

Un algoritmo eficiente para similitud melódica en los *cantes* flamencos

Proyecto Fin de Carrera

Juan Carlos Rizo Massia
Universidad de Sevilla, Enero 2014

Dirigido por Dr. D. José Miguel Díaz Báñez
Dpto. Matemática Aplicada II



Ingeniero Industrial

ÍNDICE

Pág.

Capítulo 1: Introducción

1.1 Motivación y Marco de trabajo	3
1.2 Objetivo	5

Capítulo 2: Teoría computacional y flamenco

2.1 Antecedentes	9
2.2 Similitud melódica	15
2.2.1 Distancia de edición	17
2.2.2 Subsecuencia común de mayor longitud	19
2.2.3 Dynamic Time Warping (DTW)	20

Capítulo 3: El algoritmo SDTW

3.1 Dynamic Time Warping (DTW)	25
3.1.1 Introducción	25
3.1.2 Definición de la medida de similitud DTW	26
3.1.3 Restricciones en las trayectorias de alineamiento	28
3.1.4 Algoritmo de programación dinámica	30
3.1.5 Complejidad del algoritmo	32
3.1.6 Algoritmos de aproximación del método DTW	32
3.2 Métodos de segmentación	35
3.2.1 SmsTools 2	36
3.2.2 Algoritmo Fitting	40
3.3 La estrategia combinada	41

ÍNDICE

Pág.

Capítulo 4: Evaluación

4.1	Corpus: Tonás	61
4.1.1	Introducción	61
4.1.2	Los cantes por Tonás para un estudio computacional	62
4.1.3	Descripción y características musicales: debla y martinete	63
4.1.4	El Corpus Tonás: Un metadato de acceso libre	66
4.2	Medidas de rendimiento	67
4.2.1	Análisis filogenético: Generación de grafos	67
4.2.2	Test de Mantel	70
4.2.3	Precisión y efectividad	71
4.3	Evaluación empírica: Precisión y eficiencia	75

Capítulo 5: Conclusiones

5.1	Aportaciones del proyecto	91
5.2	Posibles líneas futuras de investigación	93

Referencias

97

Anexo: Códigos Matlab

107

Capítulo 1:

INTRODUCCIÓN

1. Motivación y Marco de trabajo.
2. Objetivo.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Motivación y Marco de trabajo

El flamenco es sin lugar a duda una de las manifestaciones culturales más importantes de España, así como el principal referente de nuestra cultura por todos los rincones del mundo. Originado y desarrollado en Andalucía, siempre ha sido una expresión artística de clara tradición oral, por tratarse en sus orígenes de una manifestación popular. Este hecho, unido a que los flamencólogos (expertos en música flamenca) no han seguido tradicionalmente una preparación académica estándar, ha provocado un reducido acercamiento del método científico a todo lo relacionado con el flamenco, por lo que la cantidad y calidad de la investigación existente en torno al flamenco es bastante limitada.

Debido a la escasa investigación científica en la musicología del flamenco, se requieren distintas fases de actuación para su puesta en valor en la comunidad científica, que abarcan desde la creación de un corpus de transcripciones o anotaciones musicales significativo hasta, el diseño de algoritmos eficientes para calcular distancias entre dos melodías. Entre los aspectos fundamentales que demandan avances pueden destacarse: los orígenes del flamenco, evolución y relación entre los distintos estilos, propiedades de preferencia de estilos, influencia de músicas externas a Andalucía o búsqueda de estilos ancestrales.

La industria de la música fue una de las primeras en verse completamente reestructurada por los avances de la tecnología digital, y hoy en día tenemos acceso a miles de canciones almacenadas en nuestros dispositivos móviles y a millones más a través de servicios en internet. Dada esta inmensa cantidad de música a nuestro alcance, necesitamos nuevas maneras de describir, indexar, buscar e interactuar con el contenido musical. Desgraciadamente, las aplicaciones informáticas existentes en la actualidad no suelen ser útiles para el contexto musical flamenco, por lo que se requiere adaptar o diseñar nuevos algoritmos que abran las puertas del mundo tecnológico a la música flamenca. Este es el objetivo del grupo de investigación COFLA¹, que ha cimentado el área de investigación Teoría Computacional del flamenco, marco donde se desarrolla este proyecto. El grupo realiza actualmente las tareas en el proyecto de excelencia COFLA: Análisis Computacional de la Música Flamenca (P09-TIC-4840).

Las técnicas computacionales y modelos matemáticos utilizados en el área de investigación de la Teoría Computacional del flamenco, pertenecen al campo de la Music Information Retrieval (MIR) (Orio, 2006), ciencia interdisciplinar que se dedica a la recuperación de información de la música. MIR es un campo de conocimiento en

¹ <http://mtg.upf.edu/research/projects/cofla>

crecimiento que involucra a distintas ramas de conocimiento como pueden ser: la psicología, la formación musical académica, tratamiento de señal, etc.

Las aplicaciones del MIR pueden ser:

- Sistemas de recomendación musical.
- Separación y reconocimiento de instrumentación.
- Transcripción automática.
- Caracterización automática.

Desde los comienzos del MIR como campo de conocimiento, la mayoría de las técnicas y modelos han sido desarrollados para una corriente particular de la música tradicional, la conocida como música popular de tradición occidental. En los últimos años ha surgido un especial interés en aplicar las técnicas desarrolladas por el MIR para el estudio de otras músicas tradicionales, folk o música étnica (ISMIR)², ya que etnomusicólogos, musicólogos y otros profesionales de la música podrían beneficiarse considerablemente de esta investigación. Aunque las técnicas computacionales han demostrado ser de gran interés cuando se aplica a diferentes repertorios musicales, es evidente que se deben desarrollar técnicas y algoritmos matemáticos específicos para comprender, modelar y procesar diferentes repertorios musicales.

Debido a que el flamenco es una música de tradición oral, está fuertemente ligada a la cultura del entorno donde nace, en este caso la historia y cultura andaluza. Por ello, no debe realizarse una investigación científica ajena al contexto cultural y las implicaciones que esto supone en el desarrollo y sentido de la música flamenca. De hecho, existe un interés reciente por el estudio tecnológico interdisciplinar de músicas de tradición oral, lo que ha llevado a la consideración de una nueva área de investigación conocida como Etnomusicología Computacional. La Etnomusicología Computacional se remonta a 1978 (Halmos, 1978) cuando Halmos, Köszegi y Mandl, dos matemáticos y un ingeniero proporcionaron una interesante reflexión sobre el papel de los ordenadores en las cinco principales áreas de investigación en Etnomusicología: recopilación de datos, administración, notación, selección y sistematización, y tratamiento científico. El término fue recientemente redefinido por Tzanetakis (Tzanetakis, Kapur, Schloss, & Wright, 2007) en 2007 como "el diseño, el desarrollo y la utilización de herramientas informáticas que tienen el potencial para ayudar en la investigación etnomusicológica". Con este enfoque se ve a la Etnomusicología como la base científica que se sirve de los avances en otras disciplinas de apoyo tales como Matemáticas Computacionales, Musicología o estudios de carácter cultural. Cuando nos acercamos a la Etnomusicología Computacional de esta manera, se adopta un nuevo marco mental que ayuda a reestructurar problemas y percibir las relaciones entre sus elementos bajo una perspectiva diferente.

² The International Society for Music Information Retrieval (www.ismir.net)

1.2 Objetivo

Muchos problemas relacionados con la teoría y tecnología musical son fundamentalmente geométricos por naturaleza, esto es, miden alguna característica que contiene el corpus musical (citamos al prof. Toussaint (Toussaint, 2010) como trabajo reciente que encabeza la lista de los artículos que despiertan más interés en la revista *Comp. Geom. Theory and Applications*). Pongamos un ejemplo: dos melodías pueden ser representadas por poligonales ortogonales (funciones escalón) y una posible medida de similitud es el área comprendida entre las dos curvas (permitiendo traslaciones verticales y horizontales). De esta forma, el problema se traslada al campo de Visión Computacional, donde coincide con el problema de emparejamiento de formas poligonales (Aloupis, et al., 2006). Un buen trabajo donde se revisan medidas de similitud melódica es (Hewlett & Selfridge-Field, 1998).

Este proyecto se centra en un problema fundamental del Análisis Computacional de la Música: la *similitud melódica*. La trascendencia de la similitud melódica ha sido, y es, tal que numerosas disciplinas se han ocupado intensivamente de su estudio:

- En Etnomusicología, por ejemplo, para entender la lógica musical, para evaluar los estilos y sus características, para conocer los criterios de improvisación. (Bartók & Lord, 1951), (Seeger, 1966), (Holzapfel, 2010).
- En Análisis Musical, para construir modelos tanto teóricos como computacionales. (Lerdahl & Jackendoff, 1983), (Meyer, 1973).
- En la resolución de conflictos de propiedad intelectual (Cronin, 1998).
- En Tecnología Musical, para aplicar los modelos obtenidos tras el correspondiente análisis. (Holzapfel, 2010).
- En Psicología de la música, para comprender mejor el hecho musical, para aportar conocimiento a un análisis integral de la música. (Hahn, Chater, & Richardson, 2003).

La investigación existente en similitud musical está centrada fundamentalmente en música occidental y en los sistemas actuales de recomendación, es esta música la que se encuentra etiquetada. Sin embargo, existe un creciente interés en el campo para analizar y etiquetar músicas tradicionales y folclóricas.

En este proyecto trabajaremos en el contexto de la música tradicional más importante en Andalucía y, que posee además, gran interés internacional. La música flamenca es fundamentalmente de tradición oral, lo que dificulta, y hace más interesante a la vez, su estudio científico. La conservación de esta música de generación en generación hace que la melodía juegue un papel crucial en la evolución y clasificación de los distintos estilos del flamenco, siendo éste uno de los problemas abiertos más apasionantes del área. Puesto que el cante flamenco no ha sido llevado a partituras (al

menos hasta ahora no existe consenso sobre la transcripción del cante flamenco), trabajaremos con datos sonoros y se hará uso de un sistema de transcripción automática que se requiere implementar tal como los existentes para otras músicas (Klapuri & Davy, 2006).

Teniendo en cuenta lo anterior, en este proyecto fin de carrera pretendemos dar un nuevo paso que permita realizar una aproximación al flamenco a través del uso de técnicas computacionales con objeto de analizar con más profundidad y con rigor científico este milenario arte. El objetivo concreto será el de diseño de un nuevo método de análisis automático de las formas melódicas flamencas, estableciendo un algoritmo más eficaz y eficiente que los actuales que aporte una medida de similitud melódica, para poder así servir de métrica entre los diferentes estilos e interpretaciones. En concreto, vamos a considerar dos estilos flamencos que corresponden al grupo de cantes a capela del flamenco, las deblas y los martinetes. Aunque en este proyecto se desarrolla una metodología adaptada a estos cantes a capella, puede llevarse a cabo una estrategia similar para otros estilos musicales.

Capítulo 2:

TEORÍA COMPUTACIONAL Y FLAMENCO

1. Antecedentes.
2. Similitud melódica.

CAPÍTULO 2: TEORÍA COMPUTACIONAL Y FLAMENCO

2.1 Antecedentes

Pese a que el flamenco posee un gran peso específico en el marco cultural español y un gran impacto mediático internacional, no ha sido hasta hace relativamente poco tiempo, cuando se ha querido someter al flamenco al irrevocable juicio de la ciencia. Posiblemente este hecho ha estado motivado por el carácter anárquico y sentimental del propio *cante*, ya que aunque sus formas y estilos están claramente diferenciados unos de otros, la tradición oral ha hecho que éstos se superpongan y mezclen de tal forma que resulta muy complejo establecer un patrón fijo e inamovible de cada palo que permita a una simple función matemática ser capaz de diferenciar un palo del resto, olvidando por completo la intención, y lo que es más importante, el fondo que el *cantaor* quiere transmitir con su *cante*.

La Teoría Computacional de la Música es un campo de relativa reciente creación. La música y las matemáticas han estado relacionadas desde al menos 2.500 años, cuando Pitágoras descubrió que los sonidos más consonantes venían de proporciones simples y pequeñas. Desde entonces, la interacción entre música y matemáticas no ha hecho sino crecer. Y ha sido en las últimas décadas cuando se ha producido una revitalización como consecuencia de la introducción de los métodos computacionales en el análisis de la música.

Es en 2005, a raíz de un estudio sobre un análisis filogenético del compás flamenco: “*Similaridad y evolución en la rítmica del flamenco: una incursión de la matemática computacional*”, cuando la comunidad científica aborda el análisis de la música flamenca desde la perspectiva de la Teoría Computacional y las Matemáticas. El citado trabajo se publica en la revista *La Gaceta de la Real Sociedad Matemática Española* y su versión en inglés en los Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science, Winfield, Kansas, 61-70, 2004. En este estudio, pionero en lo que los medios no tardaron en denominar como la *Teoría Computacional de la Música Flamenca*¹, se analizan aspectos rítmicos de los estilos ternarios del flamenco y se extraen algunos indicios sobre genealogía y preferencia de estilos en lo que respecta a patrones rítmicos. En dicho trabajo, y a modo de conclusiones, se insta a musicólogos y otros investigadores del flamenco a contrastar este tipo de análisis científico con otras metodologías usadas en sus áreas de investigación, a saber, Historia, Antropología o Teoría Musical, creándose así el germe para la creación del primer grupo de investigación multidisciplinar de Teoría computacional de la música flamenca. Debido a que el problema tratado en este proyecto coincide con una de las líneas de trabajo del

¹ véase http://www.elpais.es/articulo/elpfutpor/20050907elpepifut_6/Tes/

grupo, incluimos aquí una perspectiva histórica del grupo de investigación COFLA para entender la motivación del mismo.

En el año 2007 nace el primer proyecto de investigación relacionado con el flamenco, el proyecto COFLA (Análisis Computacional de la Música Flamenca) que surge como respuesta a las inquietudes de un grupo de investigadores, encabezados por el Dr. José Miguel Díaz Báñez, responsable del proyecto y tutor de este proyecto fin de carrera. Está subvencionado por la Junta de Andalucía y su objetivo principal es el de realizar una investigación exhaustiva y rigurosa de la música flamenca mediante un enfoque multidisciplinar, combinando los conocimientos de áreas como la literatura, la musicología, la psicología musical, las matemáticas, la recuperación de información musical (MIR) o el procesamiento de señales de audio digital, entre otros. Asimismo, se persigue fortalecer el estudio del flamenco como una cuestión académica en la universidad y dejar las bases de una nueva área de investigación, la Teoría Computacional del flamenco.

Además, los objetivos que persigue el proyecto COFLA aparecen recogidos en la Reforma del Estatuto de Autonomía de Andalucía², aprobada en referéndum el 18 de Febrero de 2007, que en su Título II, Capítulo II, Artículo 68, dice lo siguiente: “Corresponde asimismo a la Comunidad Autónoma la competencia exclusiva en materia de conocimiento, conservación, investigación, formación, promoción y difusión del flamenco como elemento singular del patrimonio cultural andaluz”.

Entre los aspectos fundamentales en los que el grupo pretende indagar pueden destacarse: los orígenes del flamenco, evolución y relación entre los distintos estilos, propiedades de preferencia de estilos, influencia de músicas externas a Andalucía o búsqueda de estilos ancestrales. Todo ello mediante un estudio analítico de las estructuras musicales flamencas usando técnicas ya exploradas en otras músicas, que se basan en el uso herramientas computacionales.

