición de las melodías.

Es por eso, que se ha optado por utilizar un banco de filtros. Ya que en el sistema anterior, hemos conseguido asignar anchos de banda a cada una de las notas. Esto con la finalidad de aislar cada una de las frecuencias fundamentales de las notas, y obtener los tonos comprendidos en un principio desde 82.4 Hz hasta 1046.5Hz. Que comprenden las notas analizadas en el sistema anterior, de E2 hasta C6. En [25] se toman este rango de frecuencias, lo cual hemos tomado de igual forma para tener referencia y poder hacer comparación de resultados, aunque al final se ha optado por hacer ciertos ajustes como serán explicados mas adelante.

3.5.1 Asignación de bandas de banco de filtros.

La asignación de bandas para nuestro banco de filtros, es la misma que se ha planteado para el sistema anterior. Las bandas estarán determinadas por los parámetros especificados en la sección 3.2 de este capitulo, así como los datos utilizados estarán basados en la tabla 3.1.

Primeramente, para la construcción de nuestro banco de filtros se tomaron en cuenta filtros elípticos. Estos son los que mejor se adaptan a las necesidades de este trabajo, debido a su precisión marcada respecto a sus frecuencias de corte. Dado que tenemos que tener una certeza, de que las bandas de cada una de las notas están asignadas correctamente.

Basándose en diversas especificaciones que se exponen [15]. Como es el caso de los filtros elípticos con similares factores de calidad Q = 25. Asignamos las bandas según nuestras propias características, a diferencia de las proporcionadas en ese trabajo. Donde se descartan frecuencias que no cumplen con las especificaciones que muestran los filtros. Además de no tomar el número de octavas que tiene un piano, ya que este trabajo se centra en la voz humana y melodías que este pueda interpretar con facilidad.

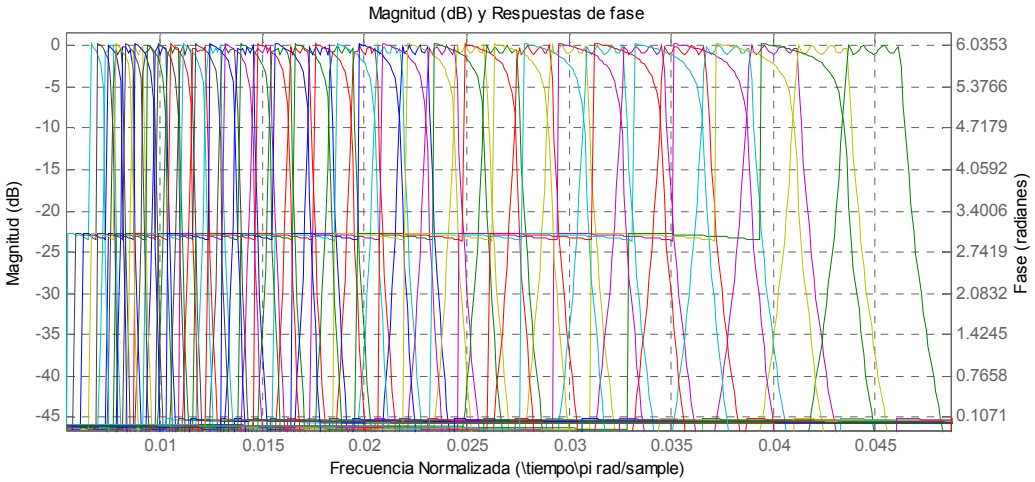


Figura 3.18 Distribución de bandas en el banco de filtros desde E2 a C6

Una de las características, que no se ha tomando en cuenta de igual manera aparte de eliminar octavas; es que se ha tomando únicamente una tasa se muestreo que equivale a 22050Hz. Debido a que el audio destinado para este sistema tiene esa tasa de muestreo. Dado que se trata completamente de música, se quiso conservar una buena calidad para no perder detalles en cuanto a las notas que se están interpretando en la canción. Y no la cantidad de octavas es menor que en [15], en el cual se dividen en diferentes tasas de muestreo.

Cada una de las bandas del filtro corresponde a cada una de las notas desde E2 a C6. Y cada una estas bandas concentra la energía que pasa por ella en un determinado momento. Si tomamos en cuenta, que las mayores magnitudes presentes en cada una de las bandas son

las notas predominantes en un periodo de tiempo. Realizamos un análisis por ventana tomando solamente aquellos puntos predominantes.

La ventana que se ha escogido es de 200 muestras equivalentes a 9ms. Esta presenta una resolución suficientemente buena para poder identificar notas relativamente cortas (con un tiempo reducido).

Entonces tomando en cuenta la ventana, se tomo el siguiente criterio para cada una de las bandas; agrupando cada banda en una sola matriz que será denominada como la huella digital de nuestra canción.

```
Mientras {i<=longitud de la canción-- cada una de las bandas

   Para {j=1 hasta el tamaño de ventana y hasta el tamaño de la canción
      Si {j>1
      j=j-1;
      }

   Muestras de análisis=matriz de bandas sin submuestreó(i , j:(ventana+j));
    Máximo=valor máximo de muestras de análisis;
      Matriz con bandas submuestreadas(i, k)=Máximo;
      k=k+1;
   }
   i=i+1;
   k=1;
}
```

Teniendo como se muestra los puntos máximos, de cada una de las ventanas analizadas de cada una de las bandas obtenemos la huella digital de la canción. La cual, puede ser visualizada y procesada con mayor facilidad que trabajar con la señal entera de la canción.

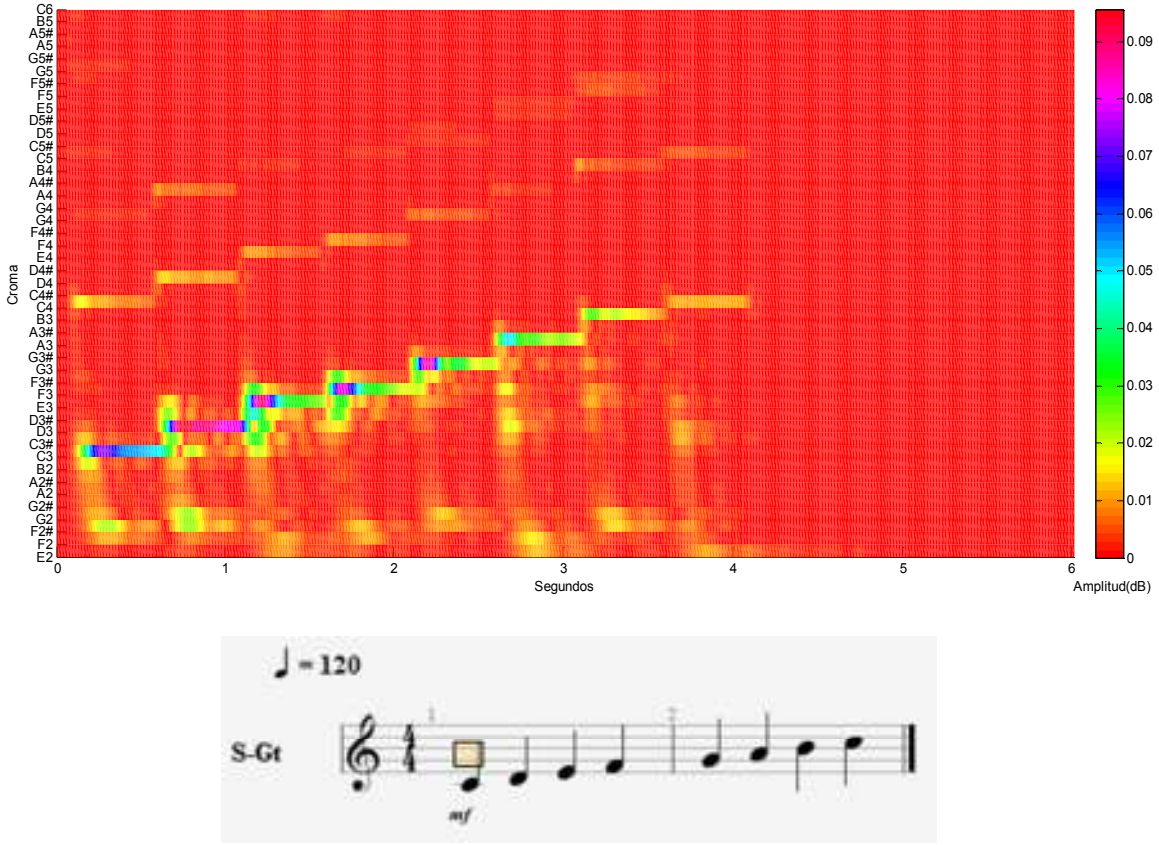


Figura 3.19 Huella digital y partitura de la escala de Do mayor desde C3 a C4 en guitarra

Podemos apreciar en la figura 3.19, la escala ascendente de Do mayor perfectamente marcada desde C3 hasta C4. Los cuales muestran ser las notas que tienen mayor energía en ese preciso momento. De igual manera podemos percibir que cada nota tiene una duración similar. Lo cual es perfectamente correcto si hacemos su comparación con la partitura de la ejecución.

Si introducimos una canción que tenga más de un solo instrumento como se observa en la figura 3.20. Observamos que el resultado, son las líneas más significativas del fragmento de la canción. Ya que para este sistema, se trabajara con fragmentos importantes de la canción y no con todo el archivo de audio, ya que la gran mayoría de la música occidental es redundante. Es preciso reservar recursos para el cálculo de las comparaciones dado que se tratan de millones de canciones; aunque para nuestro caso se ha evaluado para un número limitado de canciones por cuestiones de tiempo y capacidad de cómputo.

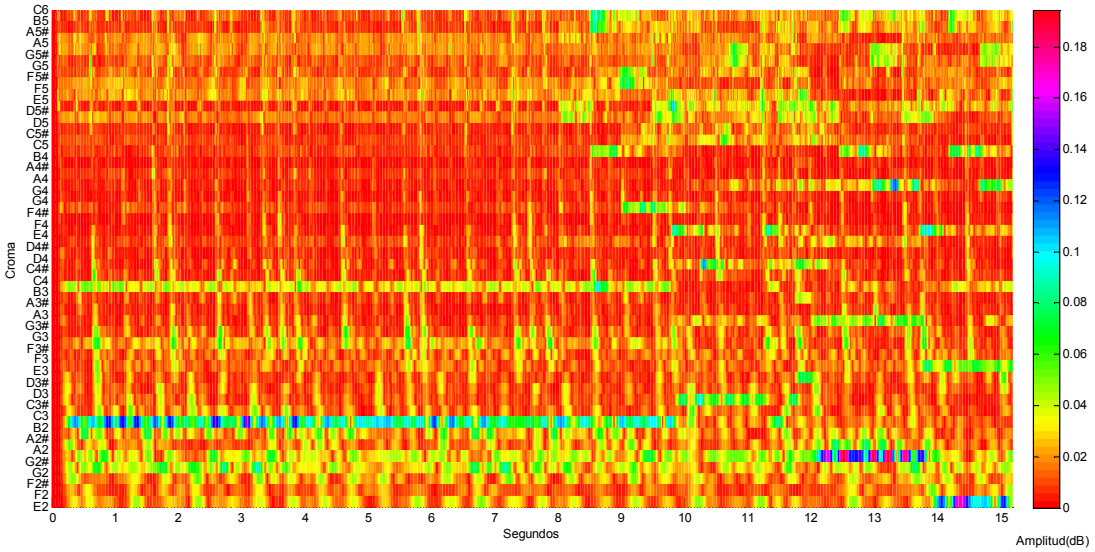


Figura 3.20 Huella digital de fragmento de la canción "Starlight"

A pesar de obtener las notas más significativas de la canción, existe un problema que en la mayoría de las canciones que se han analizado se presenta. Y es que las líneas ejecutadas por el bajo son demasiado altas, y normalmente salvo en casos excepcionales se tararea el bajo. Así que si analizamos las líneas principales del bajo, aparecen dentro de las notas comprendidas entre E2 a C3.

Si usamos el resultado obtenido en la figura 3.20 no obtendremos una respuesta adecuada. Si observamos la respuesta del tarareo de esta canción en la figura 3.21 tenemos resultados un tanto distintos de primera impresión.

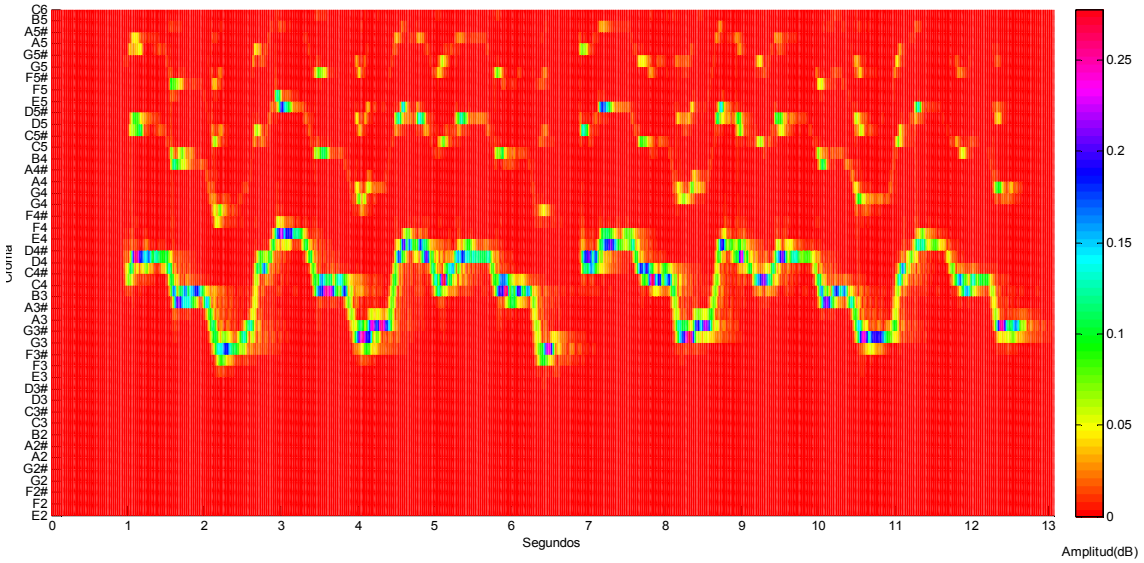


Figura 3.21 huella digital del tarareo de la canción "Starlight"

Si es comparado de simple vista las huellas digitales parecen tener muy pocas coincidencias, o no muy claras. Pero si tomamos en cuenta la eliminación de las líneas de bajo como se ha sugerido, solamente desde C3 hasta B4 podremos obtener una huella más clara que es mucho más similar a la mostrada en figura 3.20.