Es en julio de 2008, en la 4ª Conferencia sobre Musicología Interdisciplinar (CIM '08) en Tesalónica, cuando se presenta ante la comunidad científica el primer trabajo del grupo COFLA. En este trabajo se comparan los diferentes enfoques sobre el análisis melódico y tonal de los cantes flamencos a capela, para poder así llevar a cabo un estudio multidisciplinar que proporcione las herramientas necesarias para comparar diferentes versiones del mismo estilo. Para ello, se realizó un examen melódico y tonal manual, y automático pero asistido por herramientas de descripción melódica automática. Con el fin de cubrir los objetivos de estudio, también fue necesario reunir un corpus musical representativo compuesto por doce martinetes y doce deblas, interpretados cada uno de ellos por diferentes cantaores. Esto supuso una ardua tarea debido a la variedad de soportes (sobre todo de vinilos y cintas magnéticas) y a la mala

² Junta de Andalucía BOJA nº 12 de 17 de Enero, 2007

calidad de las grabaciones existentes (Cabrera, Díaz-Bañez, Escobar-Borrego, Gómez, & Mora, 2008). Fue también en esta conferencia donde se presentó por parte del grupo una nueva herramienta para la transcripción melódica automática de piezas flamencas, centrándose su aplicación en fragmentos monofónicos a capela. Esta herramienta ayudará a la búsqueda de un nivel de representación adecuado para las características del flamenco y para el análisis comparativo de los cantes y estilos (Gómez & Bonada, 2008).

Continuando con las inquietudes del grupo en torno a la investigación sobre el Arte flamenco, así como la necesidad de protección, conservación, tratamiento documental y difusión del patrimonio cultural andaluz, se crea en junio de 2009 el primer Congreso Interdisciplinar de Investigación y flamenco (INFLA), organizado por los miembros del grupo COFLA en la Escuela Superior de Ingeniería de Sevilla, donde el formato de este evento coincide con el de los congresos universitarios en áreas tradicionales como matemáticas o ingeniería, esto es, se abre la participación a cualquier investigador y aceptan los trabajos tras ser revisados por expertos. El objetivo fundamental que se persigue en este evento es crear un espacio universitario que constituya un motor de investigación científica en torno a la música flamenca, entendiendo que ésta se encuentra interrelacionada con otras músicas y ciencias, cuyos enfoques constituyen un tratamiento riguroso desde el punto de vista académico. En este marco se ubica también el Programa de Doctorado *El flamenco. Acercamiento multidisciplinar a su estudio* de la Universidad de Sevilla.

Tras el primer trabajo, el grupo detecta una limitación en los modelos de cálculo estándar para la caracterización melódica y computacional de la similitud en el contexto particular de la música flamenca. Es por eso que realiza una investigación sobre la relación entre la similitud musical y el estilo, y la clasificación sub-estilo, en los cantes flamencos a capela. Con el fin de evaluar las características melódicas propuestas, se crea una colección de grabaciones de los cantaores más representativos (maestros) de los cuales se han extraído manualmente las características propuestas de dos tipos de cantes *deblas* y *martinetes*, creando además una diferenciación intra-estilo al diferenciar entre dos tipos de *martinetes* (llamados en el trabajo *martientes1* y *martienetes2*). De cada tipo de cante se eligieron distintas características diferenciadoras, por ejemplo, en las *deblas*, se fijan como características propias del cante: el comienzo con la palabra ¡Ayi, la duración, la nota inicial, etc, mientras que en los *martinetes* se fijan la repetición del primer hemistiquio, la simetría en el segundo hemistiquio o la nota final. En base a estas características, se definió una medida de similitud basada en el concepto de distancia de edición (Mongeau & Sankoff, 1990) en la que se ponderan las características propias de la música flamenca según su contorno musical (distancia MC) y según un descriptor musical (distancia MD) obtenido a partir de las características fijadas. Dicha ponderación se realiza de la forma $d_I = \alpha \cdot d_{MC} + (1 - \alpha) \cdot d_{MD}$, donde α determina el peso de cada métrica, obteniendo finalmente un

clustering o agrupación ,y unos gráficos filogenéticos (ver Figura 1) obtenidos a partir de la matriz de distancias creadas con las medidas de similitud obtenidas en las evaluaciones a los distintos cantes.(Mora, Gómez, Gómez, Escobar-Borrego, & Díaz-Bañez, 2010).Este trabajo es aceptado en la prestigiosa conferencia International Society for Music Information Retrieval Conference (ISMIR).

Durante 2010, la actividad del grupo continúa activa con la cita del II Congreso INFLA, celebrado en septiembre de 2010, con relevancia especial ese año, puesto que coincidía con la *Bienal de flamenco* de Sevilla, por lo que el congreso se convirtió en una actividad paralela.

Tras meses de trabajo surge por parte de algunos investigadores del grupo la necesidad de realizar un análisis sobre la ornamentación de la música flamenca, ya que aunque en su concepción está cargada de adornos y melisma, no existía ninguna investigación al respecto, hecho que no ocurría con otras músicas en las que la ornamentación ha sido tabulada, como en la música clásica occidental. Con este estudio se definirá el papel estructural que los melismas dentro del cante flamenco, y se verá como estos ocupan un lugar importante para la definición de los estilos. La metodología llevada a cabo fue tomar como punto de partida un conjunto de adornos tomados principalmente de la música clásica, para tras seleccionar un corpus flamenco compuesto de grabaciones de audio, se examinó el conjunto de adornos prefijado. Para hacer esto de forma automática se diseño un algoritmo de detección de patrones, basado en una adaptación del algoritmo de Smith-Waterman. (Smith &Waterman, 1981). Los resultados obtenidos fueron prometedores y constituyen un primer paso hacia un estudio sistemático de la ornamentación en la música flamenca (Gómez, Pikrakis, Mora, Díaz-Bañez, & Gómez, 2011). Este trabajo fue presentado en el 4th International Workshop on Machine Learning and Music: Learning from Musical Structure, en Granada.

Los objetivos que plantea el grupo COFLA para 2012, son: extracción manual de aspectos armónicos del cante, extracción manual frente a extracción automática de las melodías del cante, buscando siempre la forma más efectiva de extracción automática y a la vez más fiable, ampliación del corpus de deblas y martinetes inicial, análisis de otros estilos, y aplicación de nuevas medidas de similitud melódica.

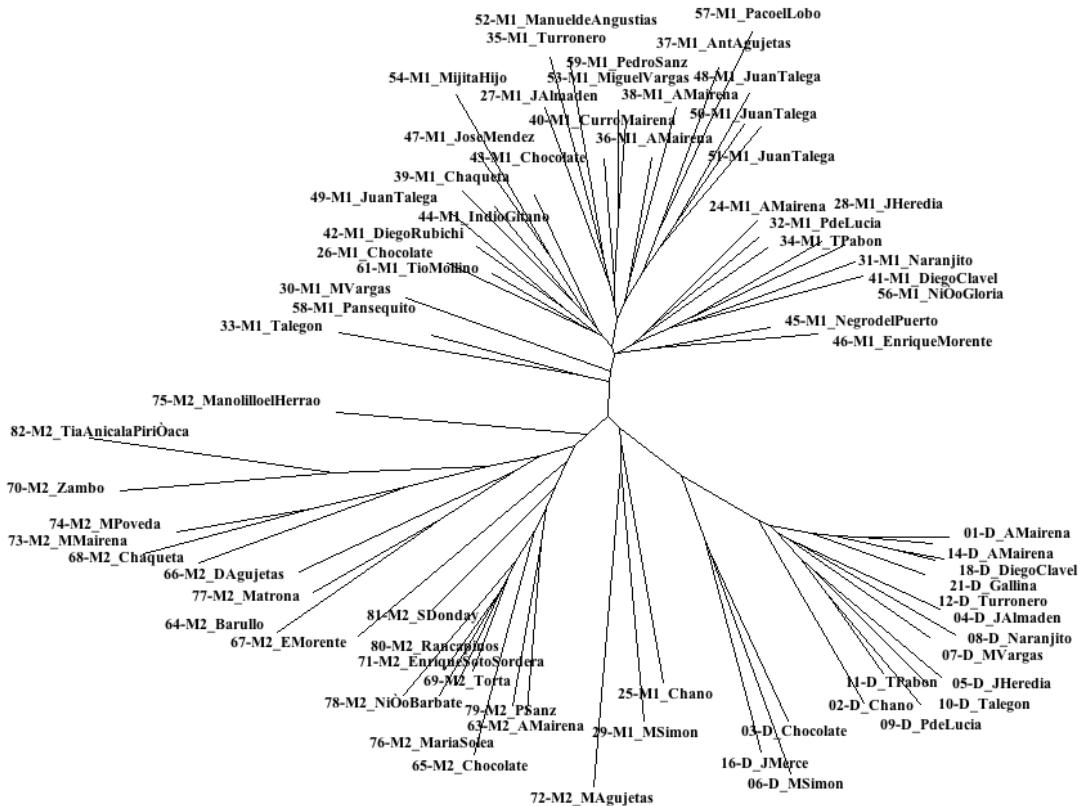


Figura 1: Árbol filogenético para las distancias MC.

En abril de 2012, el III Congreso INFLA se celebró conjuntamente con la segunda edición del Encuentro Internacional de Análisis de Música Folklórica (Folk Music Analysis, FMA en sus siglas inglesas), donde el grupo presenta dos de sus trabajos. En el primer trabajo, siguiendo el camino iniciado iniciado en 2008 por el trabajo Automatic Melodic Transcription of flamenco Singing (Gómez & Bonada, 2008), son los autores del mismo los que presentan una versión mejorada y ampliada del citado trabajo. En él se propone un nuevo método para la transcripción automática de los cantes extraídos del nuevo corpus de grabaciones polifónicas y monofónicas, compuesto por 72 piezas representativas de diferentes cantes a capela (tonás) monofónicos, y por 20 fragmentos de una duración media de 57 segundos de música polifónica (con acompañamiento de guitarra), en este caso fandangos. El sistema de transcripción usado sistemas de transcripción se basa en la estimación de la frecuencia fundamental (f_0) de la voz cantada, y sigue una estrategia iterativa para la segmentación y etiquetado de la nota. Las transcripciones generadas se utilizan en el contexto de la similitud melódica, clasificación de estilos y detección de patrones. En este estudio, se discuten las dificultades encontradas en la transcripción del cante y en la evaluación de las transcripciones obtenidas, además se analiza la influencia de los diferentes pasos del algoritmo, y se fijan las principales limitaciones del enfoque tomado discutiendo los desafíos adquiridos para futuros estudios. (Gómez, Bonada, & Salomon, 2012).

El segundo trabajo presentado por el grupo en el congreso se centra en la localización de patrones melódicos predefinidos y diferenciadores de estilos, en un corpus de archivos de audio sobre fandangos de Huelva. Se define un conjunto de algoritmos de detección de patrones melódicos siguiendo un método que cuenta con dos etapas: en la primera, se extrae la curva melódica de la voz de las grabaciones de audio por medio de una técnica de estimación de frecuencia fundamental predominante (Gómez, Bonada, & Salomon, 2012), y en segundo lugar, los patrones se detectan directamente en las curvas melódicas por medio de un algoritmo de programación dinámica adaptado que se ha desarrollado para músicas de tradición oral griegas (Pikrakis, Theodoridis, & Kamarotos, 2003).

Para probar el método propuesto, se realizó un análisis de los patrones melódicos característicos en el contexto del estilo de los fandangos. Con este fin, se definieron una serie de patrones en formato simbólico por expertos en música flamenca y después se constituyó un corpus, compuesto de grabaciones de audio tomadas de dos estilos de fandango, fandangos de Valverde y fandangos de Huelva capital. Estos dos estilos son representativos de la tradición del fandango. Por último, la estrategia en la evaluación del rendimiento de los algoritmos fue discutida por expertos en flamenco y sus conclusiones presentadas en este trabajo. (Gómez F. , Pikrakis, Mora, Diaz-Bañez, Gómez, & Escobar, 2012)

Finalmente, destacamos la publicación de un corpus-metadato de libre distribución de nombre TONÁS³, y la publicación de la primera tesis fin de máster realizada en el ámbito del análisis computacional de la música flamenca (Kroher, 2013). Este trabajo se centra en la descripción automática de las características particulares de distintos cantaores como son timbre, vibrato, etc. con el fin de encontrar combinaciones que permitan la identificación automática del cantaor y del estilo que interpreta.

Finalmente, destacamos que el desarrollo natural del proyecto COFLA ha sugerido el planteamiento de problemas de matemáticas de interés como pueden consultarse en los artículos: (Díaz-Bañez, 2013) (Aichholzer, Díaz-Báñez, Fabila-Monroy, Ochoa, & Nigsch, 2013) (Díaz-Báñez, Korman, Pérez-Lantero, Pilz, Seara, & Silvera, 2013).

³ <http://mtg.upf.edu/download/datasets/tonas>

2.2 Similitud melódica

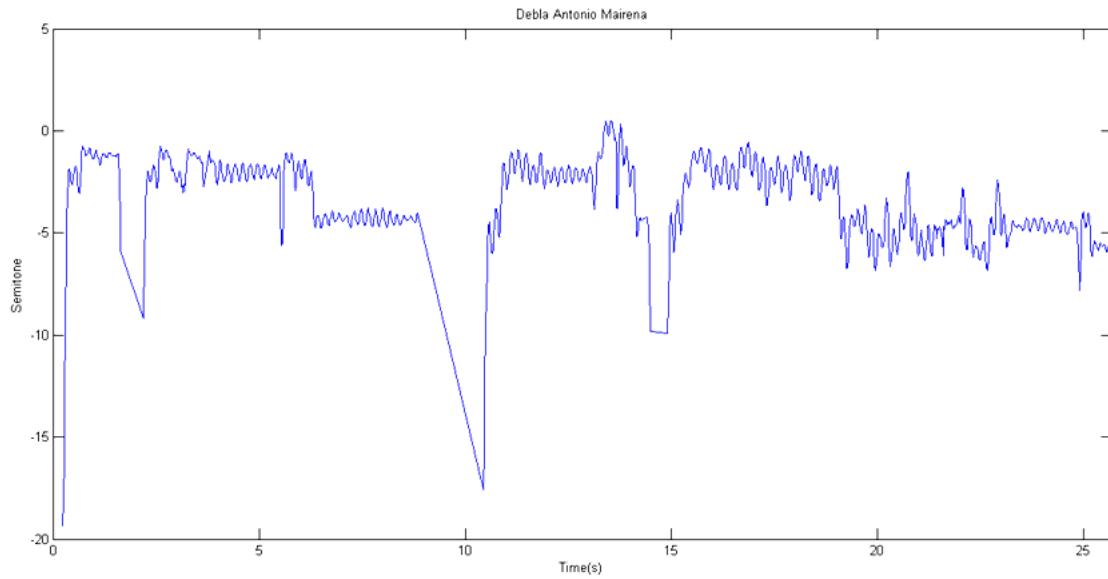
Tanto músicos como psicólogos han estudiado en profundidad las posibles medidas de similitud, debido en parte, a multitud de aplicaciones comerciales tales como sistemas de recomendación, demandas de plagio, organización de bases de datos (audios), clasificación de estilos, etc. Entre las medidas de similitud que se han propuesto destacamos las de carácter geométrico y las distancias de transporte. Puesto que no existen en la actualidad modelos computacionales contrastados que traten la melodía del flamenco (en realidad, la investigación científica del flamenco se inició recientemente), éste es un tema actual de estudio. Los trabajos del grupo Cofla suponen los primeros pasos en este ámbito.

De forma general, las medidas de similitud melódica se construyen a partir del siguiente proceso:

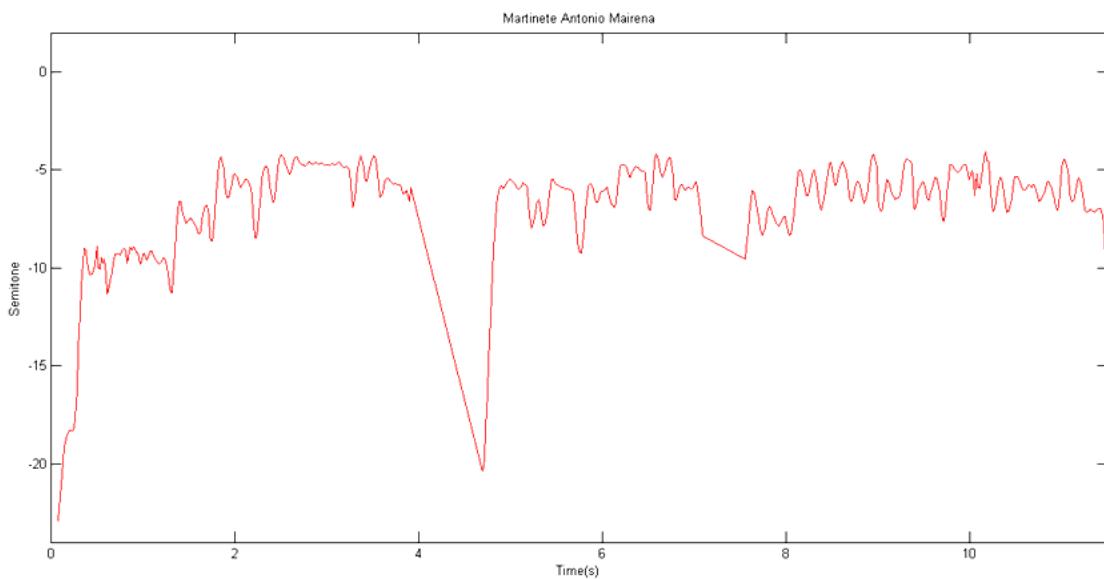
1. **Obtención de la melodía:** Esta puede venir descrita en formato audio, como un fichero de sonido de una grabación, o en formato simbólico, como un fichero midi o un fichero de partitura (Finale, Sibelius, etc.).
2. **Representación de melodías abstractas:** Según los propósitos perseguidos, la melodía se representa de varias formas. Estas representaciones que extraen ciertas características de las melodías reales se llaman representaciones abstractas. Varían en función del propósito final. Por ejemplo, en el análisis de cantes a capela en el flamenco, se ignora la duración de las notas, puesto que son cantes sin pulso regular y con mucho *rubato*.
3. **Transformación de la melodía:** La melodía abstracta es sometida unas transformaciones que permiten el cálculo efectivo de la medida de similitud.
4. **Diseño de la medida de similitud:** Inicialmente, la mayor parte de las medidas se concentraron en un aspecto de las melodías, el cual medían con más o menos precisión. Poco a poco se han ido diseñando medidas que cuantifican varios aspectos de la melodía. El principal problema es cómo ponderar todos esos aspectos de manera coherente

Como se verá a continuación, existen distintas formas de establecer una medida de similitud entre distintas piezas musicales, todas ellas se basan en medir la distancia entre ciertas características musicales de las mismas. Una de las características más empleada a la hora de estudiar la similitud melódica es el *contorno melódico*. Por tanto, el problema de medir la similitud melódica se transforma en cómo medir la distancia entre contornos melódicos. Para ciertos tipos de música ese método puede ser apropiado de manera que puede producir buenos resultados. No es el caso de los cantes flamencos, un tipo de música caracterizada por propiedades especiales por lo que reducir la similitud melódica al cálculo de la distancia entre contornos melódicos deja fuera variables significativas, resultando, de hecho, una medida limitada.

Las funciones a comparar serán la representación de la evolución de la frecuencia fundamental f_0 (semitones) en el tiempo (Figura 2a y 2b).



*Figura 2a: Representación de la frecuencia fundamental.
(Debla de Antonio Mairena)*



*Figura 2b: Representación de la frecuencia fundamental.
(Martinete de Antonio Mairena)*

Existen varios métodos para establecer medidas de similitud, los más importantes son:

2.2.1 Distancia de edición.

La distancia de edición (Mongeau & Sankoff, 1990), también llamada distancia de Levenshtein, es un método usado normalmente en el ámbito de comparación de cadenas de caracteres. Generalizando, se define la distancia de edición (normalmente se usa su nombre en inglés, *edit distance*) entre dos secuencias como el número mínimo de operaciones que necesitamos para pasar de una secuencia a la otra. Estas operaciones son inserción, omisión y sustitución de un carácter de la secuencia. Esta distancia se aplica cuando queremos conocer cuán similares son dos cadenas de caracteres, como puede ser el caso de correctores ortográficos.

Se puede generalizar esta definición de distancia de edición mediante la introducción de las llamadas funciones de costo. Esto consiste en asignar un peso a cada una de las operaciones antes mencionadas. De esa forma, podremos penalizar ciertas operaciones, asignándoles una función de costo con un valor alto, o suavizar otras mediante la asignación de un valor bajo. En general, para la distancia de edición, los costos que se utilizan son:

- Costo = 0, cuando se trata de una coincidencia de carácter.
- Costo = 1, si se produce la sustitución de un carácter por otro.
- Costo = 1, si se produce una omisión o inserción.

Para la implementación del algoritmo de cálculo de la distancia de edición se hace uso de la Programación Dinámica (Lemström, 2000). Dadas las dos secuencias de caracteres $A = \{a_1, a_2, \dots, a_m\}$ y $B = \{b_1, b_2, \dots, b_n\}$, de longitudes m y n respectivamente, se construye una matriz D de tamaño $(m+1, n+1)$ de forma recursiva, donde el elemento d_{ij} representa la distancia entre las secuencias $\{a_1, a_2, \dots, a_i\}$ y $\{b_1, b_2, \dots, b_j\}$. Tendremos que inicializar la primera fila y la primera columna de esa matriz, de la siguiente forma:

$$D(i,0) = i \quad 0 \leq i \leq m$$

$$D(0,j) = j \quad 0 \leq j \leq n$$

La ecuación de recurrencia que utilizamos para ir calculando la matriz es de la forma:

$$d_{ij} = \min \left\{ \begin{array}{l} d_{i-1,j} + 1 \\ d_{i,j-1} + 1 \\ d_{i-1,j-1} + 0, \text{ si } a_i = b_j \end{array} \right\}$$

La distancia entre las secuencias es el elemento d_{mn} de la matriz que hemos generado. Para realizar este algoritmo de programación dinámica, la cantidad de operaciones requeridas es del orden de $m \times n$. En la Tabla 1 tenemos un ejemplo de cálculo de la distancia de edición entre dos cadenas de caracteres.

	<i>P</i>	<i>L</i>	<i>A</i>	<i>N</i>	<i>E</i>	<i>T</i>	<i>A</i>
0	1	2	3	4	5	6	7
<i>P</i>	1	0	1	2	3	4	5
<i>L</i>	2	1	0	1	2	3	4
<i>U</i>	3	2	1	1	2	3	4
<i>T</i>	4	3	2	2	2	3	3
Ó	5	4	3	3	3	3	4
<i>N</i>	6	5	4	4	4	4	D=5

Tabla 1: Cálculo de la distancia de edición usando programación dinámica.

Esta medida de similitud puede usarse cuando se trabaja con secuencia de notas. Por lo que al tratarse de melodías, hay que considerar que cada carácter representa una nota, y cada nota es una combinación de altura y duración. Tendríamos que definir por tanto una distancia de edición que tenga en cuenta esas dos dimensiones.

Como se comentó en el apartado precedente, para el flamenco se ha desarrollado una medida de similitud basada en la edit distance en la que se ponderan las características propias de la música flamenca según su contorno musical (distancia MC) y según un descriptor musical (distancia MD) obtenido a partir de la características fijadas, dicha ponderación se realiza de la forma $d_I = \alpha \cdot d_{MC} + (1 - \alpha) \cdot d_{MD}$, donde α determina el peso de cada característica (Mora, Gómez, Gómez, Escobar-Borrego, & Díaz-Bañez, 2010).

2.2.2 Subsecuencia común de mayor longitud (LCS).

Esta medida de similitud se puede entender como una particularización de la edit distance con la salvedad de que ahora no hay penalización para los huecos de cualquier tamaño entre los símbolos iguales. Por ello se puede entender que a mayor sea la subsecuencia común entre una pareja de melodías más similares son, siendo el caso extremo de dos melodías iguales en las que la subsecuencia común sería la melodía completa.

Se ha manifestado en diferentes trabajos que se puede utilizar el algoritmo LCS como medida de similitud entre melodías, ya que éste tiene en cuenta las variaciones en la velocidad y la imprecisión en el ritmo entre las dos melodías comparadas (Guo & Siegelmann, 2004).

Formalmente el problema de LCS se define como sigue. Dada una secuencia $X = (x_1, x_2, \dots, x_m)$ y una secuencia $Y = (y_1, y_2, \dots, y_n)$, encontrar una secuencia Z , de tal manera que Z sea la secuencia más larga que a la vez es subsecuencia de X e Y . La subsecuencia es definida como una secuencia $Z = (z_1, z_2, \dots, z_k)$ donde existe una sucesión estrictamente creciente de los índices de X tal que para todo $j = 1, \dots, k$, $x_i = z_j$ (Cormen, Leiserson, & Rivest, 1990).

La solución al problema LCS implica la resolución de la siguiente ecuación de recurrencia, donde el coste para las operaciones de edición se almacena en c .

$$c(i,j) = \begin{cases} 0 & \text{si } i = 0 \text{ ó } j = 0 \\ c(i-1, j-1) + 1 & \text{si } i, j > 0 \text{ y } x_i = Y_j \\ \max [c(i-1, j), c(i, j-1)] & \text{si } i, j > 0 \text{ y } x_i \neq Y_j \end{cases}$$

El uso de la LCS como una medida de similitud entre dos secuencias tiene la ventaja de que las dos secuencias que se están comparando pueden ser de diferente longitud y tener huecos colocados de manera aleatoria. En el contexto de la tecnología musical, permite el uso de entradas parciales y con ruido.

2.2.3 Dynamic Time Warping.

Este algoritmo de cálculo de similitud, conocido por sus iniciales DTW (Sakoe & Chiba, 1978), se aplica cuando las dos secuencias que vamos a comparar pueden variar en tiempo o velocidad, es decir, cuando ambas secuencias tienen diferente longitud. Es muy utilizado en aplicaciones de reconocimiento automático del habla, donde el sistema puede recibir una misma palabra enunciada a diferentes velocidades.

En nuestro caso, las secuencias de entrada serán las dos melodías que queremos comparar. Se define una matriz con las distancias entre cada par de notas que forman ambas melodías, y mediante un algoritmo basado en Programación Dinámica, se busca el camino que minimiza esa suma de distancias. Ese camino óptimo es el que define de alguna manera cómo deben ser transformadas las secuencias de entrada, comprimiéndose o expandiéndose, para que presenten el mejor ajuste posible y así se suavice el efecto introducido por el hecho de que las melodías tengan diferente velocidad.

En general, el funcionamiento del algoritmo se basa en la búsqueda de un camino óptimo de coincidencia entre dos secuencias con ciertas restricciones. Las secuencias son transformadas no linealmente en el tiempo, de modo que son comprimidas o expandidas en el tiempo para que tengan el mismo largo, y así poder compararlas punto a punto.

El algoritmo genera una matriz de distancias (usando en este caso la distancia euclídea) entre cada par de puntos que forman las dos secuencias de entrada. Sobre esa matriz, se busca un camino óptimo en el cual la suma de distancias sea la menor posible, cumpliendo los requisitos de continuidad y monotonía (no pueden existir vueltas atrás).

Una vez obtenido el camino óptimo, se calculará el coste global de “deformación o emparejamiento” de las dos melodías i y j , o *Global warp cost*, se define como (Myers, Lawrence, & Aaron, 1980):

$$GC_{ij} = \frac{1}{p} \sum_{k=1}^p w_k$$

donde w_k son todos los elementos que están en el camino óptimo de la matriz de distancias, y p el número de ellos.

- Algunas propiedades son:

- $GC_{ij} = GC_{ji}$
- $GC_{ii} = 0$

El principal problema de esta medida de similitud es el gran coste computacional que supone, ya que al tratarse de programación dinámica dadas dos cadenas de longitud n y m respectivamente el coste computacional del algoritmo es del orden de $n \times m$.

Mejorar el coste computacional del algoritmo DTW será la principal motivación de éste proyecto, por lo que la Sección 3.1, se centrará exclusivamente en el desarrollo y análisis del DTW clásico y de las diferentes extensiones del mismo construidas para mejorar su coste computacional.

Capítulo 3:

EL ALGORITMO SDTW

1. Introducción.
2. Dynamic Time Warping.
3. Métodos de segmentación.
4. La estrategia combinada.

CAPÍTULO 3: EL ALGORITMO SDTW

En el presente capítulo se desarrollará la propuesta de un nuevo algoritmo de cálculo de similitud melódica, el Segmentation & Dynamic Time Warping (SDTW). Con este nuevo algoritmo se pretende acelerar el método DTW clásico (Sección 3.2) cuando se comparan contornos melódicos. Nuestra propuesta se basa en realizar un preprocesado sobre el contorno melódico extraído automáticamente (métodos de segmentación de los contornos melódicos de la Sección 3.3) para después realizar el cálculo de la distancia de similitud con el algoritmo DTW clásico sobre las correspondientes simplificaciones.

Para ello se propondrá una estrategia que combine métodos de segmentación y medidas de similitud melódica. Este método resulta novedoso con respecto a las distintas corrientes anteriores que proponen aproximaciones del método DTW, pues se propone una aproximación no del método DTW sino de los datos de entrada al mismo. Con esto, se direcciona el peso de la estrategia en la elección de un método de segmentación eficaz y eficiente que sea capaz de captar las particularidades propias del contorno melódico usando el menor número de componentes posibles.

Previo al diseño de nuestra estrategia haremos una revisión rápida de los métodos DTW y segmentación de la literatura con objeto de contextualice nuestro algoritmo y permita realizar las comparaciones tanto en eficiencia como en eficacia entre los distintos procedimientos.

3.1 Dynamic Time Warping (DTW)

3.1.1 Introducción

Si tenemos dos series de caracteres de un alfabeto dado

$$A = (a_1, a_2, \dots, a_n) \text{ y } B = (b_1, b_2, \dots, b_n)$$

una manera natural de medir la similitud entre ambas es a través de la distancia Euclídea

$$d(A, B) = \sqrt{\sum_{i=1}^{i=n} (a_i - b_i)^2},$$

o también de la distancia

$$d(A, B) = \sqrt{\sum_{i=1}^{i=n} |a_i - b_i|}.$$

Se podría aplicar alguna de estas distancias, como medida de similitud entre las series pero un primer inconveniente, radica en la imposibilidad de comparar dos series de distinto tamaño. Pero incluso, en el caso de series de igual tamaño, una dificultad que presentan las distancias mencionadas, radica en que son muy sensibles a distorsiones en el eje del tiempo, aun cuando estas sean pequeñas.

Por ejemplo, si dos señales son iguales salvo por un desplazamiento en el tiempo, la distancia euclíadiana las reconocería como distintas por su característica de alineamiento punto a punto. Lo anterior se puede apreciar gráficamente en la parte superior de la Figura 3. Un alineamiento entre los puntos de ambas series, como se sugiere en la parte inferior de esa figura, resulta intuitivamente más conveniente.

En este sentido, fue introducida la denominada DTW (*Dynamic Time Warping*), que algunos autores traducen al español como *Alineamiento dinámico temporal*. Se reconoce su principal antecedente en el trabajo de Hiroaki Sakoe y Seibi Chiba (Sakoe & Chiba, 1978) en el marco de la resolución de problemas vinculados al reconocimiento de voz.

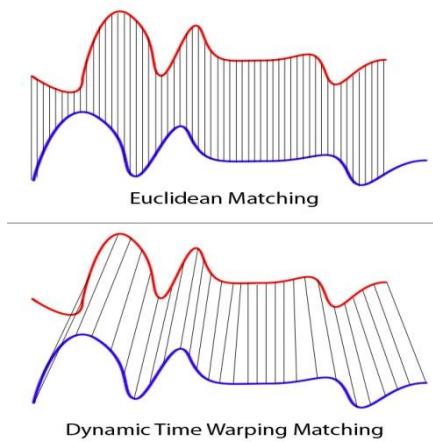


Fig.3 Tipos de alineamiento.