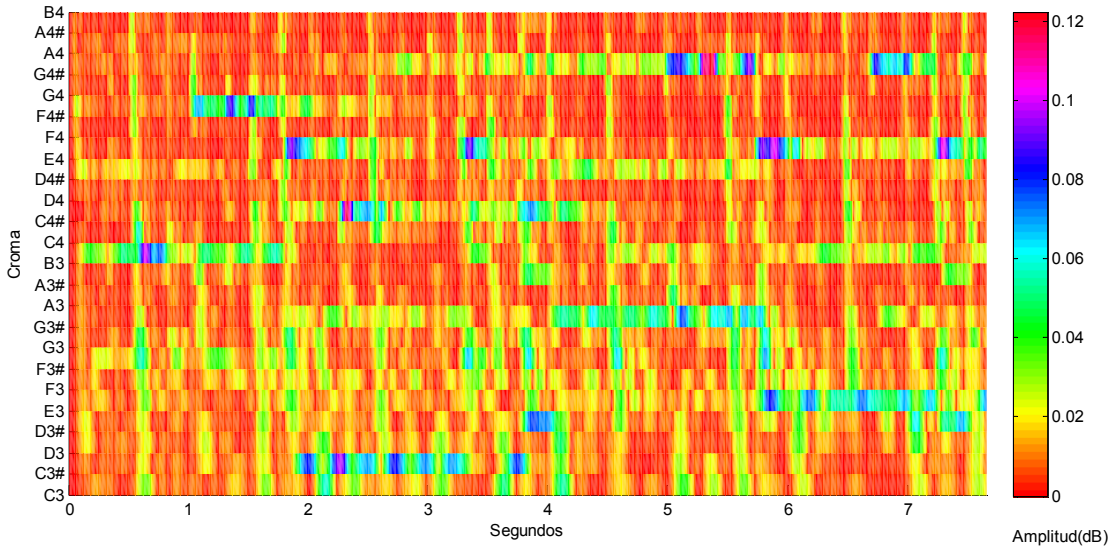


Figura 3.22 recorte de las líneas de bajo limitando a B4

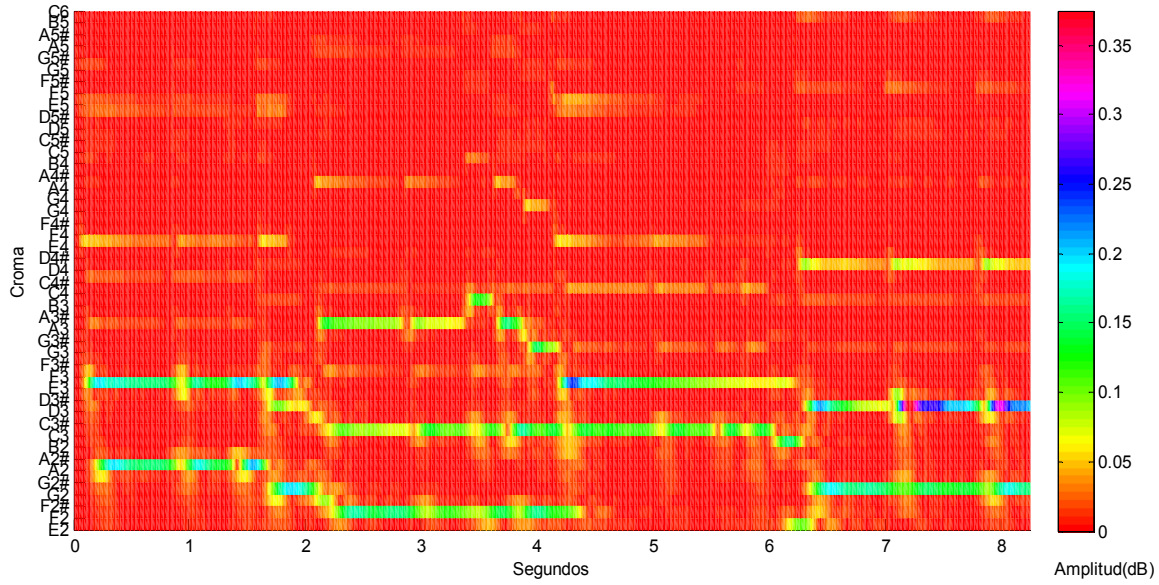
Ahora, podemos apreciar mayores puntos de similitud entre las dos huellas digitales. Si se observa bien aun aparecen líneas de bajo comprendidas desde C3 a G3# las cuales pueden ser eliminadas. Puesto que tenemos una referencia, ya sea por un tarareo a por la misma partitura de la canción que es analizada. Esto solamente en casos donde las coincidencias no sean del todo evidentes.

3.5.2 Obtención del vector de comparación desde la huella digital de la canción.

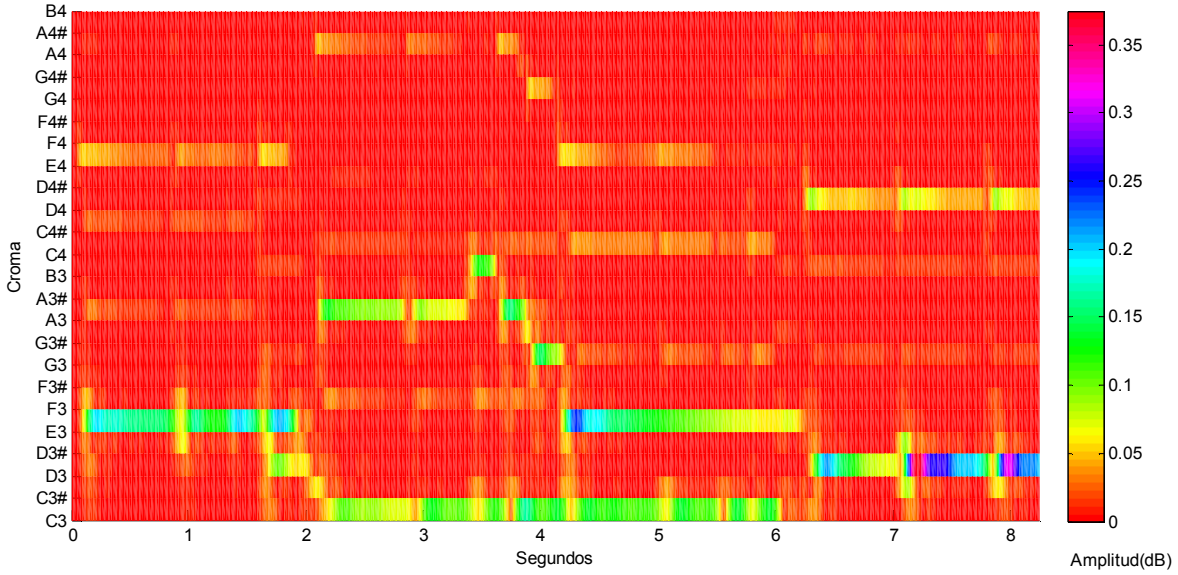
Obteniendo la huella digital de ambas partes, es preciso adquirir el descriptor que requerimos para hacer la identificación de las canciones. A pesar de que la huella digital de la canción por si sola es un buen descriptor; debido a que la dimensión de los datos que son requeridos para formarla es grande para hacer una comparación rápida y eficaz. Se ha optado por extraer los datos más relevantes de la huella digital, y formar un solo vector descriptor de la melodía que ha sido extraída de la huella digital.

Así siendo, que nuestra huella digital muestra los puntos de interés en un determinado momento de tiempo. Estos a su vez son los puntos máximos, podemos extraer estos puntos de acuerdo a que si tomamos los puntos máximos en cada muestra de la huella digital. Obtendremos un vector el cual será el descriptor requerido para nuestra búsqueda dentro de nuestra base de datos.

En la figura 3.23, podemos observar la extracción de la huella digital de una parte relevante de una canción. Para este ejemplo mostramos una canción la cual es fácil de identificar a simple vista, con la finalidad de hacer mas claro el proceso de obtención del descriptor.



a)



b)

Figura 3.23. a) Extracción de huella digital de un fragmento principal de la canción "Otherside". b) Huella digital omitiendo líneas de bajo.

Una vez teniendo de manera adecuada la huella digital, obtenemos nuestro descriptor de la siguiente manera:

```
[~, Columnas]=tamaño de huella digital;
Para {i=1: Columnas
   Columna (i)
   x=máximo de Columna (i)
   y=posición del máximo de Columna (i)

 Si {x==0
   y=0;
   }

   Vector descriptor (i) = y;

}
```

De esta manera, obtendremos el vector que se muestra en la figura 3.24 a partir de la huella digital obtenida en la figura 3.23.

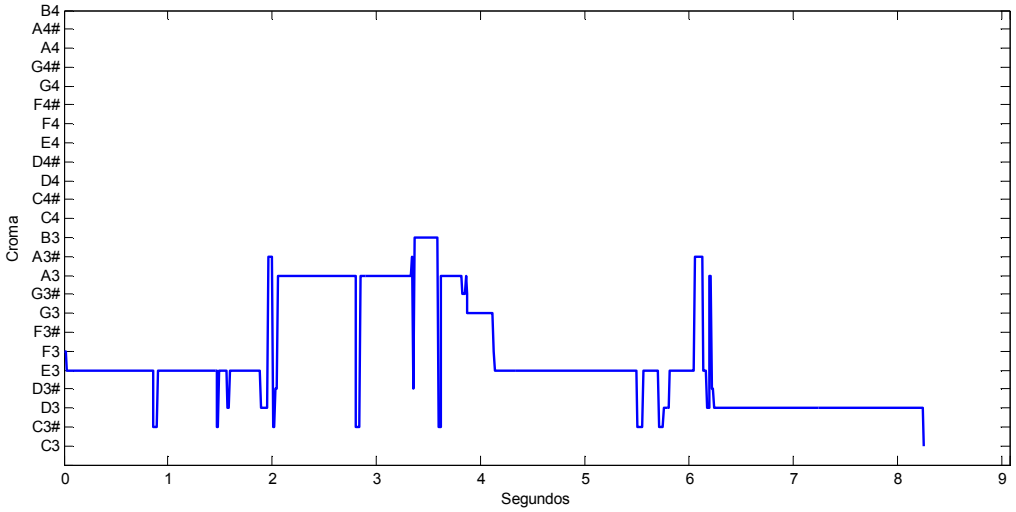


Figura 3.24 Vector descriptor obtenido de la huella digital

En la figura 3.24 se observa el vector de nuestro descriptor, el cual se utilizara para hacer la consulta en nuestra base de datos. Un detalle que puede ser observado, es que el vector presenta ruido en forma de variaciones súbitas de la tonalidad por cortos periodos de tiempo. Por lo cual se opto por aplicar un filtro de manera similar al ya efectuado en el sistema monofónico.

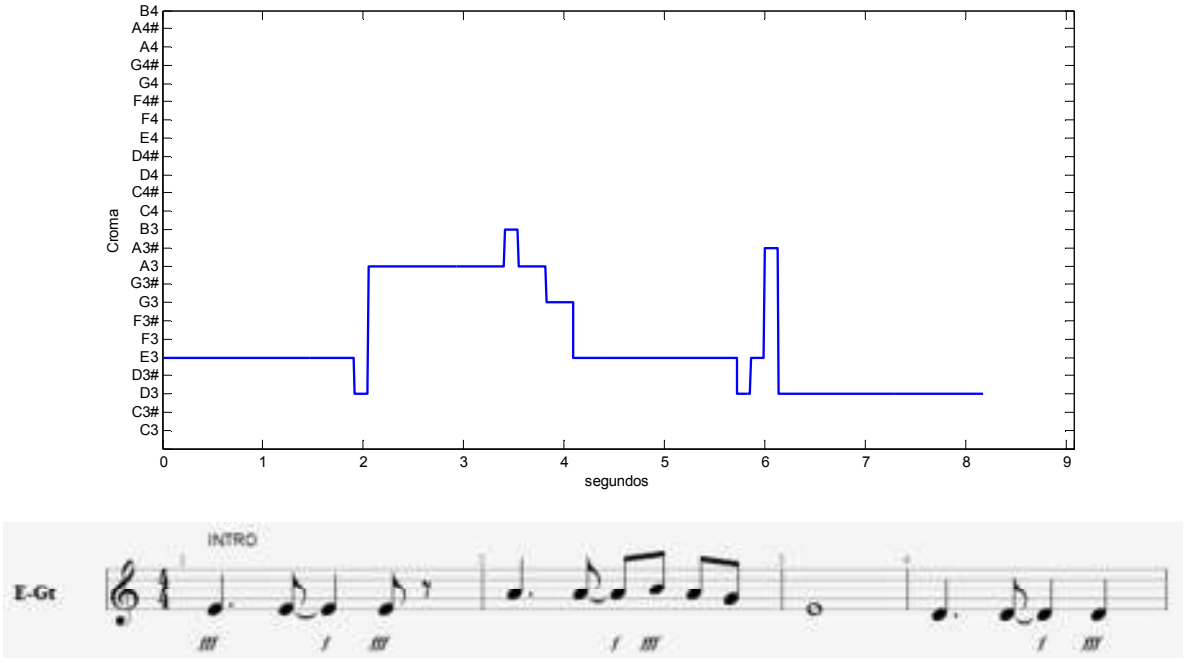


Figura 3.25 Descriptor de melodía sin ruido y partitura del mismo fragmento.

Mostrando en la figura 3.25 un descriptor sin ruido, podemos tener la certeza de que las notas que están siendo capturadas sean lo mas fiables posibles para su comparación con los tarareos de los usuarios. También, si se compara las notas que se encuentran en la partitura y las obtenidas en nuestro descriptor coincide perfectamente; lo cual nos indica que se ha extraído exitosamente la melodía de la canción. Por tanto podemos almacenarla en nuestra base de datos teniendo en mente de igual manera la transposición expuesta ya anteriormente.

Ahora de manera similar obtenemos la huella digital por este método para el tarareo del usuario, para así tener los dos elementos a comparar. En la figura 3.26 se observa la obtención de dicha huella, la cual puede compararse totalmente con la huella obtenida de la canción. Cabe notarse que la voz se encuentra entre el rango de frecuencia de C3 en adelante. Haciendo varias pruebas con distintas voces, se ha llegado a la conclusión de que no es necesario analizar la voz por debajo de C3 ya que son demasiado bajas; salvo en casos muy raros un usuario tarareara por debajo de este rango. Considerando los casos raros el algoritmo es capaz de reconocer si hay energías considerables por debajo de C3, para no perder datos de aquellas notas por debajo de nuestro rango de notas establecido.

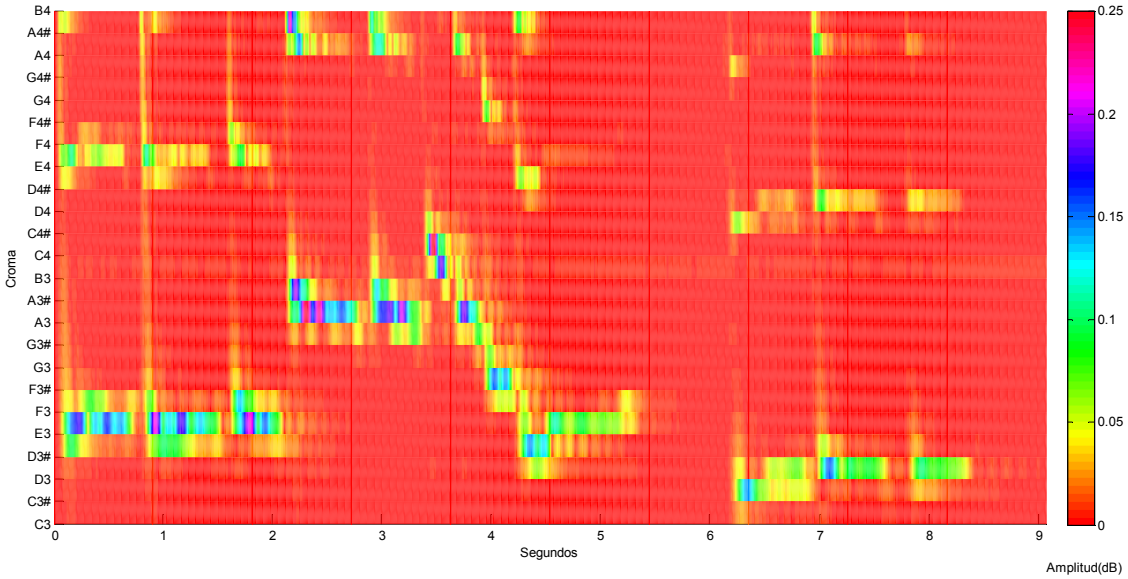


Figura 3.26 huella digital del tarareo de la canción "Otherside"

El tarareo presentado en la figura 3.26 podría considerarse como ideal. Ya que este es casi preciso en la sucesión de notas de la melodía. Así como la duración de esta, lo cual en resumen nos proporciona una buena probabilidad de éxito en la identificación de la canción. Con esto podríamos considerar que entre mejor sea el tarareo del usuario, mas acertado será la búsqueda y por lo tanto las probabilidades de éxito se elevan en estos casos.