3.1.2 Definición de la medida de similitud DTW

Sean dos series de números reales A y B, de tamaño I y J respectivamente

(2.1)

$$A = a_1, a_2, \dots, a_i, \dots, a_I$$

$$B = b_1, b_2, \dots, b_j, \dots, b_J$$

y consideremos (ver Figura 4) en el plano de ejes \vec{i}, \vec{j} , alineamientos entre los índices de ambas series, a partir de trayectorias F a calcular de la forma:

(2.2)

$$F = c_1, c_2, \dots, c_k, \dots, c_K$$

donde

$$c_k = (i_k, j_k) \text{ son puntos del plano}$$

con

$$\max(I, J) \leq K \leq I + J$$

A estas trayectorias F, les exigiremos que cumplan ciertas restricciones que describiremos en la Subsección 3.1.3

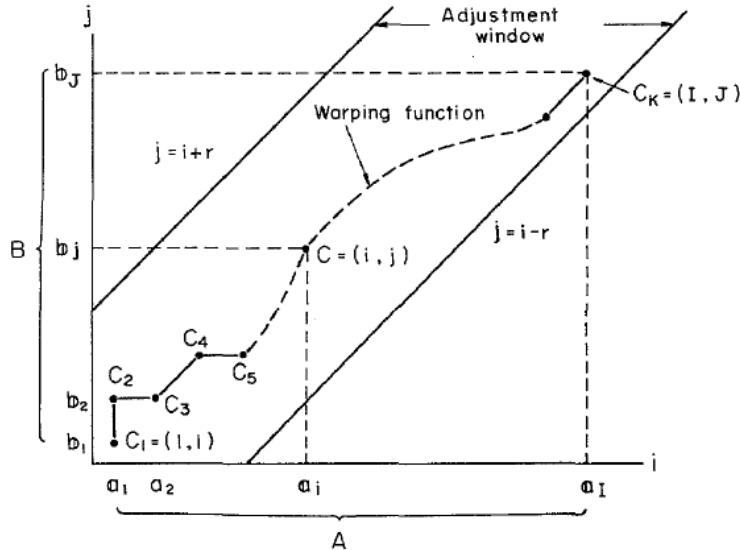


Figura 4. Trayectoria de alineamiento de la trayectoria DTW entre las series A y B, con ajuste de ventana de Sakoe-Chiba de tamaño r. (Sakoe & Chiba, 1978).

Para cada punto c_k , sea $d_k(c_k) = d_k(i_k, j_k) = \|a_{i_k} - b_{j_k}\|$ la distancia entre los dos valores de ambas series que fueron alineados. Dada una trayectoria F, se define la suma ponderada de dichas distancias,

(2.4)

$$S(F) = \sum_{k=1}^{k=K} d(c_k) w_k \text{ con } w_k \geq 0$$

Una vez eliminadas todas las diferencias debidas a “deformaciones” en el tiempo, es razonable pensar que encontramos la trayectoria óptima F_{opt} y que la suma resultante $S(F_{\text{opt}})$ es la “verdadera” distancia entre ambas series.

Lo anterior motiva la siguiente definición de *medida de similitud DTW* entre dos series A y B:

(2.5)

$$DTW(A, B) = \min_F \{S(F)\}$$

3.1.3 Restricciones en las trayectorias de alineamiento

A las trayectorias F a calcular se les impone ciertas condiciones naturales, de manera que las deformaciones temporales en el alineamiento queden sujetas a algunas restricciones:

- Condiciones de monotonía:

$$i_{k-1} \leq i_k \quad y \quad j_{k-1} \leq j_k$$

- Condiciones de continuidad:

$$i_k - i_{k-1} \leq 1 \quad y \quad j_k - j_{k-1} \leq 1$$

Observemos, que a consecuencia de estas dos condiciones, el punto c_{k-1} solamente puede tomar tres valores:

(2.6)

$$c_k = \begin{cases} (i_k, j_k + 1) \\ (i_k + 1, j_k + 1) \\ (i_k + 1, j_k) \end{cases}$$

- Condiciones de borde:

$$i_1 = 1, j_1 = 1, i_K = I, j_K = J$$

- Condiciones sobre la pendiente:

Se trata de que la pendiente de la trayectoria no sea extrema, es decir, ni muy suave ni muy alta, con objeto de impedir que un tramo corto de una de las series se corresponda con uno largo de la otra. La condición es, que si el punto c_k se mueve m veces de forma consecutiva en la dirección de uno de los ejes, entonces deberá moverse n -veces en la dirección de la diagonal, antes de volver a moverse en la misma dirección del eje anterior, ver Figura 5. La manera de medir la intensidad de esta restricción es a través del parámetro $p=n/m$. Cuanto más grande es p , mayor es la restricción. En particular, si $p=0$ no hay restricción y si $p=\infty$ la trayectoria está restringida a moverse sobre la diagonal, es decir que no se permite deformación alguna.

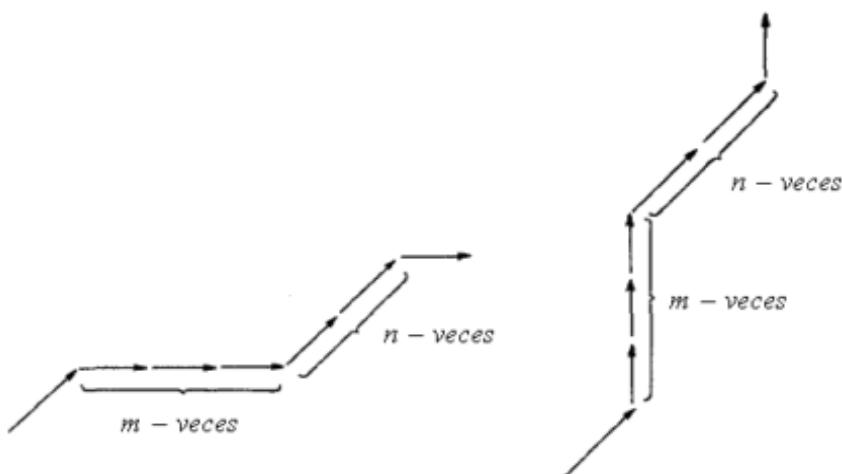


Figura 5. Ejemplo de restricción en la pendiente para $p = \frac{2}{3}$. Tres movimientos del punto c_k en la dirección del eje vertical u horizontal, obliga a efectuar al menos dos movimientos en la dirección de la diagonal.
[(Sakoe & Chiba, 1978)].

3.1.4 Algoritmo de programación dinámica

El problema de alineamiento se resuelve mediante técnicas de programación dinámica, motivo por el que se conoce a ésta técnica como alineamiento temporal dinámico (*Dynamic Time Warping*).

Debido a la gran cantidad de información que contienen los patrones a comparar, se trabajará con un modelo más sencillo. Para la similitud melódica entre dos secuencias, sólo se aplicarán las restricciones vistas de monotonía, continuidad y borde. Además se tomarán todos los coeficientes de ponderación igual a 1, ya que se le da la misma importancia a cualquier intervalo de tiempo dentro del patrón a comparar. Por lo que se obtiene

(2.7)

$$DTW(A, B) = \min_F \sum_{k=1}^{k=K} d(c_k)$$

Para este caso, veamos cómo implementar un algoritmo de programación dinámica. Supongamos que tenemos como en (2.1) dos series A y B y queremos calcular la trayectoria DTW (A,B), definida por la expresión (2.7). El procedimiento de determinar todas las trayectorias F, calcular $\sum_{k=1}^{k=K} d(c_k)$ para cada una de ellas y al final hallar el mínimo es muy ineficiente. En lugar de eso, se utilizará el paradigma algorítmico de programación dinámica.

Se comienza por definir la llamada *matriz de costos D*, de dimensión I x J, cuyas entradas $D(i, j)$ quedan definidas por las sub-series:

(2.8)

$$\begin{aligned} A' &= (a_1, a_2, \dots, a_i) \quad i = 1, \dots, I \\ B' &= (b_1, b_2, \dots, b_j) \quad j = 1, \dots, J \end{aligned}$$

de la siguiente manera:

$$D(i, j) = DTW(A', B')$$

Se observa, que a consecuencia de las restricciones de monotonía y continuidad impuestas a las trayectorias resumidas en (2.6), se puede calcular cada elemento $D(i, j)$ de la matriz a partir de sus elementos adyacentes $D(i - 1, j)$, $D(i, j - 1)$ y $D(i, j)$ y de la distancia $d(i, j)$ entre a_i y b_j , mediante la siguiente fórmula de recurrencia:

(2.9)

$$D(i, j) = d(i, j) + \min\{D(i - 1, j), D(i, j - 1), D(i - 1, j - 1)\}$$

Para calcular la matriz D, se va llenando cada columna de abajo hacia arriba, comenzando por la columna de más a la izquierda y continuando hacia la derecha, ver

Figura 6. Cuando se ha terminado este proceso y por la propia definición (2.8), obtenemos la medida de similaridad buscada:

$$DTW(A, B) = D(I, J)$$

A su vez, a partir de la matriz de costos podemos determinar la trayectoria óptima (best path)

$$F_{opt} = \tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_k, \dots, \tilde{c}_K$$

Comenzando por el punto $\tilde{c}_K = (I, J)$, en un paso genérico k , para el punto $\tilde{c}_k = (i_k, j_k)$, se busca en cuál de los tres adyacentes ubicados a la izquierda y abajo $D(i_k - 1, j_k)$, $D(i_k, j_k - 1)$ y $D(i_k - 1, j_k - 1)$ es mínimo, así se va construyendo la trayectoria hasta llegar a $\tilde{c}_1 = (1, 1)$.

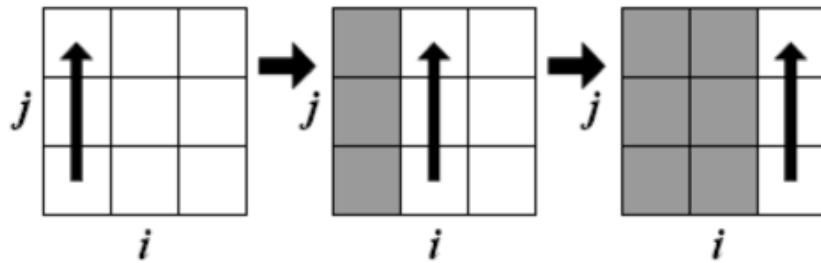


Figura 6. Orden según el cual se calcula la matriz de costos. [(Salvador & Chan, 2007)].

La implementación del algoritmo se puede consultar en el Anexo, en el archivo *Dtw_Sakoe_GC.m*.

3.2.5 Complejidad del algoritmo

De la construcción del algoritmo se puede deducir que su complejidad es $O(I \times J)$ (ya que hay que llenar toda la matriz D). En el caso particular de $I = J$ su complejidad es $O(I^2)$. La complejidad cuadrática es un importante inconveniente al tratar (como hacemos en este proyecto) con series muy largas, del orden de mil componentes.

Por otro lado, en un contexto musical tenemos un número lineal de series ($O(n)$ melodías a comparar) y se necesita calcular la trayectoria DTW entre todas ellas la complejidad pasa a ser de orden $O(I^2 \times n^2)$, ineficiente en aplicaciones reales.

3.2.6 Algoritmos de aproximación del método DTW

Con motivo de intentar mejorar el orden de complejidad del algoritmo DTW clásico, se encuentran en la bibliografía distintas estrategias que tienen como objetivo acelerar los cálculos cuando aparece involucrada la trayectoria DTW. Estas estrategias devuelven una aproximación del cálculo óptimo DTW y se pueden englobar en tres tipos:

- Mediante modificaciones en las restricciones en las trayectorias.
- Mediante modificaciones en la representación de la serie.

También existe un método de simplificación por etapas llamado FastDTW, el cual no pertenece a ningún grupo de los anteriormente citados.

- Restricciones en las trayectorias:

Se basan en la inclusión de una restricción del tipo:

$$|i_k - j_k| \leq r$$

Con esto se impiden excesivas deformaciones temporales creando una “ventana” (ver Figura 7) en la que se encontrará el camino óptimo. Estas ventanas son conocidas como “Bandas de Sakoe-Chiba”. Otras ventanas usadas frecuentemente son las denominadas “Paralelogramo de Itakura”(Itakura, 1975). En estos casos, los algoritmos son más rápidos, pues no se necesita calcular toda la matriz de costos, solamente la zona creada por la ventana, aunque obviamente, la trayectoria y el correspondiente costo DTW encontrados no son los óptimos sin embargo esto deja de ser un inconveniente cuando las trayectorias óptimas se encuentran cerca de la diagonal. Por el contrario, cuando las deformaciones en el tiempo son importantes (como es el caso de comparación de melodías ornamentadas), las DTW encontradas pueden alejarse de las verdaderas.

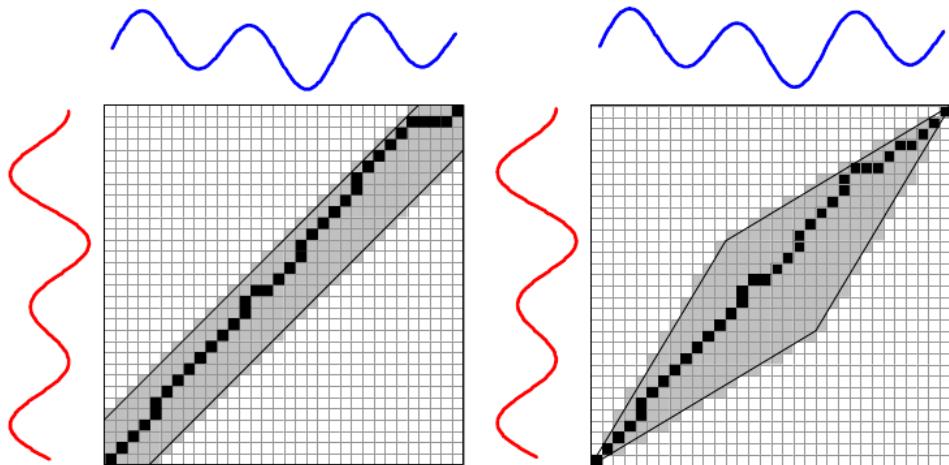


Figura 7. A la izquierda: Banda Sakoe-Chiba. A la derecha: Paralelogramo de Itakura

La implementación del algoritmo presentado en esta sección se encuentra en el archivo `dtw_banda.m` del Anexo.

- Representación de las series en forma reducida:

Otra forma de disminución de la complejidad del algoritmo consiste en calcular el camino óptimo DTW sobre una representación reducida de las series. Una manera elemental de hacerlo es partir de la serie en intervalos iguales, para luego tomar como representación reducida, la serie formada por los promedios de los valores de la serie original en cada intervalo.

En (Chu, Keogh, Hart, & Pazzani, 2002), se propone una representación reducida de este tipo, denominada PPA (Piece Aggregate Aproximation) y un algoritmo que da una aproximación de la media DTW, la llamada PDTW, que resulta de calcular la trayectoria DTW entre las representaciones reducidas, ver Figura 8. En este trabajo se plantea un algoritmo llamado IDDTW (Iterative Deepening Dynamic Time Warping) mediante el cual, dado un nivel preestablecido de tolerancia para falsos disímiles y mediante una fase de entrenamiento, se determina cuál es nivel de reducción óptimo \tilde{l} sobre el cual trabajar.

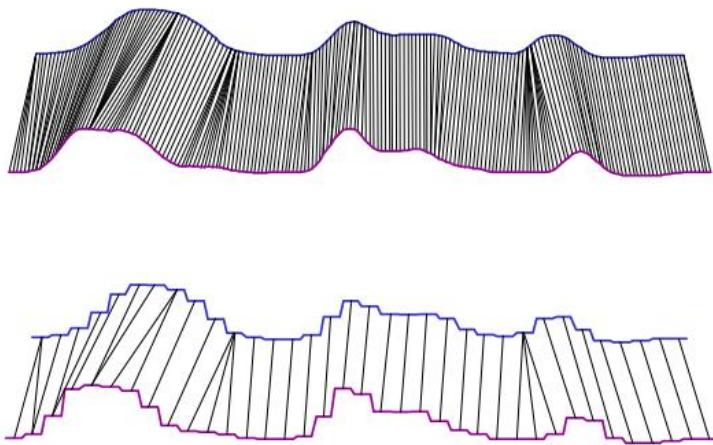


Figura 8. Arriba: Alineación producida por el algoritmo DTW. Abajo: Alineación producida por la PDTW sobre las representaciones reducidas de las series originales.