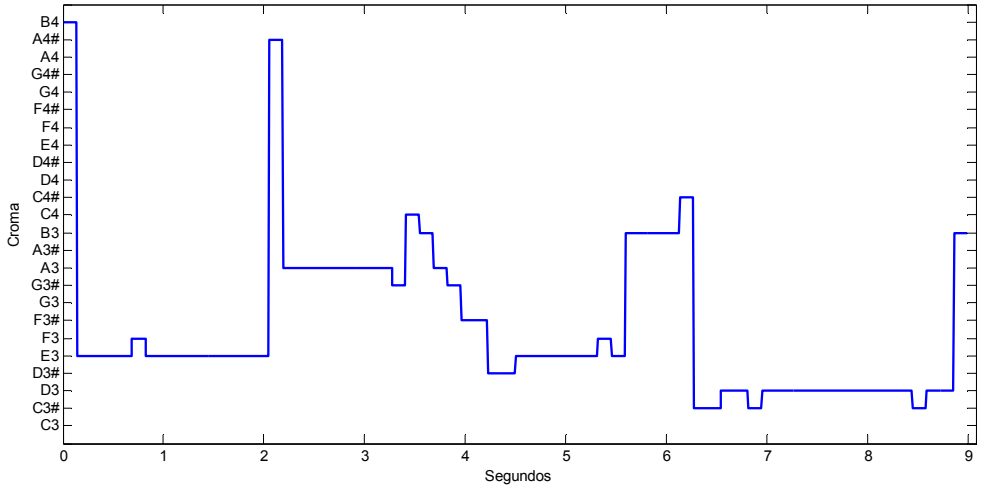


Figura 3.27 Vector descriptor de la huella digital del tarareo

Despreciando los picos de ruido que se han formado, aun usando los filtros es evidente que el vector obtenido es idéntico. Y puede compararse perfectamente con el obtenido de la canción original con los métodos que hemos estando utilizando. Sin embargo aun falta tomar en cuenta la transformación de este vector de descripción a el vector de no transposición; ya que como es sabido los tarareos serán en diferentes tonalidades de inicio y secuencia, por ende tendremos que seguir utilizando la eliminación de la transposición.

En ocasiones al obtener el vector descriptor de la huella digital, este presenta saltos en cortos periodos como se muestra en figura 3.28. Lo cual es causado porque se presentan armónicos que son más altos en potencia que el fundamental lo cual provoca variaciones de este tipo.

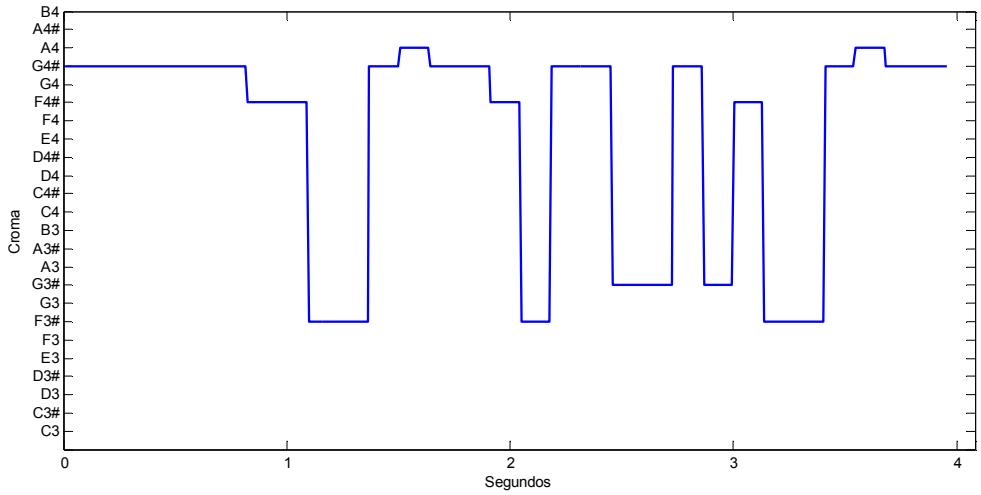
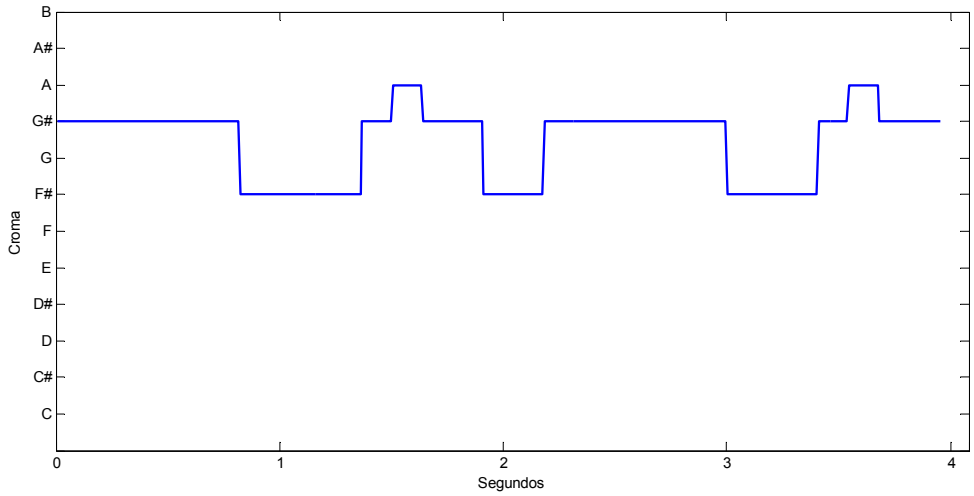


Figura 3.28 Ejemplo de problema de armónicos ("The man who sold the world").

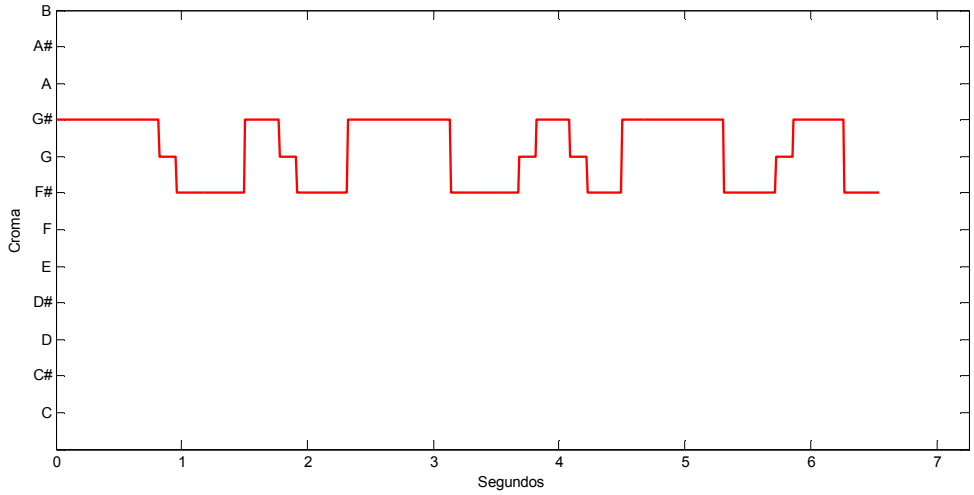
Con la finalidad de eliminar estos saltos y tener una mejor referencia, se decidió que las melodías obtenidas deberán ser normalizadas a una sola octava. Así no importando que haya variaciones de este tipo en la melodía, la secuencia de las notas no se vera alterada y por consecuencia tendrá una mejor probabilidad de éxito en la búsqueda.

Ahora teniendo en cuenta el resultado obtenido, usando una sola octava para el ejemplo de la figura 3.28 en la figura 3.29 obtenemos una forma mejor definida de la melodía si la comparamos con su contraparte tarareada.

Este es un ejemplo que es muy comúnmente encontrado en este sistema, y en otros previamente utilizados [25] por lo que se concluye que puede ser de gran utilidad hacer esta transformación de la melodía para precisar de manera efectiva la búsqueda.



a)



b)

Figura 3.29. a) Normalización a una octava de la figura 2.28 ("The man who sold the world"). b) Descriptor del tarareo obtenido siguiendo el mismo procedimiento.

3.6 Representación del pitch en coeficientes de Hadamard.

Como se ha observado en los subtemas anteriormente expuestos, el pitch o tonalidad de una canción puede ser representada por una señal cuadrada con variaciones en el tiempo, ya sea periódica o no figura 3.30. Al obtener este tipo de señal con estas formas nos indica que puede ser resumida o comprimida usando una transformada adecuada, que nos proporcione una reconstrucción de la señal con menos datos de los adquiridos.

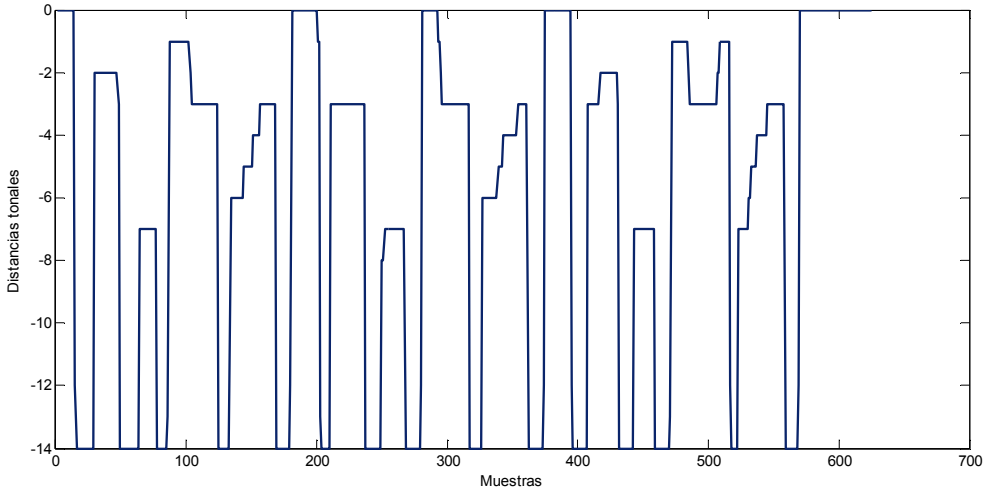


Figura 3.30 Secuencia de tonalidades de una canción

Es evidente que al tener menos datos que describan nuestra canción y aun así tener una identificación sin degradación, el proceso será aun más rápido y en consecuencia nuestra base de datos puede ser mas robusta con menos peso computacional.

La transformada de Hadamard esta íntimamente relacionada con la transformada de Walsh, y en muchas ocasiones se logra encontrar un único tratamiento para ambas, se hace referencia a esta como transformada de Walsh-Hadamard (WHT). A diferencia de la transformada de Fourier que se basa en términos trigonométricos, esta consiste en series de funciones básicas cuyos valores son de 1 y -1.
Las propiedades más importantes de esta transformada se verán enumeradas a continuación:

1.- La transformada de Hadamard [H] es real, simétrica y ortogonal:

$$[H] = [H]^* = [H]^T = [H]^{-1}$$

2.- HDT (Hadamard Discrete Transform) es un transformada rápida, ya que esta solo contiene valores de +/- 1, por lo que no se requieren multiplicaciones para su calculo. Así es que el numero de sumas o restas puede reducirse desde N^2 hasta $N\ log_2 N$.

3.- HDT puede compactar una buena cantidad de energía para imágenes altamente correlacionadas o para nuestro caso secuencias.

Para el caso de este trabajo de tesis, solo se analizara el caso unidimensional de esta transformada. Como nuestras señales de pitch solo son de una dimensión y serán compactadas para un mejoramiento en tiempo de cálculo.

3.6.1 HDT Unidimensional.

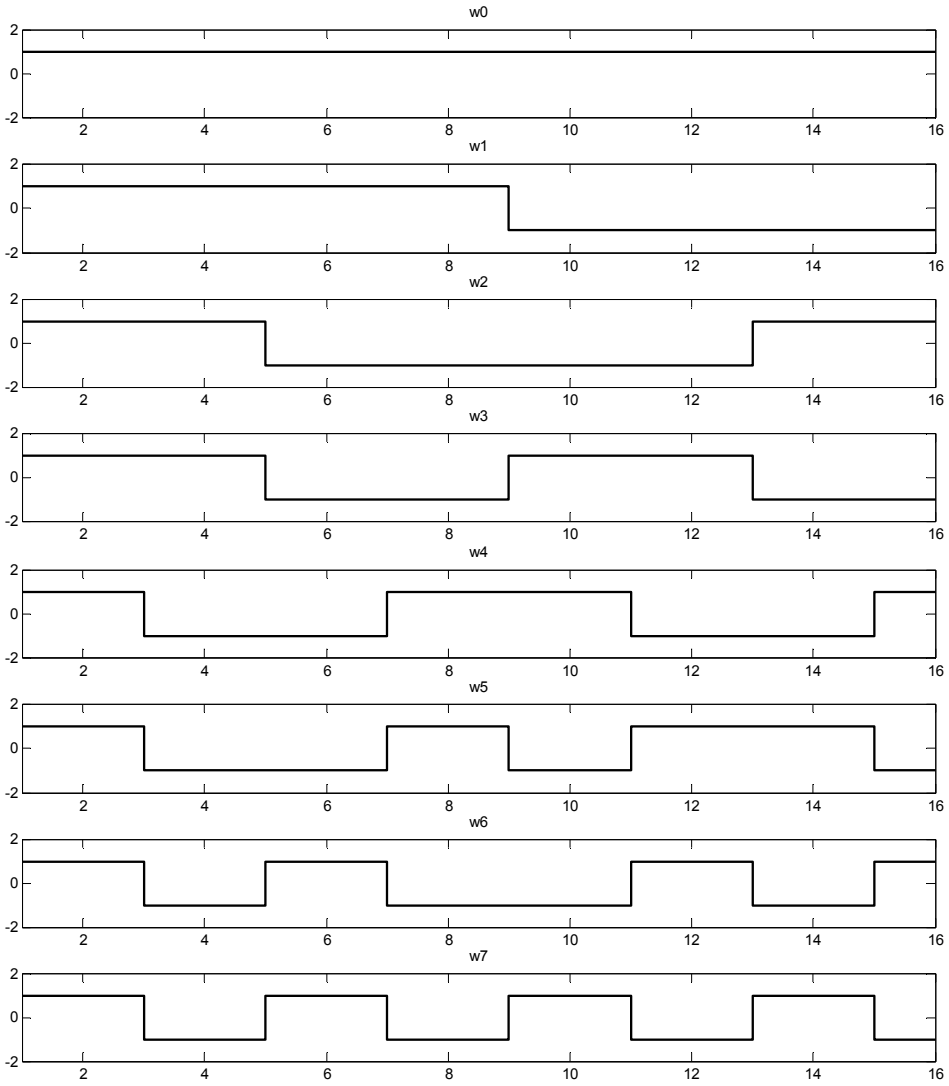
La formulación para función de la transformada de Hadamard [30] esta determinada por la siguiente relación.

$$g(x,u) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

La suma del exponente se realiza en modulo 2, siendo $b_k(z)$ el bit k-ésimo de la representación binaria de *z* y *N* el numero de muestras.
A partir de la función base, la transformada da Hadamard se expresa como

$$H(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Donde $N = 2^n$, y *u* toma valores en el intervalo 0, 1,2…, $N-1$



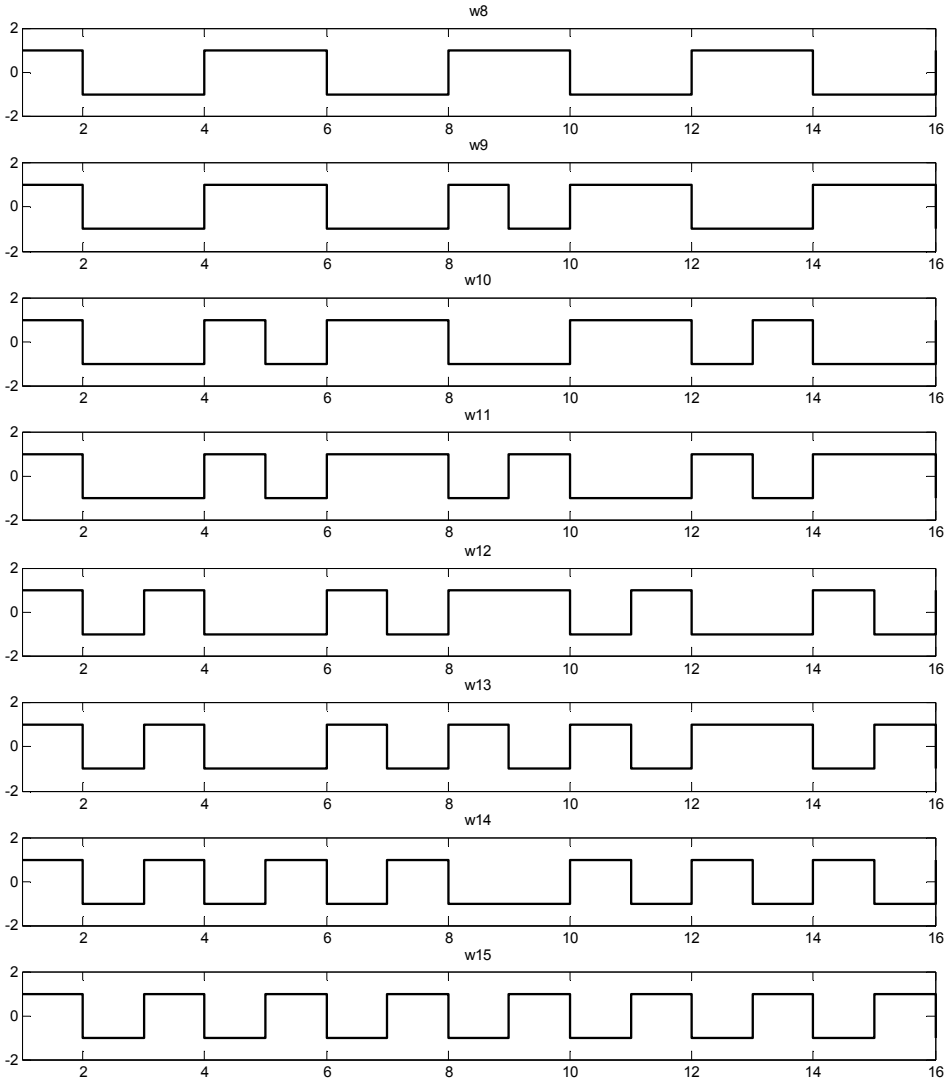


Figura 3.31. Funciones base de la transformada de Hadamard unidimensional (N=16)

Como en el caso de la transformada de Walsh, la función base de Hadamard forma una matriz cuyas filas y columnas son ortogonales. Esta condición nos lleva a una función base inversa que salvo el factor $1/N$, es igual a la función base Hadamard directa; es decir,

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Por lo tanto, la transformada inversa de Hadamard unidimensional se expresa como,

$$f(x) = \sum_{u=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Para $x = 0, 1, 2, \ldots, N - 1$

Para nuestro caso el número de muestras promedio de nuestras canciones pueden ser desplazadas a 512 muestras figura 3.32. Con lo que calculando la transformada de Hadamard a cada una de nuestras canciones y reconstruir la señal de entrada con los primero 50 coeficientes de esta se determinó que era suficiente para ser reconstruida.

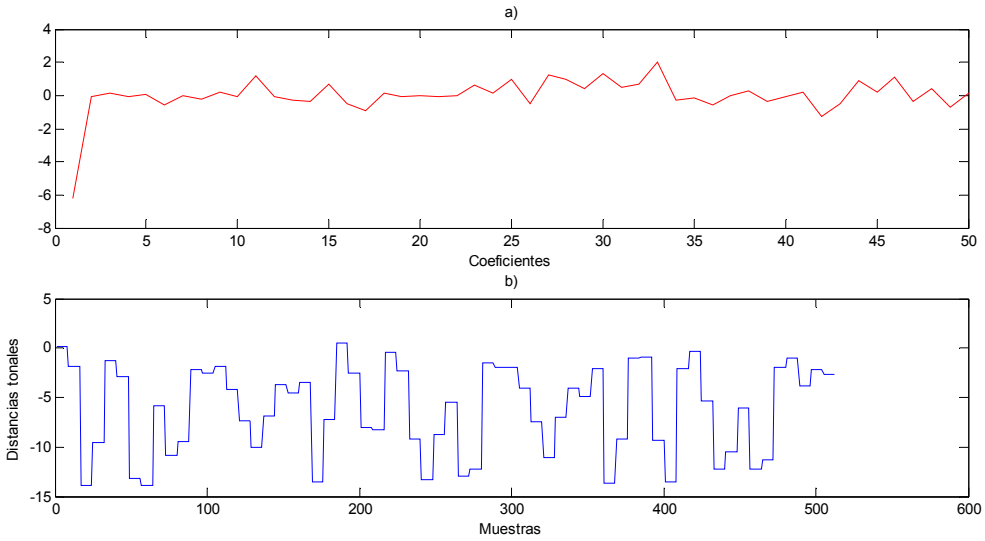


Figura 3.32. a) Transformada de Hadamard (512) primeros 50 coeficientes. b) Reconstrucción de la figura 3.30.

Al tener esta reconstrucción, se elaboro una evaluación del sistema con estos 50 coeficientes dando similares resultados de éxito, que ocupando cada uno de las muestras del pitch. Cabe señalar que para esta evaluación solo se ha tomado en cuenta el cálculo de distancias directas; ya que el método DTW es demasiado sensible a las variaciones teniendo un desempeño en la evaluación mucho menor.

Puede observarse que los resultados pueden ser son positivos para nuestros propósitos pero de igual manera, se aprecia que nuestra señal de pitch es relativamente grande y sus variaciones pueden ser decimadas a un 50% de su longitud figura 3.33.

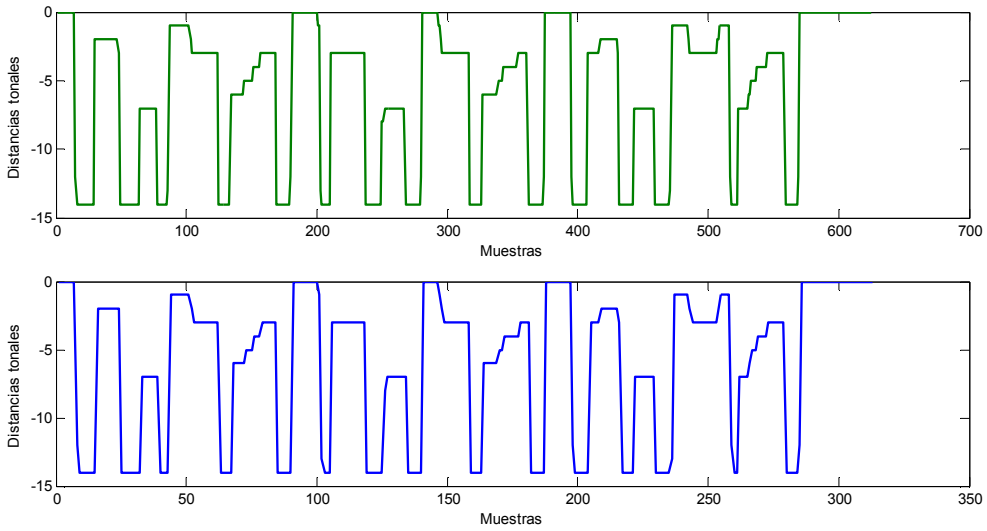


Figura 3.33 Decimación de las tonalidades de una canción

Así que se opto por realizar la decimación de la señal para reducir su tamaño para realizar su transformada. Pero esta vez con solo 256 muestras y tomando para su reconstrucción solo 25 muestras figura 3.34.

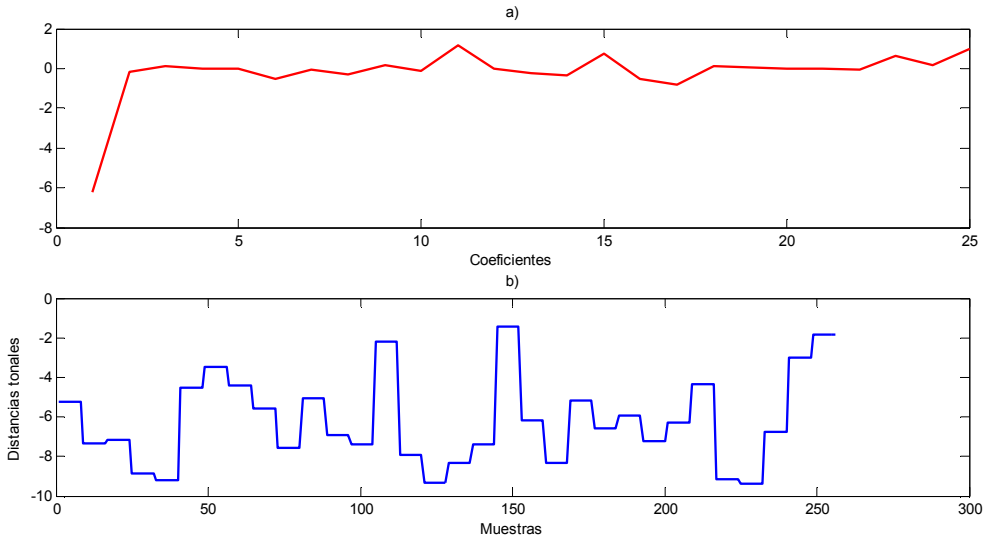


Figura 3.34. a) Coeficientes de la transformada de Hadamard (256) primero 25 coeficientes. b) Reconstrucción de la decimación de la figura 3.33.

Al compararse la señal reconstruida y la señal original, existen variaciones que se observan a simple vista aunque en su forma son muy similares. Con lo que se pudo realizar la misma evaluación y determinarse que aun así tiene un desempeño similar que al de trabajar con las secuencias de pitch originales. Las evaluaciones realizadas con secuencias originales y las realizadas con la transformada de Hadamard tomando 25 coeficientes serán expuestas en el siguiente capitulo.

4.7 Conclusiones.

La representación adecuada del pitch nos ha proporcionado una ventaja para ambos sistemas. Para el caso del sistema monofónico, nos da la posibilidad de realizar una compresión lo cual es de suma importancia ya que al disminuir los datos involucrados las operaciones se reducen bastante. Haciendo que el sistema, sea capaz de manejar un mayor numero de canciones con menor información. El sistema polifónico cuenta con una huella digital, la cual puede ser analizada para obtener diversos descriptores a partir de ella. Haciéndola una herramienta útil no solo para este sistema sino para encontrar alguna otra característica de la canción.

Capitulo 4.

Evaluación de los sistemas.

4.1 Evaluación del sistema monofónico (Voz Vs Voz)

La evaluación de nuestro sistema, consistirá en el conteo de éxitos de cada uno de las canciones existentes en nuestra base de datos que consta de 50 canciones diferentes. Principalmente música popular occidental; algunas de estas canciones son mas complicadas de tararear con la finalidad de tener diversidad en las melodías, y que esta evaluación sea objetiva. Tomando en cuenta, que cada una de las canciones esta fragmentada en sus partes más representativas o características como son: verso, el coro y en ocasiones la introducción o el puente según sea el caso, nuestra base de datos se forma de un total 345 fragmentos. Por lo que para una identificación se comparara contra todos los archivos involucrados.

Los archivos capturados para nuestra base de datos, fueron grabados en el formato requerido por el sistema; es decir, formato WAV a 8KHz de tasa de muestreo. No obstante el usuario puede haber hecho su grabación en otro formato como mp3 o inclusive AAC (Advanced Audio Coding). En cuyo caso, para ser compatible se procede a transformar el archivo al formato compatible con el sistema. Así dando la posibilidad de que mas usuarios puedan utilizar la aplicación.

Para esta evaluación se requirieron de diversos usuarios entre hombres y mujeres con tonos de voces distintas. Basándose en las tonalidades de voces expuestas en el capitulo 2 tema 7; algunos de estos usuarios tienen conocimientos en música y por otra parte, la mayoría no tienen ningún conocimiento en música. Siendo posible una aproximación de la efectividad del sistema.

Para hacer una descripción, de como ha de funcionar el sistema en forma sintetizada se exponen los procesos a continuación:

1. Grabación del tarareo por parte del usuario.
2. Envío de la grabación al servidor de la aplicación.
3. Revisión de compatibilidad del archivo de audio recibido en el servidor.
4. En caso de estar en un formato incompatible se transforma al formato deseado, en caso contrario se procede al paso 5 directamente.
5. Extracción del pitch.
6. Transformación de pitch a distancias tonales.
7. Comparación con la base de datos.
8. Identificación de los candidatos.
9. Envío de datos al usuario de la aplicación.

Hay diversos factores que se han tomado en cuenta, como el número de intentos en obtener una respuesta acertada. Así como la posición de la sugerencia que proporciona el sistema. Nuestro sistema proporciona 3 candidatos figura 4.1 a la canción, tomando en cuenta que la menor distancia proporcionada por el sistema será la mas parecida para este. Considerando que si el sistema despliega el nombre de la canción correcta dentro de estas 3 sugerencias, será tomada en cuenta como búsqueda exitosa. En caso contrario si se requirieron de más intentos, no serán tomados en cuenta en la evaluación; esto con la finalidad de tener datos lo mas apegados a la realidad del uso del sistema y tener un porcentaje de eficiencia mas preciso.

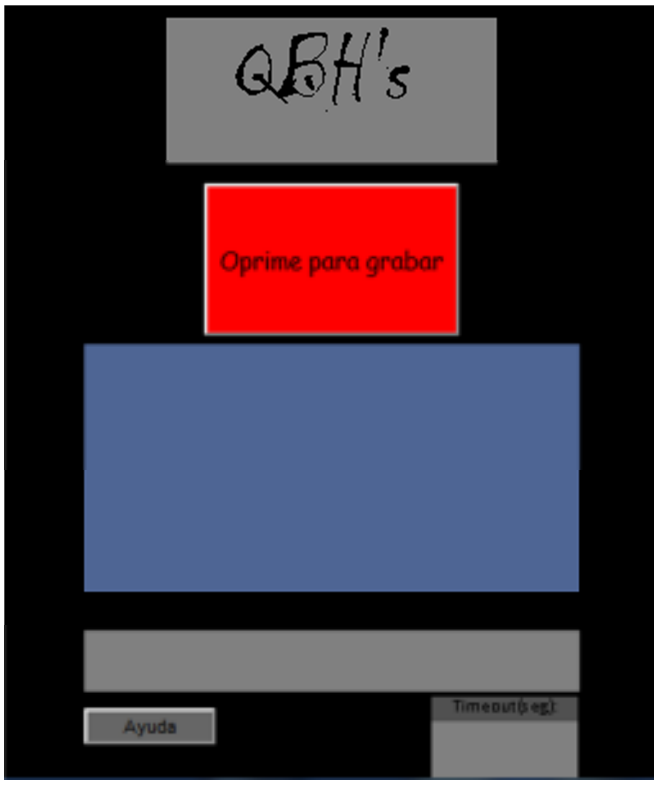


Figura 4.1 Muestra de la aplicación del sistema

Las instrucciones que se le dan al usuario son desplegados al oprimir el botón de ayuda. Este proporciona las instrucciones básicas para usar la aplicación, como se muestra en la figura 4.2.