- El método FastDTW :

El método FastDTW (Salvador & Chan, 2007), comienza por obtener un representaciones reducidas A^l y B^l para un l “pequeño” y calcular la trayectoria óptima correspondiente a la $DTW(A^l, B^l)$. A continuación, se pasa a la siguiente representación A^{l+1} y B^{l+1} y se proyecta la trayectoria encontrada en este nuevo cuadro. Luego, se busca la trayectoria óptima según la trayectoria DTW para este nuevo nivel de reducción, pero restringiendo la búsqueda a las celdas adyacentes de la trayectoria proyecta, ver Figura 9. Así sucesivamente, hasta llegar a la resolución máxima ($2^u = I$), donde tomamos como aproximación de la verdadera $DTW(A, B)$ a FastDTW(A, B). Para aumentar en cada paso la trayectoria proyectada, y con ello la posibilidad de poder encontrar un mejor camino óptimo, en cada paso se aumenta el radio (s) de celdas adyacentes a la última proyección. En la Figura 9, las celdas sombreadas suavemente corresponden a un radio $s=1$.

Al tratarse de una aproximación del DTW, el método FastDTW proporciona alineaciones casi óptimas con una complejidad con un orden $O(I)$ frente a la complejidad de orden $O(I \times J)$ del método DTW clásico.

Los autores comparan el rendimiento del método FastDTW (A, B), con dos de las técnicas que mencionamos antes, a) bandas de Sakoe- Chiba y b) representación reducida de las series, sobre un amplio espectro de bases de datos. Reportan mejoras en la precisión, que van desde 51 veces mejor que a) y 143 que b) para $s = 0$, hasta 3 veces mejor que a) y 15 que b) para $s = 30$. En cuanto a la velocidad, comparan la performance de la FastDTW respecto de la DTW. En series de tamaño 100, no hay mejoras significativas, para tamaño 1000, FastDTW es 46 veces más rápida que DTW

con $s = 0$ y del mismo orden con $s = 100$. Cuando la longitud de las series es 10000, mejora 151 veces la velocidad con $s = 0$ y 7 veces con $s = 100$. Con longitud 100000, 117 veces con $s = 0$ y 38 veces con $s = 100$. Concluyen que para series de longitud menor que 300, no vale la pena aplicar la FastDTW, ya que la mejora en velocidad no es significativa y siempre resulta más precisa la DTW.

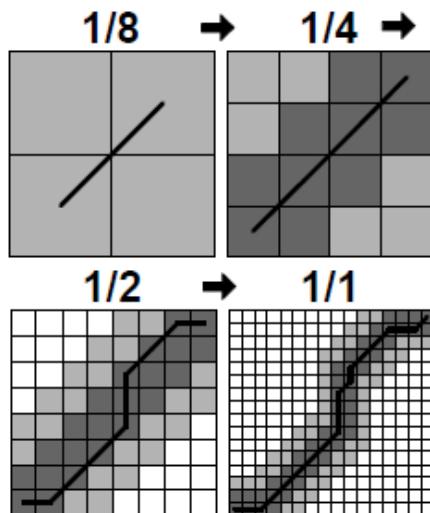


Figura 9. Cuatro etapas del método FastDTW.

3.3 Métodos de Segmentación

Como veremos más adelante, la idea clave que usaremos para aumentar la eficiencia del algoritmo *DTW*, se basa en que éste no evalúe la curva melódica original, ya que contiene una gran información, parte de la cual carece de relevancia en el estudio realizado debido al carácter ornamental del flamenco. Para ello se “simplificará” la curva melódica del cante, obteniendo con ello una más simple que, aunque pierde matices propios de cada interpretación, mantiene el esqueleto melódico o parte de él, lo cual nos va a permitir realizar labores de discriminación entre estilos.

Para lograr dicha “simplificación” se realizará una segmentación de la curva, utilizando dos metodologías distintas; por un lado un *Asistente computacional de transcripción de cante flamenco (SmsTools2)* (Gómez & Bonada, 2008) basado en métodos de análisis del pitch y un algoritmo de aproximación que denominamos Fitting basado en el trabajo de (Díaz-Báñez & Mesa, 2001) sobre una ubicación óptima de trayectorias ortogonales.

3.3.1 Segmentación realizada por Smstools 2

Este sistema de transcripción melódica extrae de descriptores melódicos a diferentes niveles. Por un lado, usa las características de bajo nivel que poseen mayor relación con la melodía, como son: la energía (asociada al volumen), y la frecuencia fundamental (f_0) relacionada con la percepción del *pitch* o tono de la melodía. Y en un nivel estructural más alto sitúa los flujos de audio, que son segmentados en notas, de las cuales su duración y tono dan una representación simbólica de la misma.

La Figura 10, muestra los distintos pasos del algoritmo de transcripción automática realizados por el *SMSTools 2*.

Transcripción iterativa de la nota

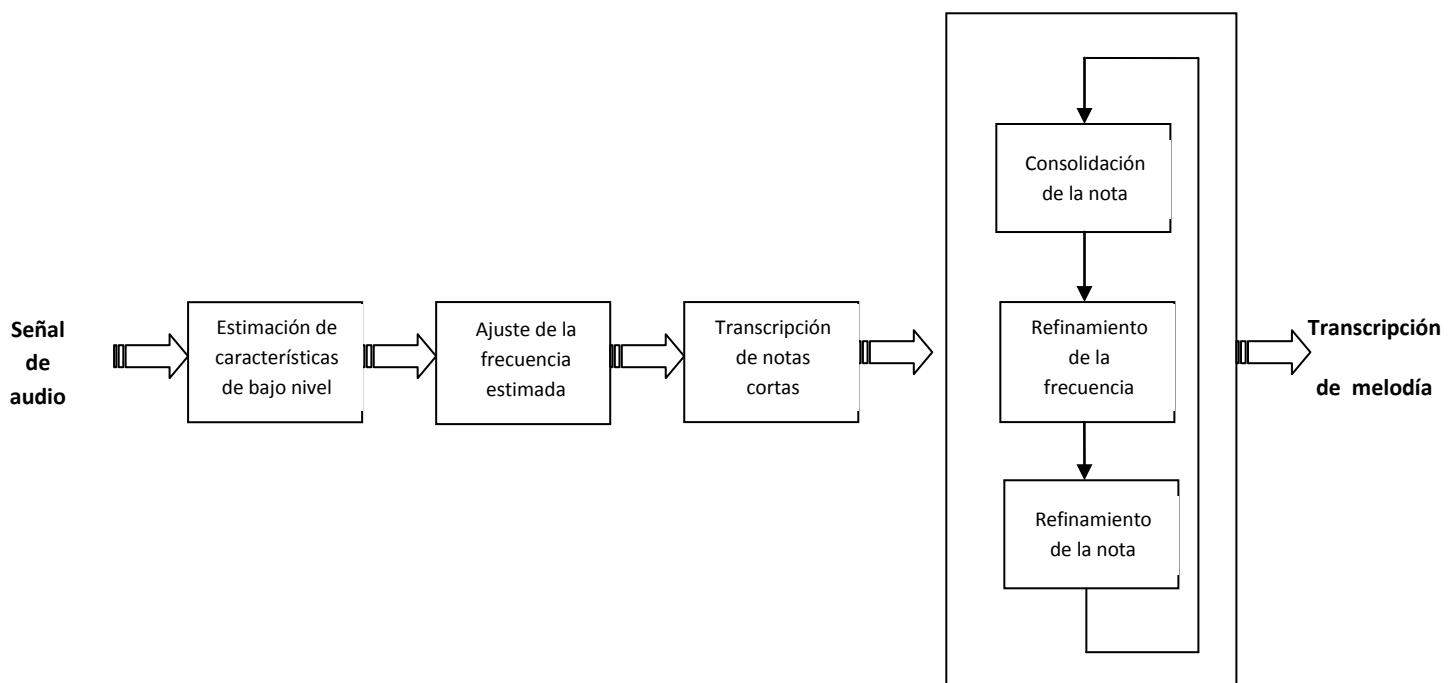


Figura 10. Diagrama para la transcripción automática Smstools2.

- Estimación de características de bajo nivel:

En primer lugar, la señal de audio es pixelada en segmentos de 50 ms aproximadamente, usando un ratio de segmentación de 172 segmentos/segundo. Para cada segmento se obtiene su espectro, esto es, energía y frecuencia fundamental estimada f_0 .

Para la estimación de la f_0 , el *SMSTools 2* realiza una autocorrelación en el dominio de la frecuencia y el tiempo, implementando el método *SAC (Spectrum AutoCorrelation)* basado en el cálculo de la amplitud en el dominio de la frecuencia mediante

correlaciones. Para ello el algoritmo realiza tres pasos fundamentales: remuestreo, estimación de candidatos de la f_0 segmento a segmento, y por último un post-procesado.

- Ajuste de la frecuencia estimada:

Como el programa analiza características de voces cantadas, la frecuencia de referencia usada por el cantaor para afinar la pieza es desconocida. Por ello hay que localizar los tonos principales en los que el cantante se *apoya* en la interpretación de la pieza.

Se realiza una estimación inicial de ese tono suponiendo que este pertenece a la escala temperada, también se supondrá que la frecuencia o tono de referencia es constante para el extracto analizado. A continuación se establece un histograma que representa la asignación de los valores de f_0 de todos los segmentos en intervalos de un semitono con una resolución de una centésima de tono. El valor añadido al histograma es un peso que representa la relevancia de cada segmento. La solución adoptada por el *SMSTools 2* es dar más peso a los segmentos donde la f_0 es estable y los valores de las desviaciones de f_0 son bajos.

Con el fin de suavizar el histograma, en lugar de añadir un valor a un solo segmento, se utiliza una ventana en forma de campana que se extiende por 20 centésimas de tono. El máximo de este histograma (c_{ref}) determina la desviación de la frecuencia de referencia en centésimas de tono referenciadas a 440 Hz, por lo tanto la frecuencia de referencia estimada en Hz va a ser $f_{ref} = 440 \times 2^{\frac{c_{ref}}{1200}}$.

- Transcripción de notas cortas:

En el siguiente paso el proceso de transcripción automática divide la señal de audio en notas cortas, mediante la búsqueda de la segmentación que maximiza una cierta función de probabilidad. La segmentación estimada se corresponde con el camino óptimo de entre todos los caminos posibles de una matriz M de dimensión 2-D (Figura 11). La matriz M tiene las posibles notas en centésimas de tono como filas ($[c_0, c_n]$ con un semitono de resolución) y los tiempos de los segmentos analizados como columnas. Se ha de tener en cuenta que las posibles notas estimadas deberían cubrir la *tesitura* del cantaor ($[c_{min}, c_{max}]$) e incluir el $-\infty$ para los tramos sin voz. La duración máxima n_{max} debe ser lo suficientemente larga para cubrir algunos periodos de vibrato con una baja modulación de la frecuencia, pero suficientemente cortos para tener una buena resolución temporal.

Los posibles caminos considerados por el algoritmo siempre van desde el primer segmento hasta el último, y avanza en el tiempo para que las notas no se solapen. Un camino P es definido por una secuencia de notas m , de la forma $P = \{N_0, N_1, \dots, N_{m-1}\}$, donde cada nota N_i empieza en un cierto segmento k_i , tiene una desviación respecto de la frecuencia de referencia c_{ref} de c_i centésimas de tono y una duración de n_i segmentos. El camino óptimo es definido como el camino que maximiza cierta función de probabilidad de entre todos los caminos posibles. La probabilidad L_P de un determinado camino es determinada como el producto de las probabilidades de cada nota L_{N_i} por la probabilidad de paso entre dos notas consecutivas L_{N_{i-1}, N_i} , que se define como:

$$L_P = L_{N_0} \times \prod_{i=1}^{m-1} (L_{N_i} \times L_{N_{i-1}, N_i})$$

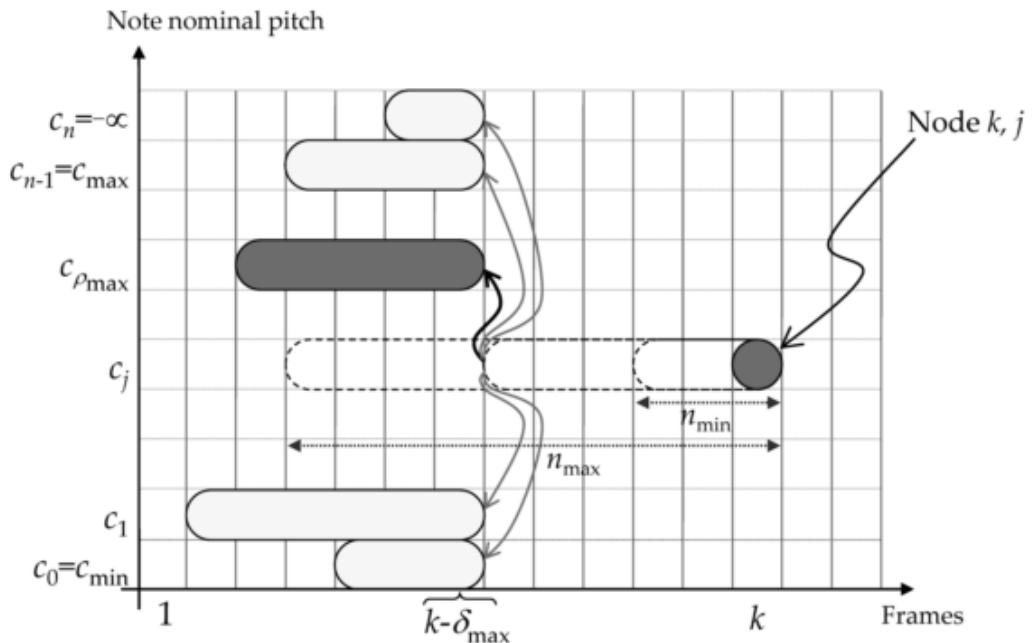


Figura 11: Matriz M usada en el proceso de segmentación de notas cortas. Se ilustra cómo el mejor camino es determinado para un nodo con segmento k y nota j . Son consideradas todas las posibles duraciones entre n_{min} y n_{max} y todos los posibles saltos con las notas previas. En este ejemplo δ_{max} se encuentra que es la duración de nota más probable y p_{max} el índice de la nota previa. (Figura extraída de (Gómez & Bonada, 2008))

Por otro lado, se debe tener en cuenta, que en este algoritmo la probabilidad L_{N_i} de una nota N_i es determinada como el producto de distintas funciones de probabilidad, basadas en diferentes criterios, como: duración (L_{dur}), frecuencia fundamental

(L_{pitch}), existencia o no de voz cantada ($L_{voicing}$), y características de baja nivel relacionadas con la estabilidad ($L_{stability}$).

De todo lo anterior, se desprende que para la nota N_i , el cálculo de la probabilidad es:

$$L_{N_i} = L_{dur} \times L_{pitch} \times L_{voicing} \times L_{stability}$$

El camino óptimo P_{max} se obtiene mediante la búsqueda del nodo de la última columna con una probabilidad máxima acumulada, y después se sigue su correspondiente salto y secuencia de notas.

- Consolidación iterativa de la nota y refinamiento de la frecuencia:

Este tercer paso constará de un proceso iterativo conformado por tres etapas: consolidación de notas consecutivas con el mismo tono y de transición suave en una única nota, refinamiento de la frecuencia de referencia estimada de acuerdo a las notas obtenidas, y reestimación del tono de la nota nominal. El proceso se repetirá hasta que no haya más consolidaciones.

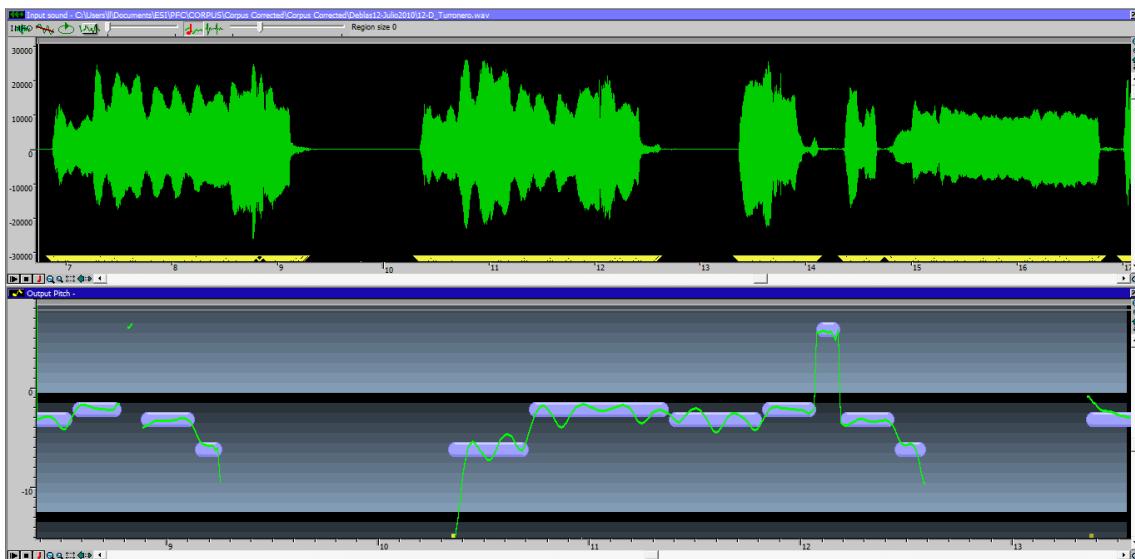


Figura 12: Ejemplo de visualización del programa SMSTools 2 para transcripción automática. Representación de “audio waveform” (parte superior) y la f_0 estimada junto con la segmentación obtenida por el programa (parte inferior).

Con relación al tiempo computacional del algoritmo de transcripción automática, en el artículo no aparece referencia alguna a este asunto. Desde el punto de vista teórico podemos decir que es al menos de tiempo cuadrático (contiene un algoritmo de programación dinámica) y, desde el punto de vista experimental, podemos decir que requiere tiempo de ejecución del orden de la duración de la pieza a segmentar.

3.3.2 Algoritmo Fitting

A continuación se describe el algoritmo que denominaremos Fitting, que fue propuesto para obtener aproximaciones de curvas usando funciones escalonadas (Díaz-Báñez & Mesa, 2001). El problema de aproximación de curvas se define como:

Dada una curva S definida por un conjunto de n puntos en el plano, y una tolerancia al error $\alpha \geq 0$, construir una curva ortogonal R de tal forma que el error de R respecto a S no sea mayor que α , y el número de segmentos de R sea el mínimo posible. El error de R respecto a S se define como $e(R,S)=\min \max \{d_v(r,s), r \text{ en } R, s \text{ en } S\}$, donde $d_v(r,s)$ denota la distancia vertical entre los puntos r y s , esto es, la diferencia de sus ordenadas en módulo.

El algoritmo que se propone es como sigue: Dado un conjunto de puntos $P = \{p_1, p_2, \dots, p_n\}$ en el plano y una tolerancia al error α , se trazan segmentos verticales V_i de longitud 2α centrados en cada punto p_i (ver Figura 13). La restricción fijada, de que cada punto esté dentro del intervalo α de la función escalón es equivalente a decir que la función escalón intercepta cada uno de estos segmentos de longitud 2α . Barriendo de izquierda a derecha, el algoritmo trata de interceptar tantos segmentos consecutivos como sea posible, antes de empezar un nuevo escalón y repetir el procedimiento.

El segmento vertical V_i define un intervalo $y = [y_i^+, y_i^-]$ donde y_i^+ e y_i^- denotan las coordenadas del punto final superior y inferior de y respectivamente. Recorriendo de izquierda a derecha se mantiene la intersección Δ de los intervalos y y de los segmentos verticales hasta que se llega a un segmento V_i en el cual el intervalo y no es interceptado por Δ , en este caso se termina la construcción del segmento, y se empieza un nuevo escalón.

Suponiendo que los puntos de partida están dados ordenados lexicográficamente, el tiempo computacional de este algoritmo es del orden $O(n)$, y construye la función escalón óptima con una tolerancia al error α .

Aplicando este algoritmo sobre la curva S , definida por el contorno melódico f_0 , se obtiene una representación discreta óptima de la curva original para un umbral de rango fijo. Teniendo en cuenta que las máximas variaciones de tono en la música

flamenca están contenidas en una octava musical, la elección de valores de umbral comprendidos entre uno y medio semitono parece adecuada para capturar el esqueleto melódico de la pieza. De todas maneras, este parámetro puede ser elegido por el usuario en función del problema que afronte.

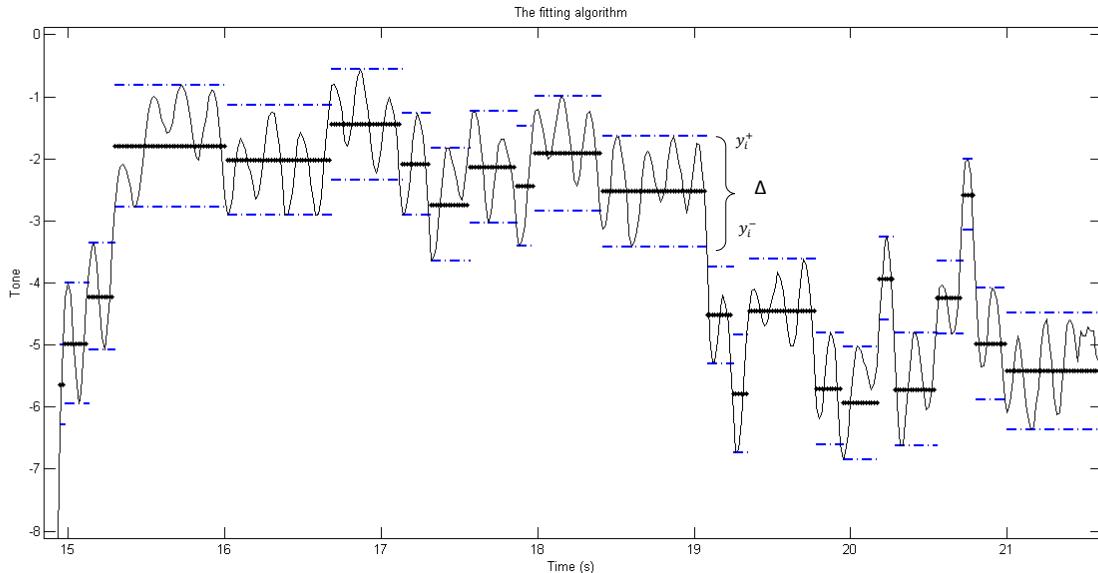


Figura 13: Segmentación de f_0 realizada por el algoritmo fitting.

El desarrollo del algoritmo *Fitting* se encuentra en el archivo *Fitting.m* del Anexo, en él también se desarrolla la subrutina *createventa.m* que realiza la determinación del segmento correspondiente al contorno melódico que captado por el segmento vertical V_i .

3.4 La estrategia combinada

La complejidad cuadrática inherente al método DTW, tanto en tiempo como en espacio, crea la necesidad de establecer estrategias o métodos que aceleren el comportamiento del algoritmo DTW original.

Como ya hemos visto, se han realizado muchos esfuerzos en acelerar la búsqueda de similitud bajo el algoritmo DTW. Dos de las propuestas más usadas son las que se basan en establecer restricciones en las trayectorias de búsqueda, como son las bandas de Sakoe-Chiba o el paralelogramo de Itakura. Recientemente se ha propuesto un algoritmo de orden lineal, FastDTW, pero que posee una limitación en la exactitud de las medidas de similaridad.

A la hora de enfocar una estrategia para ganar eficiencia en el método DTW es posible considerar dos alternativas para el diseño una medida de similitud melódica:

1. Aplicar alguna estrategia de aproximación del DTW, por ejemplo, FastDTW, bandas de Sakoe-Chiba, o alguna otra, y obtener una medida de similitud aproximada asumiendo los errores que ello conlleva.
2. Aplicar el algoritmo clásico del DTW a una aproximación de los datos de entrada calculados mediante algún método de segmentación.

En este trabajo se adoptará la segunda estrategia. Por lo tanto, nuestro objetivo se centra en discretizar la envolvente de la frecuencia fundamental extraída con Smstools2, Mami o métodos similares en pequeños segmentos o secuencias, de modo que el rendimiento de la similitud se mantenga constante.

Como ya se expuso anteriormente, la complejidad temporal de los algoritmos de segmentación utilizados varía desde el orden de la duración del fichero de audio, en el caso de usar la segmentación obtenida por el Smstools2, hasta tiempo óptimo lineal (esto es, del orden del tamaño de la entrada) ofrecido por el método basado en el algoritmo *Fitting*.

A modo de resumen la estrategia combinada se llevará a cabo en los siguientes pasos tal como se expresa en la Figura 14:

1. Obtención de la frecuencia fundamental (f_0) de las piezas a analizar (*.wav), mediante el programa Smstools2.
2. Pre-tratamiento de la frecuencia fundamental para la eliminación de silencios y posibles datos espurios.
3. Normalización de la frecuencia fundamental a una frecuencia de referencia.
4. Segmentación usando los dos métodos elegidos.
5. Aplicación del algoritmo DTW y obtención de la matriz de distancias.

○ PASO 1: Obtención de la frecuencia fundamental:

Para la obtención de la frecuencia fundamental (f_0) se hará uso del programa SmsTools 2, con él se obtendrá a partir de la grabación de audio en formato *.wav su contorno melódico en un archivo tipo *.txt . El archivo estará formado por dos columnas, en la primera aparecerá el tiempo y en la segunda el valor la frecuencia en hercios.

En primer lugar, cargamos el archivo de audio en el programa, para que el archivo de audio sea compatible con el programa éste tiene que ser: formato .wav, en mono, con frecuencia de muestreo de 44.1 kHz y cuantificación lineal de 16 bits.

Seguidamente se procederá al análisis audio para así, obtener la frecuencia fundamental, como se ha comentado en secciones anteriores, el algoritmo del SMSTools2 genera la frecuencia fundamental mediante la estimación de las características de bajo nivel y un ajuste del contorno melódico. El análisis del audio tiene una duración que depende del tamaño o duración del archivo de audio a analizar.

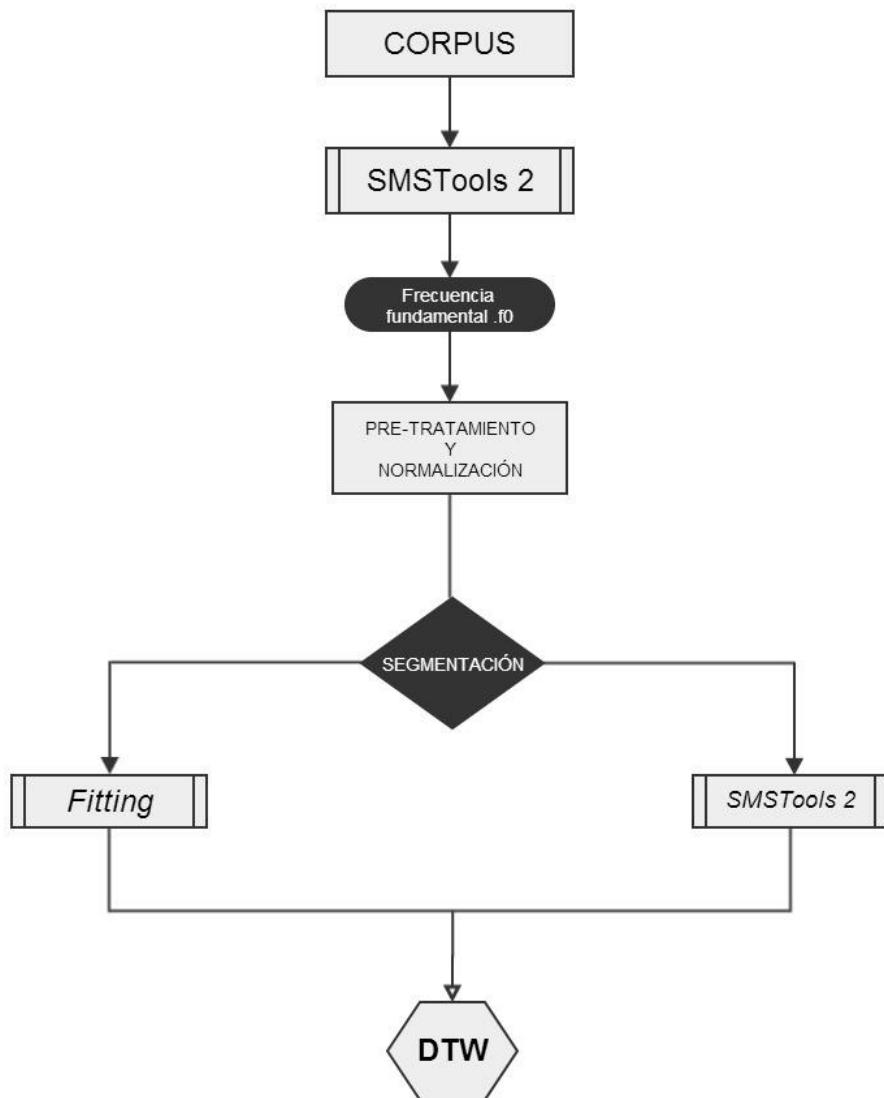


Figura 14: Esquema de la estrategia combinada SDTW.

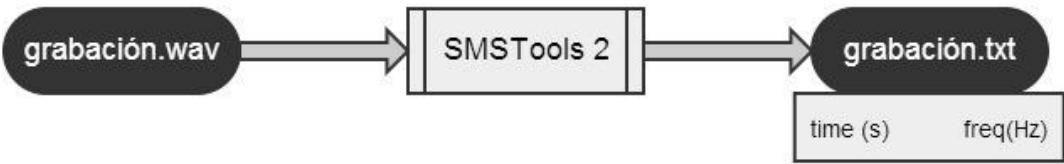


Figura 15: Metodología para la extracción del pitch como fichero de texto.

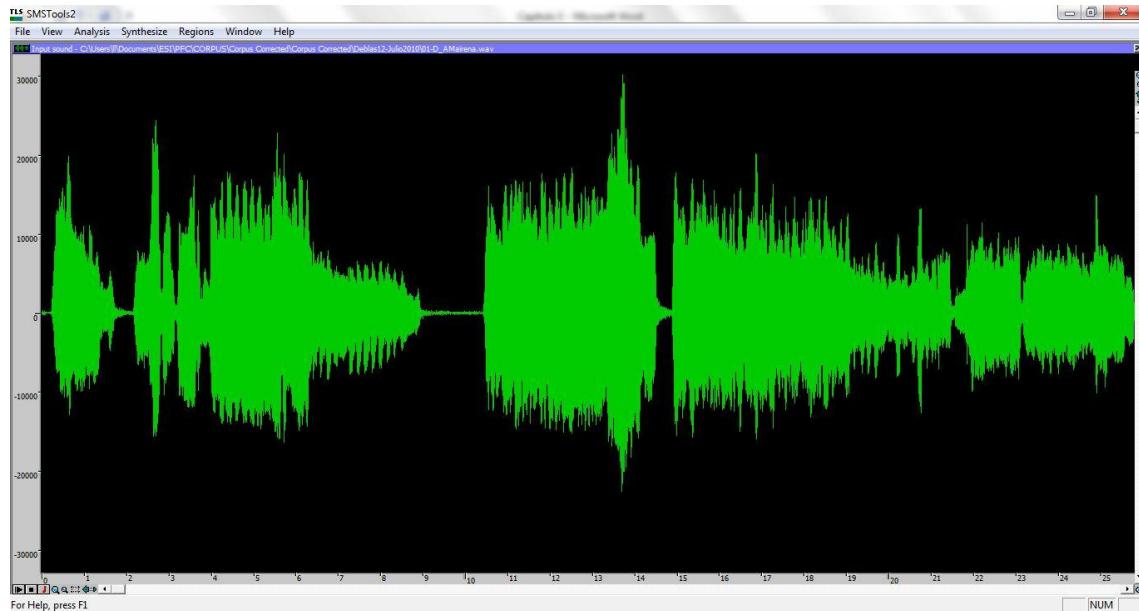


Figura 16: Representación mono del archivo *.wav, del audio cargado en el SMSTools2.