Figura 4.2 Instrucciones de uso de la aplicación

Las instrucciones recomiendan tararear lo mejor posible la canción, esto para dar una mejor probabilidad de éxito. Evitar el ruido mientras se graba es una medida importante, ya que esta puede alterar los resultados significativamente.

Dando la aplicación y las instrucciones a los usuarios se prosiguió a la evaluación del primer sistema. Guardando los resultados en una serie de tablas que serán añadidas al anexo, fueron tomadas de entre muestras de hombres y mujeres. Registrando cada uno de los resultados, para cada una de las canciones almacenadas en nuestra base de datos. A continuación se muestran las tablas de resultados de algunos ejemplos de manera sintetizada para una mejor demostración de los datos obtenidos.

Canción	Distancias directas
Smoke On The Water	2.700
Smoke On The Water	3.797
Imagine	3.805

a)

Canción	Distancias DTW
Smoke On The Water	833
Song 2	1873
Obladi Oblada	2139

b)

Tabla 4.1 Resultados de hombre en primera prueba. a) Tabla distancias directas, b)Tabla DTW.

Esta primera tabla se refiere a una prueba del sistema con un hombre tarareando cada una de las canciones de la base de datos. La tabla muestra las distancias que fueron registradas por los métodos de validación, para la primera distancia se toman en cuenta las distancias del primer método visto en esta tesis por distancias directas, el segundo método muestra las distancias tomadas con el método DTW.

En la tabla 4.1 se tiene una muestra de evaluación al tararear la canción "Smoke on the water" la cual fue exitosa con ambos métodos. Como se aprecia en ambos resultados la distancia mas corta corresponde a la canción tarareada. Si examinamos los resultados, para el primer método la distancia de entre el primero y el segundo es de 1, lo que significa que son cercanas y los resultados verifican esto, el motivo por el cual sale 2 veces la misma canción es por que a cada fragmento se le asocio un nombre. Si ahora nos centramos en el método 2 podemos ver mas claramente en las distancias arrojadas que la canción numero 1 tiene una distancia considerablemente mas corta lo que se refleja en un buen desempeño en el reconocimiento.

Cada una de las canciones fue reconocida aunque en ocasiones se requirió de repetir la prueba por problemas de ruido o grandes variaciones tonales, con lo que en segundos o terceros intentos se reconoció la canción exitosamente.

Para el caso de la prueba de la tabla 4.1 el usuario consiguió un porcentaje de éxito o eficacia de **80%** para el método de distancias directas y de **78%** para DTW lo cual refleja un buen desempeño y con similares resultados para ambos métodos. Estos resultados están sujetos a la condición de la tabla 4.3.

A continuación se muestran más tablas de resultados para aproximar una eficiencia del sistema.

Canción	Distancias directas
Ruby	1.834
Happy Birthday	1.970
Another Brick In The Wall	2.341

a)

Canción	Distancias DTW
Imagine	988

Ruby	1068
The Dark Of The Matinee	1134

b)

Tabla 4.2 Ejemplo de resultado segunda opción.

En el ejemplo mostrado en la tabla 4.2, se muestra para el caso del método uno se a tenido éxito en el reconocimiento de la canción "Happy Birthday. Pero se observa que la sugerencia correcta esta en la segunda posición. Este es un resultado muy común, dado que muchas veces el usuario tararea mal, o simplemente ha encontrado más coincidencias en esta. Sin embargo es valido ya que para el uso de una aplicación de este tipo se requiere de un cierto rango de tolerancia hacia los resultados, en nuestro caso ha sido 3. Aunque en aplicaciones comerciales pueden existir alrededor de 6 sugerencias.

Ahora si tomamos asunto en el método 2 ha fallado el reconocimiento, asegurando que se trata de la canción "Imagine" pero en realidad no lo es. Este ejemplo expone un caso fallido utilizando DTW el cual es un método que requiere más tiempo y aun así ha fallado. Esta es una observación que puede llegar a repetirse a menudo por lo que se puede dar mas grado de confianza al método de distancias directas.

Solo se mostraran algunas de las tablas de evaluación más significativas en los anexos de esta tesis para una consulta más a fondo de estos.

Cabe notar que los resultados tienen similitudes tanto en el desempeño del método 1 de identificación y el método 2 con una baja diferencia en el método 2 (DTW), aunque en ocasiones este método ha mostrado un valor mas acertado este requiere de muchísimo más tiempo en la comparación y obtención de los resultados.

El tiempo promedio de resolución con el método 1 esta apenas 1.5 segundos, cuando con el método de DTW el desempeño puede variar hasta alrededor de 30 segundos lo cual es un aumento considerable en el tiempo de respuesta. Tomando en cuenta que tienen desempeños similares cada uno de los métodos, pero marcadas diferencias en el tiempo de procesamiento. Se delibero que en primera instancia se utilizara el método 1 para brindar una respuesta más rápida a los usuarios y solo usar el segundo método para dar una segunda opinión en la respuesta al usuario para brindar una mejor calidad en el uso de la aplicación.

Una de las justificaciones que se usaron para considerar que el método 1 es mas apropiado además del tiempo de procesamiento, es que un usuario que hace su petición de búsqueda en la mayoría de los casos tiene poco tiempo de haber oído la melodía, con lo que el tiempo y la tonada serán lo más cercano a la canción original. Por lo que la búsqueda será mas acertada si la melodía que tararea el usuario es lo mas cercano a la canción original, con lo que podemos decir que entre menor sea el tiempo en que el usuario haya oído la canción

son mas grades probabilidades de que su búsqueda sea exitosa, así sacando ventaja de que esta es una aplicación sea para un dispositivo móvil el usuario puede hacer su petición inmediatamente.

Si tomamos en cuenta las referencias de éxito, respecto a si la canción tarareada fue la primera opción. El promedio baja como se muestra en la tabla 4.3.

Candidatos	Porcentaje de éxito.
Tomando primera posición	63%
Tomando hasta segunda posición	75%
Tomando hasta tercera posición	80%

Tabla 4.3. Desempeño respecto a número de candidatos

En la tabla 4.3 se reflejan los resultados globales en base al número de candidatos que se tienen en la aplicación. Si ésta cuenta con más candidatos los resultados mejoran, de tal manera que nuestras pruebas se realizaron con tres candidatos que es la que ofrece un mejor desempeño para nuestra aplicación sin poner demasiados candidatos.

Si tomáramos en cuenta que la canción pudo ser identificada en un segundo o tercer intento este porcentaje subiría considerablemente alrededor del **90%**, que dado el elevado porcentaje es ideal.

4.2 Evaluación del sistema monofónico con compresión con Transformada de Hadamard.

A diferencia del primer caso la finalidad de este es hacer al sistema más rápido y que pueda almacenar un número mayor de canciones con un menor número de datos.

El único cambio que existe entre este sistema y el expuesto en el punto anterior es que este cuenta con paso un extra, el cual involucra la transformada de Hadamard, con la finalidad de reducir el número de operaciones involucradas en el cálculo de identificación.

Como se explico en el capitulo 3 tema 6 la representación de distancias tonales puede ser comprimida reduciendo los datos de la canción a tan solo 25 coeficientes de Hadamard, esta es una aportación que se obtuvo tras diversas pruebas y que ha tenido un éxito muy favorable para el desempeño de nuestro sistema.

La observación mas relevante de esta aportación es que anteriormente para realizar el calculo de identificación se hacían en promedio 512 operaciones por fragmento de canción por lo que el numero de operaciones para nuestra base se extendía a 176640 operaciones. Sin embargo al realizar esta compresión el número de operaciones se reduce drásticamente

ya que este solo hace 25 operaciones por fragmento de canción dándonos tan solo 8625 operaciones. Haciendo notar que la compresión no es un proceso muy elaborado ya que este solo se realiza con sumas y restas, haciéndolo ideal para nuestro objetivo.

A continuación se mostraran algunas de las tablas de evaluación del sistema monofónico pero esta vez utilizando para la identificación solo los primeros 25 coeficientes de la transformada de Hadamard en las canciones, esto con la finalidad de hacer un comparativo del desempeño entre usar las secuencias del pitch y los coeficientes en el dominio de la transformada.

La evaluación consistirá en las mismas condiciones en la que se realizo la anterior omitiendo el método DTW, que no ha tenido un desempeño muy favorable para esta prueba. Se mostraran en tablas similares a las anteriores sin tomar en cuenta las observaciones que son similares en la mayoría de los casos.

Solamente se mostraran aquellos casos mas significativos de la evaluación, mostrando así el resultado de éxito mas alto, la media y el mínimo, con la finalidad de tener un panorama de los resultados de evaluación obtenidos para este sistema.

Canción	Distancias directas
Inhale	0.400
Happy Birthday	0.445
Come As You Are	0.449

Tabla 4.4 Primera tabla de evaluación utilizando coeficientes de Hadamard

En la tabla 4.4 se tienen los resultados de un ejemplo de la evaluación. En este caso se ha tarareado la canción “Inhale” la cual ha sido identificada correctamente utilizando los coeficientes de Hadamard. Aquí las distancias de los candidatos ya son tan marcadas como en los casos anteriores, o por lo menos en apariencia dado que los números obtenidos de los coeficientes de Hadamard son pequeños, y he de aquí que se observen estos resultados.

Canción	Distancias directas
Sevennation Army	0.453
Yellow Submarine	0.654
Afuera	0.660

Tabla 4.5 Segunda tabla de evaluación utilizando coeficientes de Hadamard

Para la tabla 4.5 de igual manera que para el sistema sin compresión, la canción que se a tarareado aparece en la segunda posición ("Yellow Submarine"). Vemos que la distancia que aparece en la primera posición es más pequeña. Problemas de este tipo aparecen cuando empleamos la trasformada de Hadamard pero al final el resultado es bueno así que no representa un problema grave para nuestra aplicación.

Las tablas que se encuentran en el anexo de evaluación reflejan un desempeño similar al primer sistema utilizando las secuencias originales de las canciones, esto se deriva a que nuestras canciones pueden ser representadas con menores elementos como son estos coeficientes. Una observación que vale la pena notar es que, para nuestra pequeña base de datos fue suficiente con 25 coeficientes, pero al aumentarse la base datos nuestros coeficientes deberán de aumentar en proporción al número de canciones almacenadas en nuestra base de datos, tomando en cuenta que aun así nuestras canciones podrán ser representadas con muy pocos datos, el *tiempo de respuesta se reduce de 1.5 a 2 segundos hasta 0.5 a 0.9 segundos*.

Haciendo un comparativo entre el primer método monofónico y con compresión de Hadamard en la tabla 4.6. Se pueden observar que en promedio tiene desempeños similares, lo cual refleja una ventaja al utilizar nuestra compresión dado que no existe una disminución en nuestros porcentajes.

Candidatos	Porcentaje de éxito
Tomando primera posición	62%
Tomando hasta segunda posición	76%
Tomando hasta tercera posición	81%

Tabla 4.6 Desempeño respecto a numero de candidatos utilizando Hadamard

4.3 Evaluación del sistema polifónico

Para el sistema polifónico la finalidad de la evaluación es saber si es lo suficientemente efectivo para hacer un reconocimiento exitoso de las canciones almacenadas con la base de datos ya que este método de obtención de este descriptor es uno entre tantos otros que se proponen en la actualidad para este sistema en específico. Es de importancia destacar que los resultados obtenidos en esta evaluación fueron revisados por Matthias Robine y Pierre Hanna miembros de MIREX (Music Information Retrieval Evaluation eXchange) [26]

expertos en la materia y supervisores de este trabajo, actualmente investigando sobre la obtención de descriptores para QBH polifónico.

En las siguientes tablas solo se mostraran algunos de los casos de evaluación con las mismas 50 canciones que se pusieron a prueba en el sistema anterior. Como en el caso anterior las tablas cuentan con las distancias adquiridas por los dos métodos de identificación (Método de distancias directas y DTW).

Este sistema de identificación de canciones aun se encuentra en etapa temprana por lo que no cuenta con una aplicación para un dispositivo móvil, así que esta evaluación se ha realizado con muestras tomadas de usuarios. Este método de obtención de este descriptor es nuestra propuesta y aporte a la investigación del reconocimiento de canciones por tarareo para un sistema polifónico es decir Voz vs Música.

Canción	Distancia 1	Distancia 2
Afuera	3.304532578	6334
Another Brick In The Wall	2.964774951	990
Another Brick In The Wall	5.25	7680
Another Brick In The Wall	2.746361746	2671
Black Night	3.836611195	2551
Black Night	5.773071104	20646
Blue Monday	2.542302358	3654
Breaking the law	2.750378215	3024
Break up song	3.797004992	4144
Break up song	5.472324723	10879
Break up song	7.435153584	28699
Can't take my eyes off you	6.959778086	25534
Can't take my eyes off you	2.667487685	2079
Can't take my eyes off you	4.212058212	5911
Come as you are	5.498812352	12175
La cucaracha	2.228441755	2784
La cucaracha	5.857894737	11119
Dark of the matinee	5.224770642	13504
Dreaming of you	2.677419355	3079
El baile	4.673130194	9559
Evil	1.073891626	481
Fat lip	9.539923954	41434

Happybirthday	6.642965204	29146
Happybirthday	8.524271845	42631
Happybirthday	7.69889065	29479
Highway to Hell	6.611650485	24609
Highway to Hell	3.633841886	5725
I Feel good	3.048543689	2419
I Feel good	2.622781065	2829
Imagine	5.378186969	23659
Inagadda da Vida	6.22327791	20800
Inhale	7	20374
In my place	2.599150142	3535
Ironman	7.582840237	33241
Jeremy	5.905686546	18454
Jingle	6.403326403	10330
Last kiss	7.979195562	35029
Last Nite	4.692094313	18799
Love me two times	8.090425532	26334
Munich	2	630
No me destruyas	4.77244582	13759
No me destruyas	4.842105263	17224
Obladi Oblada	5.328125	12136
Obladi Oblada	3.332871012	4344
Otherside	5.355062413	12561
Paint it black	4.128987517	4849
Paint it black	5.082191781	10369
Reptilia	3.004643963	4011
Ruby	9.458174905	37915
Ruby	6.385321101	17661
Run to the hills	3.524271845	3621
Sevenation army	2.314840499	3166
Sing	5.851595007	8821
Smells like teen spirit	5.47733711	17109
Smoke	4.085991678	4116
Song 2	4.337438424	6306
Starlight	6.821081831	34504
Sunshine	2.33148405	2776
Take me Out	7.081081081	13741
The man who sold the world	3.871559633	5644

Ven aquí	4.626907074	8766
We found love	4.839009288	12831
We found love	6.920943135	27535
Yellow submarine	2.607489598	4120
Yellow submarine	5.360610264	19270
You really got me	4.870090634	8584
Distancia mas corta	Evil	Evil
¿Éxito?	si	si
Canción tarareada	Evil	

Tabla 4.6 Primera tabla de evaluación de sistema polifónico.

Como se muestra en la tabla 4.6 una canción se puede mostrar mas de una vez, esto es debido a que se añadieron los diversos fragmentos compuestos por la canción, y el sistema debe ser capaz de discernir entre todos y cada uno de los fragmentos que llamaremos ruido en la búsqueda. La finalidad es ver si se es capaz de reconocer la canción entre todo el ruido existente en nuestra base de datos.

Un aspecto que se puede observar en nuestra tabla es que las distancias en el método de DTW se han hecho muy grandes, y esto muestra la sensibilidad del método por lo que para este tipo de sistema no resulte muy conveniente ya que hay demasiadas variaciones y como hemos visto este método compara la similitud entre las secuencias y calcula las distancias entre ellas. Para este caso ambos métodos ha sido exitosos mostrando una distancia marcadamente más corta con respecto al ruido.

Mas tablas de resultados serán puestas en la sección anexo para una mejor referencia debido a la extensión de las pruebas, se añadieron algunos de los casos más comunes en aparecer en la evaluación.

En esta evaluación se ha determinado que este funciona **en *62%*** (o 32 canciones identificadas) lo cual refleja un desempeño favorable respecto a otros métodos [25] aun sin tomar en cuenta las segundas o terceras opciones en las cuales este porcentaje sube alrededor de **70%** que es comparable con los resultados obtenidos en el sistema monofónico.

Una de las observaciones que se pueden hacer es que entre mas ruido o instrumentos tenga una canción es mas complicado obtener un identificador, por otra parte entre mas simple sea una canción es mas sencillo de tararear la melodía y por ende la canción será reconocida, y basándose en que las canciones occidentales de la cultura popular tienen en su mayoría un estructura musical simple es muy probable que el sistema pueda llevarse a una escala comercial.

Este descriptor propuesto no solo puede ser útil para sistemas QBH sino para otros sistemas de clasificación ya que con ayuda de este es posible reconocer ciertos instrumentos musicales además de ser una herramienta que puede ser de utilidad para identificación de semejanza en melodías para protección de derechos de autor.

Aunque no se ha llegado a un descriptor perfecto se han logrado obtener buenos resultados los cuales pueden ser de aporte a este complicado tema de investigación en el que actualmente se trabaja. Se a tenido contacto con los expertos en la materia en LaBRI en Bordeaux Francia como parte de la estancia de investigación que se ha realizado en dicho lugar así teniendo la certeza de tener resultados favorables para este trabajo de tesis.

4.4 Conclusiones.

La evaluación para el caso monofónico refleja un desempeño favorable, cabe destacar que al utilizar la transformada de Hadamard, esta nos presenta una mejoría al disminuir drásticamente el número de operaciones a realizarse, esta es una aportación que nos ayuda dado que el sistema mejora su rendimiento, dado que aun con esta compresión el resultado de éxito es similar al anterior.

El sistema polifónico demostró tener la capacidad de reconocer ciertas canciones, dándonos como resultado la identificación de un 62% de las canciones que resulta ser un buen resultado en comparación con trabajos del mismo tipo, cabe señalar que aunque se tienen dichos resultados estos podrían ser mejorados al encontrar un mejor método para analizarla huella digital de la canción, para así tener un desempeño similar a su contraparte monofónica.

Capitulo 5.

Funcionalidad en dispositivos móviles.

Los dispositivos móviles inteligentes en la actualidad se basan en sistemas operativos, los cuales administran todas y cada una de las aplicaciones, y en la mayoría de los casos estos cuentan con un lenguaje de programación para realizar sus aplicaciones. Los más populares sistemas operativos para la gran variedad de aparatos que existen son Android y el iOS de Mac, es por eso que nos basaremos principalmente en estos 2 sistemas operativos para la explicación del funcionamiento de la aplicación para estas plataformas.

En ambos lenguajes de programación pueden determinarse las características o tareas que debe realizar nuestra aplicación, una de estas características es vital importancia para nuestra aplicación y se trata de la captura de audio.

El esquema de nuestra aplicación es principalmente hacer una captura de audio, en nuestro caso un pequeño fragmento de una canción interpretado por el usuario con su voz. Esto con la finalidad de que nuestra aplicación sea compatible con la mayoría de los dispositivos existentes. El calculo del pitch, el filtrado y el calculo de la transformada de Hadamard, a pesar de ser operaciones relativamente sencillas pueden causar problemas para un dispositivo que no cuente con gran capacidad de cómputo, haciendo que su cálculo sea lento o simplemente no funcione, por ende la aplicación solo se dedicara a grabar la voz del usuario, y desplegar información en la pantalla haciendo de esta manera una aplicación muy sencilla y al alcance de cualquier dispositivo figura 5.1.

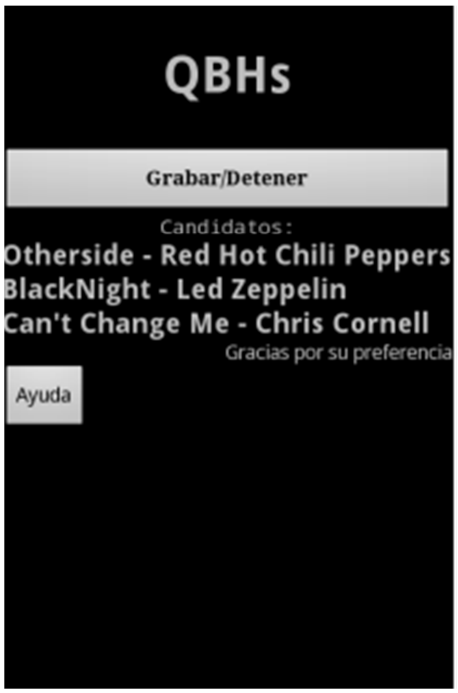


Figura 5.1 Esquema de aplicación en sistema Android 2.3.

5.1 Descripción de captura de audio para Android e iOS.

La ejecución de nuestro sistema requiere de archivos de audio originales en otras palabras el archivo .WAV que es generado comúnmente por cualquier computadora o algunos dispositivos. Sin embargo muchos de los teléfonos no tienen la opción de generar este tipo de archivos de audio, en cambio son codificados en otros tipos como son el conocido mp3 y últimamente más utilizado AAC (Advanced Audio Coding) que son archivos de audio comprimidos y por lo tanto no tienen las características requeridas por el sistema. No obstante eso no es un problema grave ya que el sistema es capaz de recibir el archivo en este formato y transformarlo al formato valido para nuestro sistema.

En la figura 5.2 apreciamos un fragmento de código que permite realizar la captura de audio para Android proporcionando la información de codificación para el archivo.

```
private void startRecording() {
    mRecorder = new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mRecorder.setOutputFile(mFileName);
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);

    try {
        mRecorder.prepare();
    } catch (IOException e) {
        Log.e(LOG_TAG, "prepare() failed");
    }

    mRecorder.start();
}
```

Figura 5.2 Código en Android para captura de audio

En el código se especifica o se crea un objeto llamado mRecorder este objeto es la grabadora que va a llevar a cabo la grabación y es este donde se vaciaran los datos sobre la grabación. Primera mente se accede al micrófono del dispositivo, posteriormente los formatos de salida y un codificador de audio.

El ejemplo el formato especificado para la aplicación es 3gpp que es un formato muy común para la compresión de voz en teléfonos celulares no obstante el sistema cuenta como antes mencionado diversos formatos que pueden ser utilizados para la codificación del audio en el caso de nuestra aplicación AAC.

int	AAC_ADTS	AAC ADTS file format
int	AMR_NB	AMR NB file format
int	AMR_WB	AMR WB file format
int	DEFAULT	
int	MPEG_4	MPEG4 media file format
int	RAW_AMR	*This constant is deprecated. Deprecated in favor of MediaRecorder.OutputFormat.AMR_NB*
int	THREE_GPP	3GPP media file format

Figura 5.3 Formatos de codificación de audio soportados por Android

En la figura 5.3 tenemos una lista de formatos de salida para codificación de audio utilizada para el sistema operativo Android, de igual manera para el sistema es manejado de manera similar para obtener el formato de compresión deseado.

Los sistemas operativos Android e iOS ambos ocupan la codificación AAC aunque pueden ser configurados en ambos para obtener los parámetros deseados para su compresión. Estos parámetros pueden ser consultados en [31] y [32]. Aunque en otros dispositivos se pueden usar capturas en formato mp3 o inclusive WAV son validos ya que siempre se adecuaran a las características requeridas para su análisis.

5.2 Envió de Datos de grabación y de respuesta.

Él envió de los datos capturados por el usuario y los datos de respuesta están pensados para ser enviados vía internet o en su defecto un mensaje multimedia siendo esta una opción para aquellos que no cuenten con acceso a la red. Al ser comprimidos en formato AAC los archivos son bastante pequeños de 100KB a 200KB por lo que él envió de estos datos atreves de internet no requieren de un gran ancho de banda aunque si influirá en la velocidad de respuesta de la aplicación.

Una de las características que debemos tomar en cuenta es que la aplicación debe tener el permiso para tener la capacidad de guardar el archivo de audio para posteriormente ser enviadas, claro esta que también debe ser especificado el puerto por el cual el archivo se va enviar y a recibir.

Los datos serán primeramente recibidos por un servidor el cual procesara inmediatamente la canción para su comparación en nuestra base de datos, una vez transformados si es necesario y analizados los archivos son comparados en la base de datos, para encontrar los mejores candidatos, una vez obtenidos los candidatos, se accede a la información asociada a ellos para ser enviada al usuario, en el caso de nuestra aplicación primordialmente el nombre de la canción y el artista que la interpreta.

5.3 Conclusiones.

El desarrollo de una aplicación para plataformas como Android o iOS es parte de este sistema, aunque dichas aplicaciones no estén disponibles en este momento para su comercialización o difusión, pueden ser desarrolladas para ser funcionales he ir siendo expandidas poco a poco, aunque la finalidad de este trabajo es proponer solamente los métodos para el desarrollo de un sistema de música por tarareo para dispositivos móviles y no el lanzamiento de la aplicación formalmente al mercado.

Capitulo 6.

Conclusiones.

Se propusieron dos métodos para la resolución del sistema de identificación de canciones. El primero basado en un método el cual consiste en la comparación de melodías tarareadas, para su posterior comparación denominado en este trabajo como sistema monofónico. El segundo sistema, o método de obtención de descriptor esta basado en el actual estudio del reconocimiento de canciones desde el audio original de la canción. Es decir voz contra música descrito aquí como sistema polifónico.

En lo que respecta al primer sistema, las pruebas de evaluación son satisfactorias. Ya que se pudieron reconocer todas las canciones y con una probabilidad de éxito de 80% aproximadamente, en la búsqueda si se usa apropiadamente la aplicación. En términos de tiempo de respuesta se muestran buenos resultados con el método de distancias directas. Debido a que es reducido el número de operaciones a realizar, permitiendo hacer una búsqueda de entre millones de canciones. Tomando en cuenta, que se podrían separar las búsquedas en grandes grupos como en géneros musicales haciendo una búsqueda más rápida. Con el método DTW se pudo cerciorar, que aunque es un buen método de evaluación este requiere de un mayor tiempo de cómputo, casi 30 veces más que el método de distancias directas. En ocasiones funciona con la misma eficacia que el método de distancias directas, por lo que se puede optar por utilizar solo éste como respaldo para las búsquedas hechas con el primero.

La aportación, que se ha realizado al utilizar la transformada de Hadamard para el sistema monofónico. Mejora el desempeño en cuestiones de cálculo de identificación respecto al utilizar toda la secuencia del pitch. Al reducir el número de datos a tan solo 25 coeficientes de Hadamard, hace que el número de operaciones involucradas disminuya en gran medida. Aun así teniendo un desempeño igual de favorable con esta compresión de datos. Cabe señalar que el uso de la transformada, no hace mas pesado el desempeño del sistema. Considerando que este solo se realiza con sumas y restas haciéndolo ideal para nuestro objetivo. Una característica que se resalta de este método, es que posiblemente pueda ser calculado directamente en el dispositivo móvil; para así mandar solamente los coeficientes de Hadamard a la base de datos y ahorrarse el tiempo que requiere el servidor de realizar todas esas operaciones de los usuarios, haciéndose más efectivo.

Una de las desventajas o características del sistema, es que solo esta diseñado para trabajar con las partes más importantes de la canción. Es decir que difícilmente el sistema funcionará adecuadamente si se tararea o se canta alguna parte al azar de la canción. O si simplemente se tararean diversas partes de ésta, pero dado que la mayoría de los usuarios tararea las partes características de la canción; es un problema que no surge a menudo aunque podría ser tomado en consideración para hacer un sistema más robusto.

En el sistema polifónico, se propuso un método de obtención de un descriptor para la identificación de melodías en base a la canción original. Para posteriormente ser comparadas con los tarareos de los usuarios, este método demostró tener un desempeño dentro de las expectativas. Las cuales, nos permitieron reconocer alrededor del 60% de las canciones correctamente. En la actualidad, se siguen diversas investigaciones sobre el tema y gracias a la experiencia de estar en una estancia de investigación en Bordeaux Francia en el laboratorio LaBRI. Se pudo tener contacto y apoyo de dos expertos en el tema de búsquedas de música por descriptores, como es QBH. Gracias a la colaboración se obtuvo un resultado, que puede ser de ayuda para posteriores investigaciones en este tema.

En el desempeño de los métodos de identificación se pudo determinar, que son medianamente satisfactorios para el caso polifónico. Ya que aunque se obtuvo un resultado favorable, existen algunos detalles aun por definir para una identificación mas acertada en el reconocimiento del sistema polifónico. Dado que éste cuenta con una huella digital la cual puede ser examinada en su totalidad. Así dejando la posibilidad de encontrar mejores resultados de búsqueda para hacer más viable este sistema de identificación.

Anexo.

Tablas de evaluación del sistema monofónico sin compresión.

1.-

Canción	Distancia 1	Distancia 2	Exito1	Éxito2	Observación
Afuera	3.65	554	Si	si	1er intento, primera sugerencia
Another Brick In The Wall	3.98	1566	Si	si	1er intento, primera sugerencia
Black Night	2.06	678	Si	si	1er intento, tercera sugerencia
Blue Monday	1.01	1568	Si	si	1er intento, primera sugerencia
Break up song	2.7	342	Si	si	1er intento, primera sugerencia
Breaking the law	1.52	524	Si	si	1er intento, tercera sugerencia
Can't change me	1.69	1331	Si	si	1er intento, primera sugerencia
Can't take my eyes off you	1.38	643	Si	no	1er intento, primera sugerencia
Come as you are	1.77	566	No	no	2do intento, segunda sugerencia
Dreaming of you	3.46	212	Si	si	1er intento, segunda sugerencia
El Baile y el salón	2.54	1423	No	no	2do intento, primera sugerencia
Evil	2.14	231	Si	si	1er intento, primera sugerencia
Fat lip	3.66	1344	Si	si	1er intento, tercera sugerencia
Happybirthday	1.64	1322	No	si	2do intento, segunda sugerencia
Highway to hell	2.74	454	Si	si	1er intento, primera sugerencia
I feel good	2.54	122	Si	si	1er intento, primera sugerencia
Imagine	2.7	2643	No	no	2do intento, segunda sugerencia
In my place	2.37	323	Si	si	1er intento, primera sugerencia
Inagadda da vida	0.81	1442	Si	si	1er intento, primera sugerencia
Inhale	1.39	1234	Si	si	1er intento, primera sugerencia
Ironman	0.87	133	Si	si	1er intento, primera sugerencia
Jeremy	4.44	1353	No	si	2do intento, tercera sugerencia
Jingle bells	3.67	766	No	no	2do intento, segunda sugerencia
La cucaracha	2.32	233	Si	si	1er intento, primera sugerencia
Last kiss	2.77	745	Si	no	1er intento, primera sugerencia
Last nite	4.5	713	Si	si	1er intento, tercera sugerencia
Love me two times	3.54	823	Si	si	1er intento, segunda sugerencia
Munich	1.07	311	Si	si	1er intento, primera sugerencia
Ninth Symphony	1.23	233	Si	si	1er intento, primera sugerencia
No me destruyas	4.37	1442	Si	si	1er intento, segunda sugerencia
Obladi Oblada	3.15	1231	Si	no	1er intento, primera sugerencia
Otherside	2.51	543	Si	si	1er intento, primera sugerencia
Paint it black	0.44	122	Si	si	1er intento, primera sugerencia