Una vez analizada la pieza por el programa se puede ver en el mismo una representación del contorno melódico mediante la opción *OutputAttpitch*, para desde esa sub-rutina del programa poder acceder al almacenamiento del archivo de texto que contiene la información de la representación temporal del contorno melódico, como se ha explicado con anterioridad, esto es un archivo de tipo *.txt formado por dos columnas, en la primera aparecerá el tiempo y en la segunda el valor la frecuencia en hercios correspondiente a cada instante. La generación del archivo de texto se realiza mediante el comando *Load a pitch envelop*.

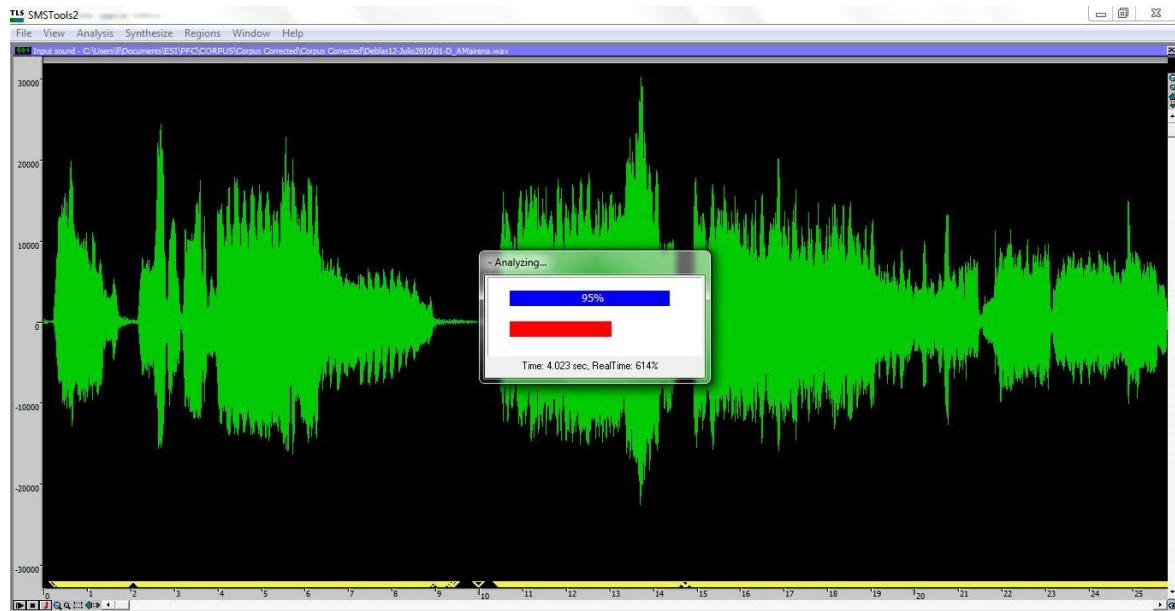


Figura 17: Análisis del audio.

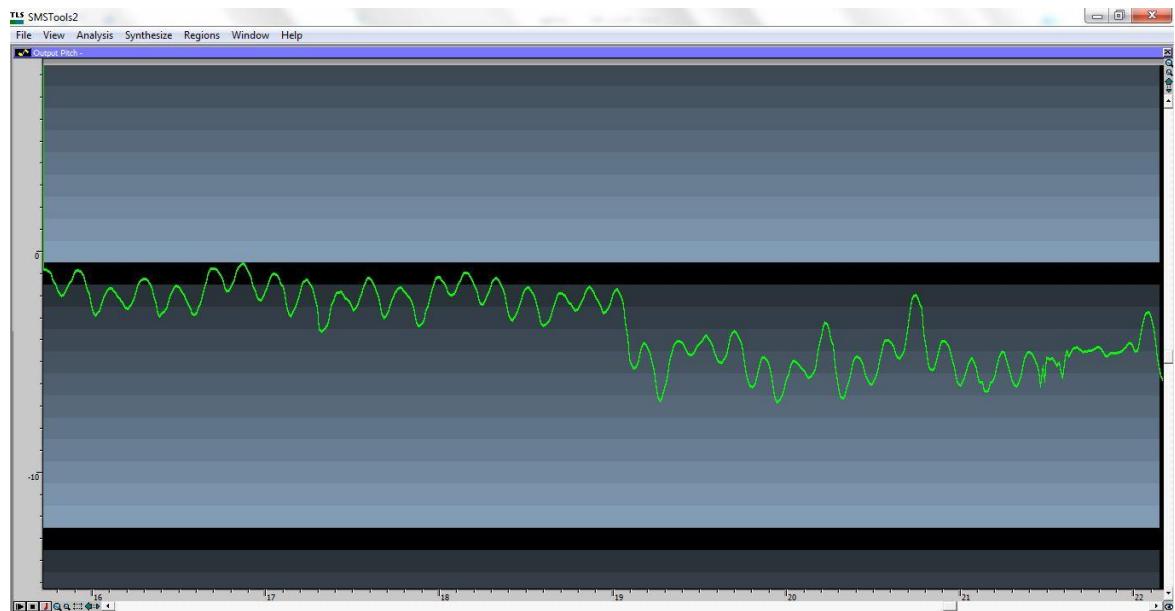


Figura 18: Representación de contorno melódico.

Como ejemplo se muestra parte del archivo de texto que contiene la información correspondiente al contorno melódico de una debla de Antonio Mairena (Figura 18):

Time (s)	Freq (Hz)		
0	0	0,267	175,478
0,017	0	0,283	204,883
0,033	0	0,3	257,341
0,05	0	0,317	312,16
0,067	0	0,333	342,812
0,083	0	0,35	365,606
0,1	0	0,367	384,971
0,117	0	0,383	-393,762
0,133	0	0,4	-395,187
0,15	0	0,417	-389,474
0,167	0	0,433	382,662
0,183	0	0,45	378,999
0,2	0	0,467	377,384
0,217	0	0,483	379,875
0,233	143,918	0,5	386,188
0,25	154,329	0,517	393,624

Tabla 2: Fragmento del archivo de texto que contiene el contorno melódico de una dobla interpretada por Antonio Mairena.

- PASO 2: Pre-tratamiento de la frecuencia fundamental:

Como se puede observar en el fragmento de archivo de texto mostrado en la Tabla 2, éste posee silencios y datos espurios que no aportan información y distorsionan la forma del contorno melódico que se va a analizar, por lo que se realizará un pre-tratamiento de los datos para así tener un perfil melódico correcto. El origen de estos datos que hay que tratar es variado:

- **Silencios:** Existen dos tipos de silencios, los que se encuentra al principio o al final de la pieza y el motivo de que aparezca es que la grabación se inicia antes de que el cantaor empiece la ejecución y termina una vez el cantaor a finalizado, y el otro tipo son los silencios que se encuentra en el transcurso del cante, propios de las respiraciones del cantaor.

El eliminar estos “ceros” del contorno no influyen a la hora de la comparación, ya que no aportan información a la hora de la definición de la estructura del cante y además, el algoritmo *DTW* no tiene en cuenta las repeticiones. Esta rutina está implementada en el archivo *f0clanner.m* del Anexo.

- **Datos espurios:** Se entiende como datos espurios los datos negativos o excesivamente altos. El origen de ellos no es claro y pueden partir de distintas fuentes: jaleos del público, fallos de la grabación, palmas, etc. Por tanto, su eliminación es fundamental para tener seguridad de que se realiza una comparación entre los contornos melódicos correctos y sin errores.

- PASO 3: Cuantización y normalización:

Al estar los datos de la frecuencia fundamental en Hz, debemos normalizarlos para poder comparar todos los cantes en base a un frecuencia de referencia. La frecuencia de afinación tomada como referencia es 440 Hz que en el código de notas MIDI representa la nota LA 4^a. Para normalizar la frecuencia fundamental, se realizará una conversión a semitonos, tomando como se ha dicho como frecuencia de referencia 440 Hz. Dicha conversión se basa en la siguiente fórmula:

$$\text{semitono} = 12 \times \log_2 \left(\frac{f}{f_{ref}} \right)$$

En la Figura 19 se muestra arriba la evolución temporal de la frecuencia fundamental en Hz y abajo su representación semitonos, una vez han sido tratados los datos. Dicha conversión se realiza con el archivo *f0asemitones.m* que se encuentra en el Anexo de códigos Matlab.

- PASO 4: Segmentación:

Como se ha comentado en la introducción a la metodología a seguir, la base fundamental de la nuestra estrategia es simplificar los datos de entrada al algoritmo *DTW* mediante una segmentación del contorno melódico. Dicha segmentación se realizará por dos algoritmos distintos: el algoritmo del SMSTools2 y el algoritmo *Fitting*.

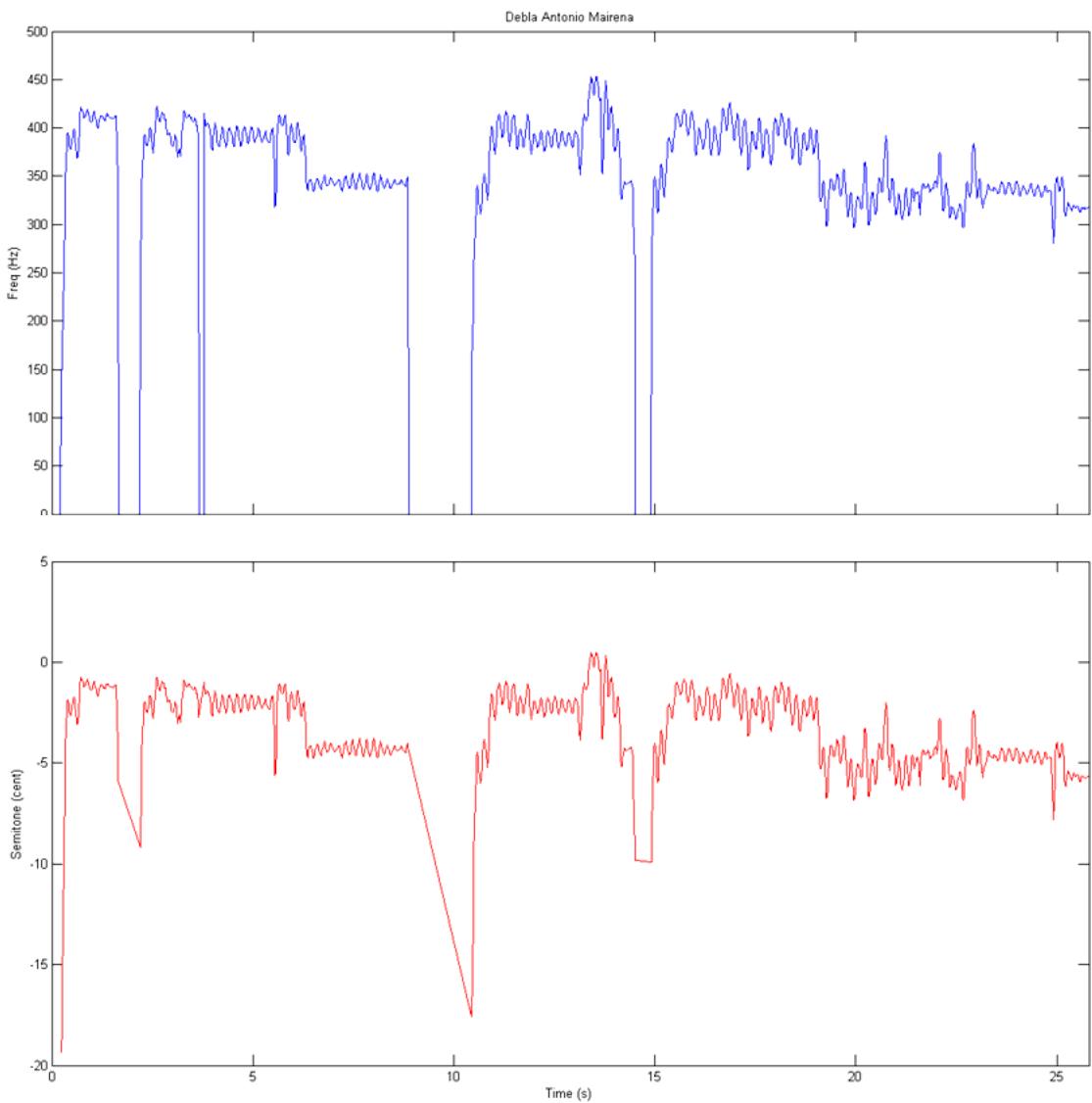


Figura 19: Representación del contorno melódico de una Debla de Antonio Mairena.

■ SEGMENTACIÓN MEDIANTE SMSTools2:

La forma de actuar para obtener la segmentación mediante SMSTools2 es similar a la utilizada para obtener los datos sobre el contorno melódico, pero modificando algunos comandos.

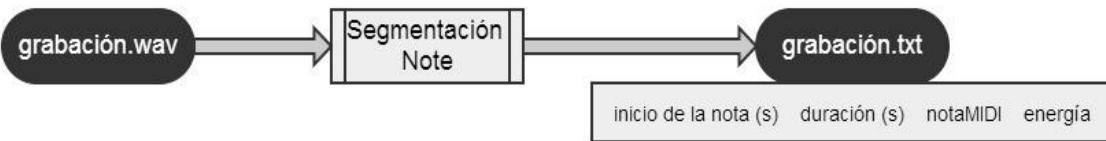


Figura 20: Segmentación via SMSTools2.

Una vez ha sido analizado el archivo de audio como se explicó en Paso 1, y se tiene la representación del contorno melódico utilizando la opción *OutputAttpitch*, se procede a la segmentación del contorno. Para ello, sobre la representación del contorno melódico ofrecida por el SMSTools2 (similar a Figura 18) en la ventana *OutputAttpitch* se pulsa la tecla ‘n’, obteniéndose con ello una representación gráfica de la segmentación realizada que podrá ser corregida manualmente si se desea (Figura 21).