Reptilia	3.39	423	Si	si	1er intento, primera sugerencia
Ruby	2.13	565	Si	si	1er intento, segunda sugerencia
Run to the hills	2.87	1234	Si	si	1er intento, primera sugerencia
Sevenation army	2.69	343	Si	si	1er intento, primera sugerencia
Sing	4.56	645	No	no	3er intento, tercera sugerencia
Smells like teen spirit	3.42	233	Si	si	1er intento, primera sugerencia
Smoke on the water	2.91	244	Si	si	1er intento, primera sugerencia
Song 2	4.44	734	Si	si	1er intento, segunda sugerencia
Starlight	4.32	1323	No	no	3er intento, segunda sugerencia
Sunshine of your love	3.54	432	No	si	2do intento, segunda sugerencia
Take me out	1.21	138	Si	no	1er intento, primera sugerencia
The dark of the matinee	0.63	1435	Si	si	1er intento, primera sugerencia
The man who sold the world	1.21	241	No	si	2do intento, primera sugerencia
Ven aquí	0.49	91	Si	si	1er intento, primera sugerencia
We found love	2.21	341	Si	si	1er intento, primera sugerencia
Yellow submarine	2.94	1543	Si	no	1er intento, primera sugerencia
You really got me	1.5	237	Si	si	1er intento, primera sugerencia
Porcentaje de éxito			**80%**	**78%**	

2.-

Canción	Distancia 1	Distancia 2	Exito1	Éxito2	Observación
Afuera	3.94	1327	si	si	1er intento, primera sugerencia
Another Brick In The Wall	2.46	326	si	si	1er intento, primera sugerencia
Black Night	2.38	1471	si	si	1er intento, tercera sugerencia
Blue Monday	1.13	N/I	si	no	1er intento, primera sugerencia
Break up song	4.98	1873	si	si	1er intento, segunda sugerencia
Breaking the law	2.41	707	si	si	1er intento, primera sugerencia
Can't change me	2.14	773	si	no	1er intento, segunda sugerencia
Can't take my eyes off you	0.95	773	no	si	2do intento, primera sugerencia
Come as you are	1.58	454	si	no	1er intento, primera sugerencia
Dreaming of you	2.99	523	si	si	1er intento, primera sugerencia
El Baile y el salón	3.24	N/I	si	no	1er intento, segunda sugerencia
Evil	1.86	434	si	si	1er intento, primera sugerencia
Fat lip	3.2	1343	no	no	4to intento, primera sugerencia
Happybirthday	1.89	N/I	si	no	1er intento, segunda sugerencia
Highway to hell	1.44	321	si	si	1er intento, primera sugerencia
I feel good	3.35	1472	si	si	1er intento, tercera sugerencia

Imagine	1.73	968	si	si	1er intento, primera sugerencia
In my place	3.64	1332	si	si	1er intento, primera sugerencia
Inagadda da vida	1.98	429	si	si	1er intento, primera sugerencia
Inhale	0.79	264	si	si	1er intento, primera sugerencia
Ironman	2.32	1232	no	no	2do intento, primera sugerencia
Jeremy	4.24	N/I	no	no	2do intento, primera sugerencia
Jingle bells	4.23	N/I	no	no	4to intento, segunda sugerencia
La cucaracha	2.18	1876	si	si	1er intento, primera sugerencia
Last kiss	1.59	1493	si	si	1er intento, primera sugerencia
Last nite	0.96	1452	si	si	1er intento, primera sugerencia
Love me two times	3.15	1896	si	si	1er intento, primera sugerencia
Munich	0.93	392	si	si	1er intento, primera sugerencia
Ninth Symphony	0.62	219	si	si	1er intento, primera sugerencia
No me destruyas	5.07	1331	no	no	3re intento, segunda sugerencia
Obladi Oblada	3.84	5177	no	si	3er intento, tercer sugerencia
Otherside	1.61	1828	si	si	1er intento, primera sugerencia
Paint it black	0.76	86	si	si	1er intento, primera sugerencia
Reptilia	5.36	341	si	no	1er intento, tercer sugerencia
Ruby	2.87	1278	si	si	1er intento, primer sugerencia
Run to the hills	3.42	332	si	si	1er intento, tercer sugerencia
Sevenation army	1.47	1064	si	si	1er intento, primera sugerencia
Sing	2.34	1348	si	si	1er intento, tercera sugerencia
Smells like teen spirit	2.15	N/I	no	no	3er intento, primera sugerencia
Smoke on the water	3.06	2342	si	si	1er intento, segunda sugerencia
Song 2	5.25	1153	si	si	1er intento, primer sugerencia
Starlight	4.2	2341	no	no	2do intento, segunda sugerencia
Sunshine of your love	3.4	3411	no	no	2do intento, segunda sugerencia
Take me out	1.97	1041	si	si	1er intento, primera sugerencia
The dark of the matinee	0.96	100	si	si	1er intento, primera sugerencia
The man who sold the world	0.84	261	si	si	1er intento, primera sugerencia
Ven aquí	0.66	124	si	si	1er intento, primera sugerencia
We found love	3.2	705	no	si	2do intento, tercer sugerencia
Yellow submarine	2.45	1532	si	si	1er intento, primera sugerencia
You really got me	1.72	744	si	si	1er intento, primera sugerencia
Porcentaje			**78%**	**72%**	

3.-

Canción	Distancia 1	Distancia 2	Exito1	Exito2	Observación
Afuera	4.07	1344	si	si	1er intento, primera sugerencia
Another Brick In The Wall	3.71	1274	si	si	1er intento, primera sugerencia
Black Night	3	678	si	si	3er intento, primeara sugerencia
Blue Monday	1.52	773	si	si	1er intento, primera sugerencia
Break up song	4.96	1345	si	si	1er intento, primera sugerencia
Breaking the law	2.44	902	si	no	1er intento, primera sugerencia
Can't change me	2.88	339	no	no	2do intento, tercer sugerencia
Can't take my eyes off you	4.11	1289	si	si	1er intento, primera sugerencia
Come as you are	N/I	N/I	no	no	sin identificar
Dreaming of you	2.7	394	si	si	1er intento, primera sugerencia
El Baile y el salón	N/I	N/I	no	no	sin identificar
Evil	3.1	455	no	no	2do intento, primera sugerencia
Fat lip	3.43	1293	si	si	1er intento, primera sugerencia
Happybirthday	N/I	N/I	no	no	sin identificar
Highway to hell	3.7	1254	no	no	1er intento, primera sugerencia
I feel good	3.64	730	si	si	1er intento, primer sugerencia
Imagine	4.1	832	si	si	1er intento, tercera sugerencia
In my place	2.14	1534	si	no	1er intento, primera sugerencia
Inagadda da vida	2.56	293	si	si	1er intento, primera sugerencia
Inhale	1.39	192	si	si	1er intento, primera sugerencia
Ironman	0.96	102	si	si	2do intento, primera sugerencia
Jeremy	N/I	N/I	no	no	sin identificar
Jingle bells	2.22	1323	no	no	2do intento, segunda sugerencia
La cucaracha	3.61	967	no	no	2do intento, primera sugerencia
Last kiss	4.37	884	si	si	1er intento, primera sugerencia
Last nite	2.89	325	si	si	1er intento, primera sugerencia
Love me two times	4.53	643	si	si	1er intento, primera sugerencia
Munich	1.01	112	si	si	1er intento, primera sugerencia
Ninth Symphony	1.24	106	si	si	1er intento, primera sugerencia
No me destruyas	N/I	N/I	no	no	sin identificar
Obladi Oblada	2.17	325	si	si	1er intento, primer sugerencia
Otherside	2.64	485	si	si	1er intento, primera sugerencia
Paint it black	2.36	539	si	si	1er intento, segunda sugerencia
Reptilia	3.66	741	si	si	1er intento, segunda sugerencia
Ruby	3.31	1284	si	si	1er intento, primer sugerencia

Run to the hills	N/I	1834	no	no	2do intento, segunda sugerencia
Sevenation army	N/I	N/I	no	no	sin identificar
Sing	2.45	423	si	si	1er intento, segunda sugerencia
Smells like teen spirit	2.51	568	si	si	1er intento, primera sugerencia
Smoke on the water	2.67	634	si	si	1er intento, primer sugerencia
Song 2	3.65	720	si	si	1er intento, primer sugerencia
Starlight	3.62	1255	no	no	2do intento, segunda sugerencia
Sunshine of your love	4.6	1634	si	si	1erintento,segunda sugerencia
Take me out	1.83	132	si	si	1er intento, primera sugerencia
The dark of the matinee	1.3	128	si	si	1er intento, primera sugerencia
The man who sold the world	0.85	92	si	si	1er intento, primera sugerencia
Ven aquí	2.31	459	no	no	1do intento, segunda sugerencia
We found love	1.77	242	si	si	1er intento, primera sugerencia
Yellow submarine	1.73	348	si	si	1er intento, primera sugerencia
You really got me	1.12	231	si	si	1er intento, primera sugerencia
Porcentaje de éxito			**68%**	**68%**	

Tablas de evaluación del sistema monofónico con compresión.

1.-

Canción	Distancia 1
Afuera	0.5
Another Brick In The Wall	0.68
Black Night	0.4
Blue Monday	0.42
Break up song	0.5
Breaking the law	0.19
Can't change me	0.58
Can't take my eyes off you	0.67
Come as you are	0.29
Dreaming of you	0.77
El Baile y el salón	0.69
Evil	0.44
Fat lip	0.69
Happybirthday	0.39

Highway to hell	0.64
I feel good	0.76
Imagine	0.53
In my place	0.57
Inagadda da vida	0.25
Inhale	0.3
Ironman	x
Jeremy	0.77
Jingle bells	x
La cucaracha	0.55
Last kiss	0.63
Last nite	0.26
Love me two times	x
Munich	0.2
Ninth Symphony	0.23
No me destruyas	x
Obladi Oblada	0.7
Otherside	0.36
Paint it black	0.09
Reptilia	0.57
Ruby	0.8
Run to the hills	0.48
Sevenation army	0.49
Sing	0.6
Smells like teen spirit	0.6
Smoke on the wáter	0.44
Song 2	0.71
Starlight	x
Sunshine of your love	0.32
Take me out	x
The dark of the matinee	0.45
The man who sold the world	x
Ven aquí	0.51
We found love	0.54
Yellow submarine	0.65
You really got me	0.67
Porcentaje de éxito	**86%**

2.-

Canción	Distancia 1
Afuera	0.48
Another Brick In The Wall	0.57
Black Night	x
Blue Monday	x
Break up song	x
Breaking the law	0.29
Can't change me	0.45
Can't take my eyes off you	0.62
Come as you are	0.29
Dreaming of you	0.64
El Baile y el salón	0.88
Evil	x
Fat lip	0.64
Happybirthday	0.34
Highway to hell	0.64
I feel good	x
Imagine	0.48
In my place	0.61
Inagadda da vida	0.21
Inhale	0.22
Ironman	0.61
Jeremy	0.64
Jingle bells	x
La cucaracha	0.42
Last kiss	0.82
Last nite	0.06
Love me two times	0.52
Munich	0.16
Ninth Symphony	0.14
No me destruyas	0.56
Obladi Oblada	0.52
Otherside	0.54
Paint it black	0.1
Reptilia	0.6
Ruby	x
Run to the hills	0.62

Sevenation army	0.4
Sing	0.64
Smells like teen spirit	0.44
Smoke on the wáter	0.46
Song 2	0.81
Starlight	x
Sunshine of your love	x
Take me out	x
The dark of the matinee	0.28
The man who sold the world	0.1
Ven aquí	x
We found love	0.26
Yellow submarine	0.67
You really got me	0.5
Porcentaje de éxito	**78%**

3.-

Canción	Distancia 1
Afuera	0.59
Another Brick In The Wall	0.44
Black Night	0.37
Blue Monday	0.54
Break up song	0.54
Breaking the law	0.27
Can't change me	0.43
Can't take my eyes off you	0.58
Come as you are	0.28
Dreaming of you	0.39
El Baile y el salón	0.69
Evil	0.53
Fat lip	x
Happybirthday	0.39
Highway to hell	0.63
I feel good	0.7
Imagine	0.79
In my place	0.51
Inagadda da vida	0.46
Inhale	0.67

Ironman	x
Jeremy	0.75
Jingle bells	0.6
La cucaracha	x
Last kiss	0.55
Last nite	x
Love me two times	x
Munich	0.25
Ninth Symphony	0.17
No me destruyas	x
Obladi Oblada	x
Otherside	0.27
Paint it black	0.22
Reptilia	0.53
Ruby	x
Run to the hills	0.54
Sevenation army	0.54
Sing	0.6
Smells like teen spirit	x
Smoke on the water	0.6
Song 2	0.73
Starlight	0.3
Sunshine of your love	x
Take me out	x
The dark of the matinee	x
The man who sold the world	0.34
Ven aquí	0.38
We found love	x
Yellow submarine	0.46
You really got me	0.66
Porcentaje de éxito	**74%**

Tablas de evaluación del sistema polifónico.

1.-

Canción	Distancia 1	Distancia 2
Afuera	0.660056657	301
Another Brick In The Wall	1.767123288	684
Another Brick In The Wall	3.5	3630
Another Brick In The Wall	2.935550936	2269
Black Night	2.505295008	2811
Black Night	4.517397882	12166
Blue Monday	2.597951344	2821
Breaking the law	2	1174
Break up song	3.675540765	5176
Break up song	3.848708487	3859
Break up song	6.433447099	22300
Can't take my eyes off you	5.158455393	13936
Can't take my eyes off you	2.443349754	499
Can't take my eyes off you	2.401247401	1575
Come as you are	3.812351544	5685
La cucaracha	3.111951589	2149
La cucaracha	4.624561404	11584
Dark of the matinee	3.419724771	5181
Dreaming of you	3.002016129	3046
El baile	2.95567867	3799
Evil	3.657635468	2730
Fat lip	8.123574144	38179
Happybirthday	5.387291982	15231
Happybirthday	6.927016645	28930
Happybirthday	6.600633914	24325
Highway to Hell	4.747759283	13104
Highway to Hell	1.978233035	1590
I Feel good	4.463508323	9651
I Feel good	2.846153846	4126
Imagine	3.626062323	9790
Inagadda da Vida	4.536817102	7030
Inhale	5.284274194	12061
In my place	2.127478754	1639
Ironman	6.24112426	20011
Jeremy	4.072983355	7891

Jingle	4.596673597	6231
Last kiss	5.938540333	23364
Last Nite	2.884763124	7119
Love me two times	6.178191489	11629
Munich	3.652173913	2970
No me destruyas	3.578947368	5029
No me destruyas	3.69504644	7669
Obladi Oblada	4.60546875	5841
Obladi Oblada	2.653008963	3799
Otherside	3.727272727	7546
Paint it black	2.47759283	2400
Paint it black	3.532289628	4645
Reptilia	2.275541796	960
Ruby	8.041825095	35760
Ruby	4.580275229	7099
Run to the hills	1.928297055	1551
Sevenation army	2.692701665	2649
Sing	4.293213828	8359
Smells like teen spirit	4.065155807	8080
Smoke	2.213828425	1381
Song 2	2.339901478	2275
Starlight	5.418693982	17989
Sunshine	4.793854033	12276
Take me Out	5.399168399	10054
The man who sold the world	2.135321101	961
Ven aquí	3.09346991	4966
We found love	3.691950464	5715
We found love	5.055057618	13104
Yellow submarine	1.772087068	796
Yellow submarine	3.751600512	10030
You really got me	3.540785498	2524
Distancia mas corta	Afuera	Afuera
¿Éxito?	si	si
Canción tarareada	Afuera	

2.-

Canción	Distancia 1	Distancia 2
Afuera	1.362606232	1819
Another Brick In The Wall	1.086105675	1005
Another Brick In The Wall	3.03125	2745
Another Brick In The Wall	2.966735967	5794
Black Night	2.549167927	5380
Black Night	3.747352496	8269
Blue Monday	2.77173913	7170
Breaking the law	1.408472012	1561
Break up song	4.179700499	12019
Break up song	3.73800738	4819
Break up song	5.737201365	19144
Can't take my eyes off you	4.592391304	10030
Can't take my eyes off you	1.773399015	1245
Can't take my eyes off you	1.871101871	1050
Come as you are	3.247030879	3679
La cucaracha	2.56580938	4216
La cucaracha	3.963157895	10459
Dark of the matinee	2.940366972	3694
Dreaming of you	2.97983871	5809
El baile	2.581717452	2344
Evil	3.736453202	7369
Fat lip	7.545627376	33186
Happybirthday	4.617246596	13699
Happybirthday	6.188858696	25810
Happybirthday	5.76703645	18126
Highway to Hell	4.327445652	9505
Highway to Hell	1.44701087	660
I Feel good	4.798913043	19459
I Feel good	3.576923077	9024
Imagine	2.968838527	10576
Inagadda da Vida	4.042755344	8044
Inhale	4.883064516	10819
In my place	1.59490085	1696
Ironman	5.51035503	17515
Jeremy	3.576086957	7264
Jingle	4.191268191	5331

Last kiss	5.588315217	19509
Last Nite	2.366847826	3274
Love me two times	5.651595745	13240
Munich	3.869565217	6945
No me destruyas	2.746130031	2566
No me destruyas	2.862229102	4441
Obladi Oblada	4.37109375	6036
Obladi Oblada	2.692934783	6799
Otherside	3.016304348	6270
Paint it black	2.120923913	2281
Paint it black	3.02739726	2944
Reptilia	2.139318885	3904
Ruby	7.463878327	30481
Ruby	4.100917431	6999
Run to the hills	1.263586957	1650
Sevenation army	2	2929
Sing	3.548913043	7024
Smells like teen spirit	3.492917847	9516
Smoke	1.36548913	1065
Song 2	1.374384236	1209
Starlight	4.635869565	19939
Sunshine	4.533967391	20479
Take me Out	4.869022869	11164
The man who sold the world	1.243119266	394
Ven aquí	2.145380435	5041
We found love	2.859133127	3574
We found love	4.591032609	14116
Yellow submarine	1.774456522	2521
Yellow submarine	3.028532609	6186
You really got me	3.132930514	2974
Distancia mas corta	Another Brick in the wall	The man who sold the world
¿Éxito?	Si	no
Canción tarareada	Another Brick in the wall	

3.-

Canción	Distancia 1	Distancia 2
Afuera	3.132102273	6945
Another Brick In The Wall	3.908023483	6683
Another Brick In The Wall	3.016666667	2805
Another Brick In The Wall	5.417879418	7650
Black Night	4.43570348	12334
Black Night	2.774583964	3570
Blue Monday	5.259943182	9000
Breaking the law	3.295007564	6324
Break up song	6.111480865	14070
Break up song	4.383763838	3076
Break up song	4.540955631	6075
Can't take my eyes off you	2.779829545	3645
Can't take my eyes off you	5.054187192	6437
Can't take my eyes off you	2.347193347	3649
Come as you are	2.890736342	3143
La cucaracha	3.807866868	5619
La cucaracha	3.328070175	6634
Dark of the matinee	2.532110092	4213
Dreaming of you	5.544354839	10308
El baile	3.008310249	2929
Evil	6.140394089	9588
Fat lip	5.600760456	10849
Happybirthday	3.045385779	5011
Happybirthday	4.092329545	5850
Happybirthday	4.389857369	5761
Highway to Hell	2.961647727	4598
Highway to Hell	2.400568182	4559
I Feel good	7.372159091	20604
I Feel good	5.468934911	16495
Imagine	3.865056818	7807
Inagadda da Vida	3.349168646	4208
Inhale	3.870967742	5680
In my place	3.200284091	7341
Ironman	3.940828402	5208
Jeremy	2.475852273	2615

Jingle	3.756756757	4411
Last kiss	3.879261364	3810
Last Nite	1.774147727	4845
Love me two times	3.981382979	4438
Munich	6.791304348	10309
No me destruyas	1.950464396	3257
No me destruyas	1.973684211	3227
Obladi Oblada	3.734375	2710
Obladi Oblada	5.286931818	10026
Otherside	2.450284091	5426
Paint it black	2.369318182	3239
Paint it black	2.643835616	2955
Reptilia	4.53869969	7787
Ruby	5.150190114	14860
Ruby	2.995412844	3483
Run to the hills	3.214488636	6506
Sevenation army	3.578125	6299
Sing	2.536931818	3998
Smells like teen spirit	2.924715909	4838
Smoke	2.46875	5411
Song 2	2.600985222	4103
Starlight	3.865056818	7707
Sunshine	6.599431818	15924
Take me Out	4.550935551	4305
The man who sold the world	2.766055046	4418
Ven aquí	3.133522727	3342
We found love	2.301857585	3226
We found love	3.005681818	4575
Yellow submarine	4.427556818	7957
Yellow submarine	2.748579545	2955
You really got me	3.259818731	2824
Distancia mas corta	LastNite	Jeremy
¿Éxito?	No	no
Canción tarareada	Black night	

4.-

Canción	Distancia 1	Distancia 2
Afuera	4.462848297	11139
Another Brick In The Wall	5.111545988	9874
Another Brick In The Wall	2.53125	1470
Another Brick In The Wall	6.706860707	17386
Black Night	5.286377709	18655
Black Night	2.349845201	1899
Blue Monday	6.653250774	23914
Breaking the law	4.397832817	7071
Break up song	7.159733777	25489
Break up song	2.335793358	1434
Break up song	3.226962457	4486
Can't take my eyes off you	1.787925697	1095
Can't take my eyes off you	6.359605911	13534
Can't take my eyes off you	2.812889813	2829
Come as you are	1.74584323	435
La cucaracha	4.46749226	11121
La cucaracha	3.173684211	4851
Dark of the matinee	2.520642202	1786
Dreaming of you	6.461693548	17425
El baile	2.246537396	706
Evil	7.435960591	22186
Fat lip	4.891634981	10189
Happybirthday	2.185758514	1399
Happybirthday	2.905572755	1144
Happybirthday	2.930269414	2251
Highway to Hell	2.182662539	630
Highway to Hell	2.933436533	3565
I Feel good	8.328173375	34480
I Feel good	6.815789474	23119
Imagine	3.280185759	4096
Inagadda da Vida	3.111638955	1390
Inhale	2.973790323	2215
In my place	3.842105263	10834
Ironman	2.325077399	964
Jeremy	2.23374613	2664
Jingle	2.002079002	1494

Last kiss	2.651702786	1719
Last Nite	2.094427245	2649
Love me two times	2.558510638	1429
Munich	8.347826087	25650
No me destruyas	1.998452012	1306
No me destruyas	1.92879257	1231
Obladi Oblada	1.890625	991
Obladi Oblada	6.371517028	17736
Otherside	2.465944272	1749
Paint it black	2.489164087	2229
Paint it black	1.937377691	825
Reptilia	5.577399381	16089
Ruby	5.038022814	11755
Ruby	1.823394495	300
Run to the hills	4.187306502	11215
Sevenation army	4.39628483	7900
Sing	1.789473684	1156
Smells like teen spirit	2.626934985	3994
Smoke	3.1625387	5679
Song 2	3.660098522	2911
Starlight	3.602167183	3769
Sunshine	7.930340557	40519
Take me Out	2.995841996	3901
The man who sold the world	3.51146789	4066
Ven aquí	3.159442724	3631
We found love	1.931888545	1659
We found love	2.53250774	1351
Yellow submarine	5.791021672	16101
Yellow submarine	1.955108359	2094
You really got me	3.764350453	3211
Distancia mas corta	Come as you are	Ruby
¿Éxito?	si	no
Cancion tarareada	Come as you are	
Cancion	Distancia 1	Distancia 2
Afuera	2.150141643	1479
Another Brick In The Wall	2.26223092	1096
Another Brick In The Wall	2.59375	1785
Another Brick In The Wall	4.280665281	3181
Black Night	2.641452345	3351

Black Night	3.6096823	6016
Blue Monday	3.311960543	3256
Breaking the law	2.226928896	1714
Break up song	4.525790349	6400
Break up song	3.464944649	3951
Break up song	5.433447099	12256
Can't take my eyes off you	4.436750999	5719
Can't take my eyes off you	2.844827586	900
Can't take my eyes off you	2.311850312	874
Come as you are	2.35391924	2086
La cucaracha	2.703479576	2119
La cucaracha	3.887719298	8029
Dark of the matinee	2.594036697	2886
Dreaming of you	3.036290323	2466
El baile	2.12465374	1699
Evil	3.699507389	2464
Fat lip	7.003802281	25356
Happybirthday	4.524962179	11871
Happybirthday	6.16464891	21334
Happybirthday	5.435816165	17080
Highway to Hell	3.796610169	5731
Highway to Hell	1.543583535	990
I Feel good	5.192211055	12019
I Feel good	3.644970414	4516
Imagine	2.991501416	9234
Inagadda da Vida	3.862232779	8226
Inhale	4.100806452	9726
In my place	1.57223796	1470
Ironman	5.353550296	11851
Jeremy	3.561743341	3994
Jingle	3.384615385	4024
Last kiss	4.92251816	11911
Last Nite	2.543583535	4066
Love me two times	5.053191489	11185
Munich	4.826086957	2820
No me destruyas	2.721362229	2739
No me destruyas	2.8374613	4779
Obladi Oblada	3.203125	3805
Obladi Oblada	3.130702836	4621

Otherside	3.308716707	4599
Paint it black	2.290556901	2254
Paint it black	2.442270059	2469
Reptilia	2.904024768	1441
Ruby	7.207224335	24481
Ruby	3.27293578	3364
Run to the hills	2.134382567	4444
Sevenation army	2.585956416	3996
Sing	3.524213075	5821
Smells like teen spirit	3.940509915	6889
Smoke	2.186440678	3096
Song 2	3.036945813	939
Starlight	4.376513317	11280
Sunshine	5.078692494	17635
Take me Out	4.374220374	6361
The man who sold the world	1.791284404	901
Ven aquí	3.002421308	3550
We found love	2.880804954	3736
We found love	4.303457106	6781
Yellow submarine	2.52905569	2386
Yellow submarine	2.927360775	5244
You really got me	3.634441088	1944
Distancia mas corta	Highway to hell	Can't take my eyes off you
¿Éxito?	si	no
Canción tarareada	Highway to hell	

Bibliografía.

[1] Google Mobile. [Online]. Disponible en: http://www.google.com/xhtml

[2] Live Search Mobile. [Online]. Disponible: http://www.mobile.live.com/search

[3] Yahoo! Mobile. [Online]. Disponible: http://www.mobile.yahoo.com/onesearch

[4] www.sonyericsson.com/**trackid**

[5] A Query By Example Music Retrieval Algorithm H. Harb AND L. Chen Maths-Info department, Ecole Centrale de Lyon. 36, av. Guy de Collongue, 69134, Ecully, France, EUROPE. E-mail: {hadi.harb, liming.chen}@ec-lyon.fr

[6] http://www.midomi.com/

[7] Y. Watanabe, K. Sono, K. Yokoizo, and Y. Okada, BTranslation camera on mobile phone,[in Proc. IEEE Int. Conf. Multimedia Expo, Baltimore, MD, Jul. 2003.

[8] M. Smith, D. Duncan, and H. Howard, BAURA: A mobile platform for object and location annotation,[in Proc. 5th Int.Conf. Ubiquitous Computing, Seattle, WA, Oct. 2003.

[9] Analysis of Sound Features for Music Timbre Recognition**,** Xin Zhang and Zbigniew W. Ras, Department of Computer ScienceUniversity of North Carolina at Charlotte. International Conference on Multimedia and Ubiquitous Engineering, 2007

[10] Mel Frecuency Cepstral Coefficients for Music Modeling, Beth Logan, Cambridge Research Laboratory, Compaq Computer Corporation, One Cambrige Center, Cambridge MA 02142.

[11] A new approach to query by humming in music retrieval, Lie Lu, Hong You, Hong-Jiang Zhang, Microsoft Research China, 5F, Beijing 100080 {i-lielu, hjzhang}@microsoft.com.

[12] A Query by Humming System for Music Information Retrieval, Mario Antonelli, Antonello Rizzi, Guido del Vescovo, University of Rome ”La Sapienza” INFOCOM Department via Eudossiana, 18, 00184 Rome, Italy. Email: mario.antonelli@gmail.com, Email: rizzi@infocom.uniroma1.it, Email: guido.delvescovo@gmail.com.

[13] MPEG-7 Audio and Beyond, audio content indexing and retrieval, Hyoung-Gook Kim, Nicolas Moreau, Thomas Siroka, Samsung Advanced Institute of technology, Korea, Technical University of Berlin, Germany, Communication System Group, Technical University of Berlin, Germany. John Wiley & Sons Ltd. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ. England. Copyright 2005.

[14] Digital Signal Processing, Third edition, John G. Proakis, Dimitris G. Manokalis, North Eastern, Boston College, 1996 Prentice Hall Inc.

[15] Information Retrieval for Music and Motion, Meinard Müller Institut für Informatik III Universität Bonn Römerstr. 164. 53117 Bonn, Germany meinard@cs.uni-bonn.de

[16]. M. Clausen and U. Baum, Fast Fourier Transforms, BI Wissenschaftsverlag, 1993.

[17] Digital Signal Processing, Schaum's Outline Series, Monson H. Hayes Professor of Electrical and Computer Engineering Georgia Institute of Technology. Copyright © 1999 by The McGraw-Hill Companies, Inc.

[18] MusicXML, A universal translator for common western musical notation. Recordare, http://www.recordare.com/xml.html, 2006.

[19] www.guitarpro.com

[20] www.sibelius.com

[21] MIDI, Musical Instrument Digital Interface. Manufacturers Association. http://www.midi.org/ 2012.

[22] www.adobe.com/es/products/audition

[23] www.steinberg.net/cubase

[24]Richard Lyon and Shihab Shamma (1996). "Auditory Representation of Timbre and Pitch". In Harold L. Hawkins and Teresa A. McMullen. *Auditory Computation*. Springer. p. 221–223. ISBN 9780387978437.

[25] Humming Method for Content-Based Music Information Retrival. Cristina de la Bandera, Ana M. Barbancho, Lorenzo J. Tardón, Simone Sammartino and Isabel Barbancho Dept. Ingeniería de Comunicaciones, E.T.S. Ingeniería de Telecomunicación Universidad de Málaga, Campus Universitario de Teatinos s/n, 29071, Málaga, Spain

[26] http://www.music-ir.org/mirex/wiki/MIREX_HOME

[27] Application Of Neural Network To Technical Analysis Of Stock Market Prediction Hirotaka Mizuno, Michitaka Kosaka, Hiroshi Yajima, Systems Development Laboratory Hitachi, Ltd. 8-3-45 Nankouhigashi, Suminoe-Ku Osaka 559-8515 JAPAN Norihisa Komoda Department of Information Systems Engineering Faculty of Engineering, Osaka University 2-1 Yamadaoka, Suita Osaka 565-0871 JAPAN

[28] Medical Diagnosis Using Neural Networks S. M. Kamruzzaman, Ahmed Ryadh Hasan†, Abu Bakar Siddiquee and Md. Ehsanul Hoque Mazumder
Department of Computer Science and Engineering International Islamic University Chittagong, Chittagong-4203, Bangladesh Email: smk_iiuc@yahoo.com, maksud_cse@yahoo.com, sumon_ctg2003@yahoo.com School of Communication Independent University Bangladesh, Chittagong, Bangladesh, Email: ryadh78@yahoo.com

[29] Song-Level Features and Support Vector Machine For Music Classification Michael I. Mandel and Daniel P.W. Ellis LabROSA, Dept. of Elec. Eng., Columbia University, NY NY USA fmim,dpweg@ee.columbia.edu

[30] Digital Image Processing Second Edition, Rafael C. Gonzalez, Paul Wintz, Electrical Engineering Department University of Tennessee Knoxville and Perceptics Corporation Knoxville, Tennessee, Addison-Wesley Publishing Company.

[31] http://developer.android.com

[32] https://developer.apple.com