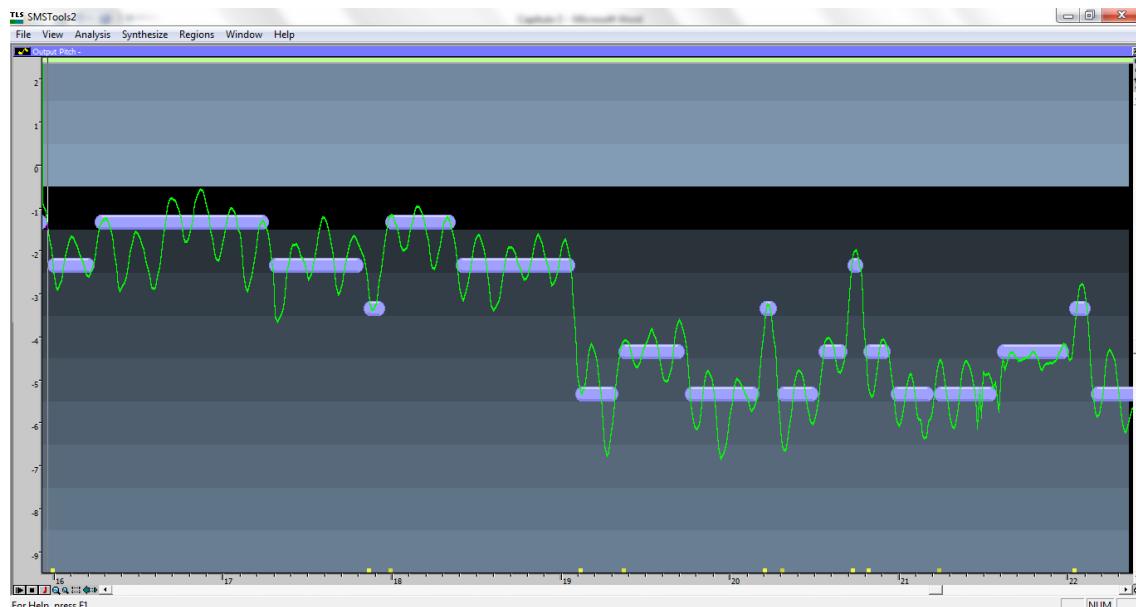


Figura 21: Representación gráfica de la segmentación realizada por SMSTools 2.

Si el usuario estima en algún momento que la segmentación realizada por el programa no es la adecuada o simplemente desea modificar alguna nota obtenida por el programa, éste da la posibilidad de realizar modificaciones sobre la transcripción automática. Para ello basta tener en cuenta los siguientes comandos, todos ellos ejecutados sobre la ventana *OutputAttpitch*:

- ‘c’ Limpiar nota: Elimina la información correspondiente a la nota.

- ‘t’ Cambio de afinación (respecto a la de referencia 440 Hz): Transpone las notas generadas a otras frecuencias de afinación en intervalos de ± 50 centésimas de tono.
- ‘e’ Elimina la limitación de la escala temperada: Permite que las notas sean estimadas pero no limitadas por la escala temperada. Esta opción es recomendable para la transcripción de cantes a capella o no afinados.
- ‘1’, ‘2’, ..., ‘9’ Resolución del contorno: Multiplica por el factor seleccionado la resolución de un semitono asignada por defecto.
- **Control + Botón izquierdo:** Edita la nota segmentada alagándola.
- **Control + d + Botón izquierdo:** Edita la nota segmentada poniéndola a cero.

Una vez el usuario estima que la segmentación realizada es la adecuada, se procede al almacenamiento de la información correspondiente a la segmentación en un archivo del tipo *.txt, en el que se muestra en la información de la siguiente forma:

Inicio (s)	Duración (s)	Nota MIDI	Energía
0,2166670	0,4333330	67	0,018007
0,6500000	1,0166670	68	0,010794
2,1833330	0,3833330	67	0,00698
2,5666670	0,3333330	68	0,03265
2,9000000	0,3333330	67	0,006997
3,2333330	0,4333330	68	0,024974
3,7833330	1,7666670	67	0,031102
5,5833330	0,3166670	65	0,033384
5,9000000	0,1666670	64	0,032187

Tabla 3: Fragmento del archivo de texto que contiene la información correspondiente a la segmentación de una debla interpretada por Antonio Mairena .

Para trabajar con toda la información en una misma “unidad” se tendrá que convertir la información obtenida en la segmentación del SMSTools2 a semitonos, para ello se hace uso de la relación existente entre los números asignados a las notas MIDI y la frecuencia que representan.

$$\text{semitono} = \text{notaMIDI} - 69$$

La conversión a semitonos se ha realizado con el archivo *noteasemitono.m*, que se encuentra en el Anexo.

A continuación se muestra una representación de la segmentación realizada por el SMSTools2 sobre una debla de Antonio Mairena (Figura 22).

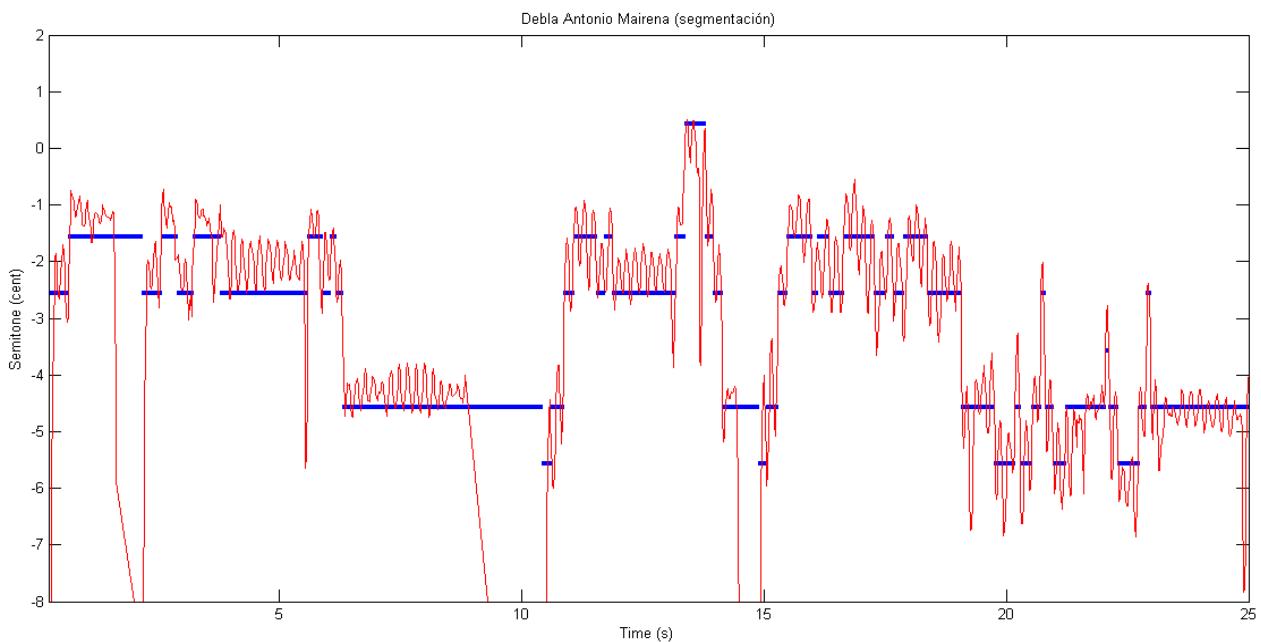


Figura 22: Representación de la segmentación completa de SMSTools 2 sobre una debla interpretada por Antonio Mairena.

Como se puede observar en la Figura 24 la segmentación baja considerablemente el número de datos utilizados, por ejemplo, en la debla que se representa, se pasa de 1567 componentes que se necesitan para describir el contorno melódico completo a 57 componentes tras la segmentación, lo que hace que al usar un algoritmo DTW, tan dependiente computacionalmente del número de elementos, se acelere notablemente el procesado de la información. El estudio de eficiencia se realiza en el capítulo siguiente.

■ SEGMENTACIÓN MEDIANTE ALGORITMO FITTING:

Como se explicó en el desarrollo del algoritmo, con este método de segmentación se obtiene un representación discreta óptima de la curva de contorno melódico, fijado un rango de tolerancia al error α , el cuál marca el grado de aproximación de la representación discreta a la real continua, esto es, a menor α más similar es la

aproximación a la real, pero por el contrario, mayor número de componentes poseerá, y a mayor α , se obtendrá una aproximación con mayor error pero con menor número de componentes. Se presenta entonces un problema de decisión para el usuario, al tener que encontrar una solución de compromiso entre la eficiencia y la eficacia del algoritmo al establecer un α determinado. Dicho problema dependerá de cada caso y se resolverá para el caso de las melodías del corpus musical que tratamos en el próximo capítulo al realizar pruebas con distintos α y determinar los tiempos de computación y los errores que éstos poseen.

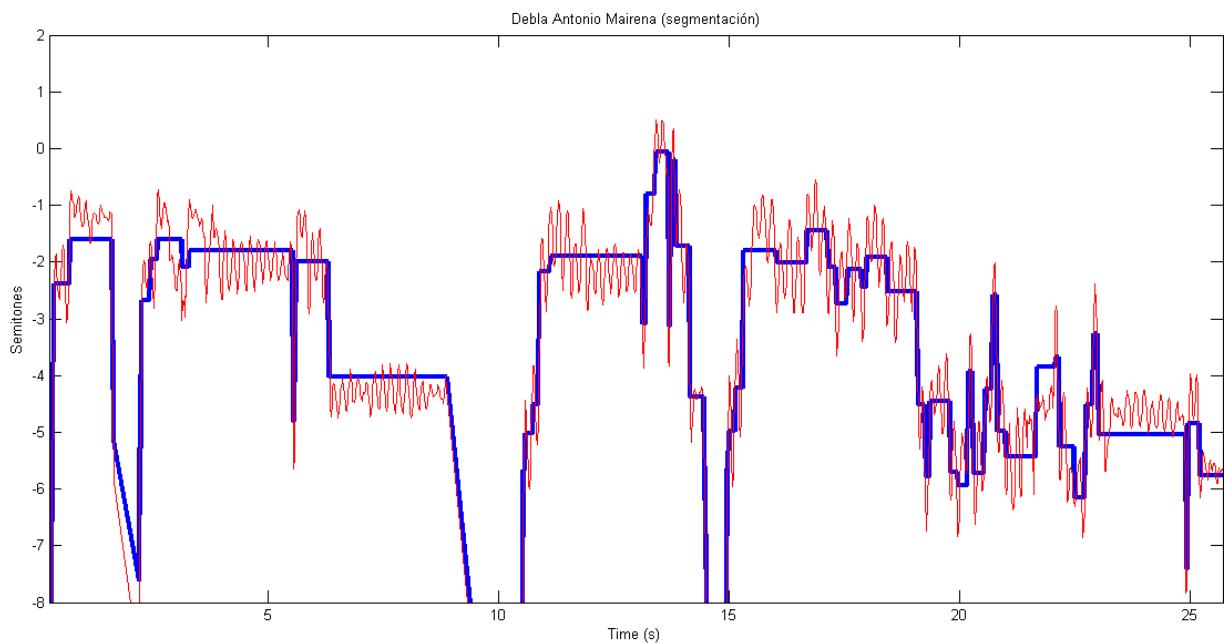


Figura 23: Segmentación completa con $\alpha = 2$.

La resolución mínima que ofrece el parámetro α es de 2 semitonos, es decir, la ventana se fija con un límite superior a 1 semitono y un inferior a -1. A continuación (Figura 23) se muestran segmentaciones con distintos valores de α .

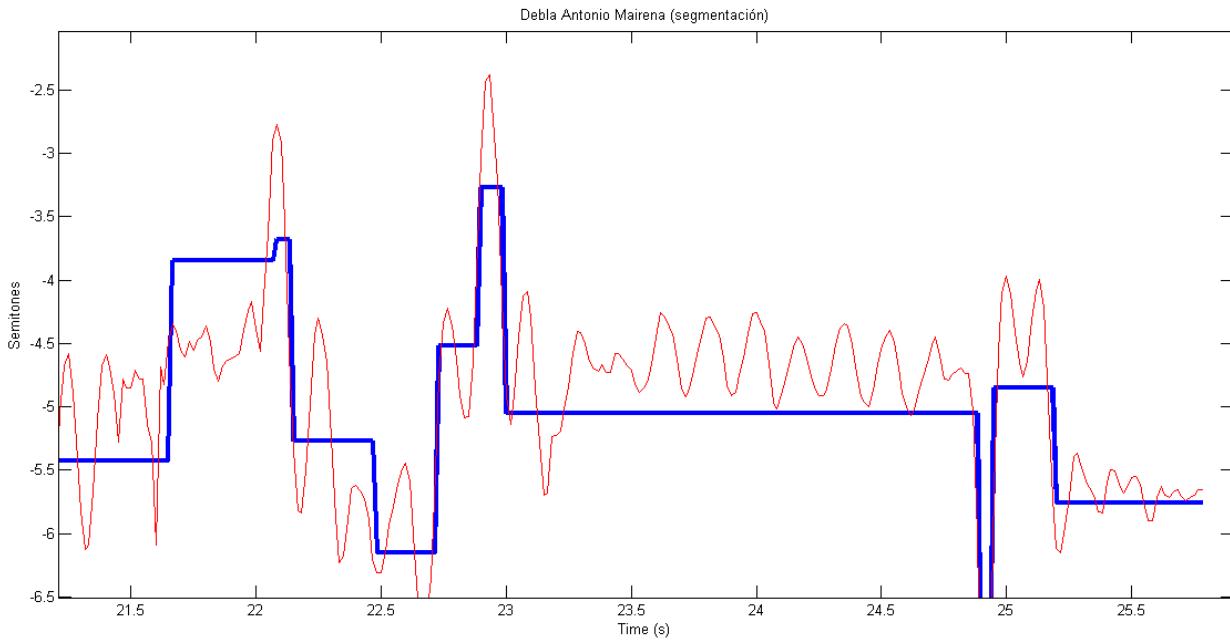


Figura 24: Detalle de la segmentación con $\alpha = 2$.

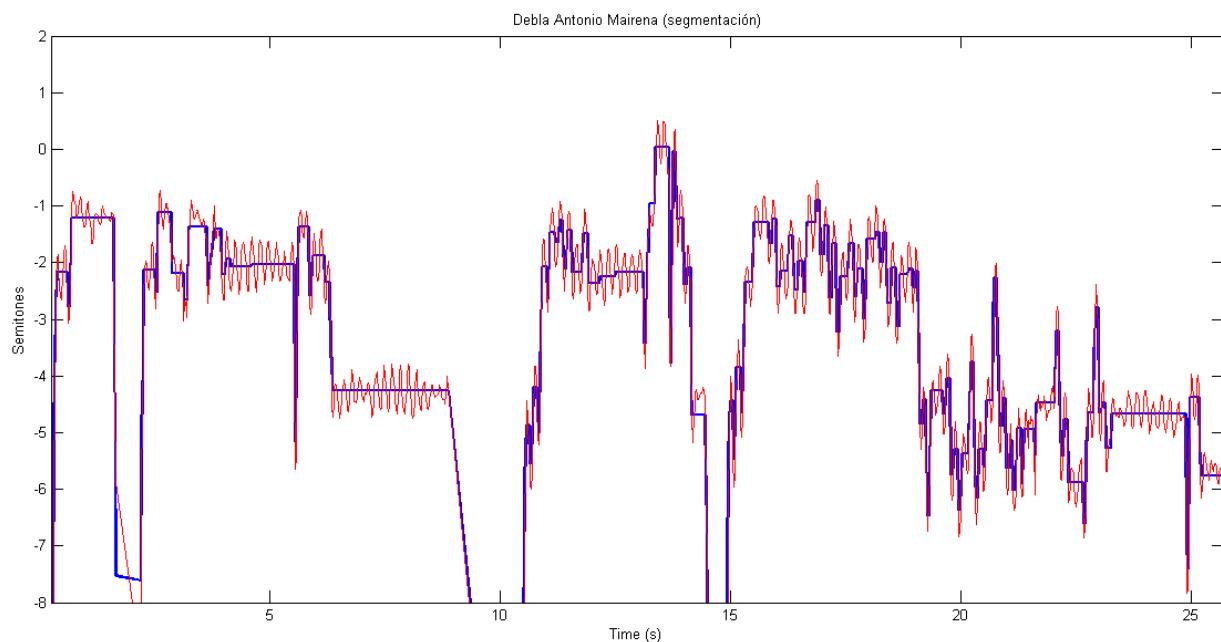


Figura 25: Segmentación completa con $\alpha = 1$.

Como se puede observar en las anteriores figuras a medida que se disminuye el parámetro α , la aproximación de la función discreta (azul) se ciñe mejor a la función continua (rojo), pero por el contrario, como se ha indicado antes, el número de componentes aumenta.

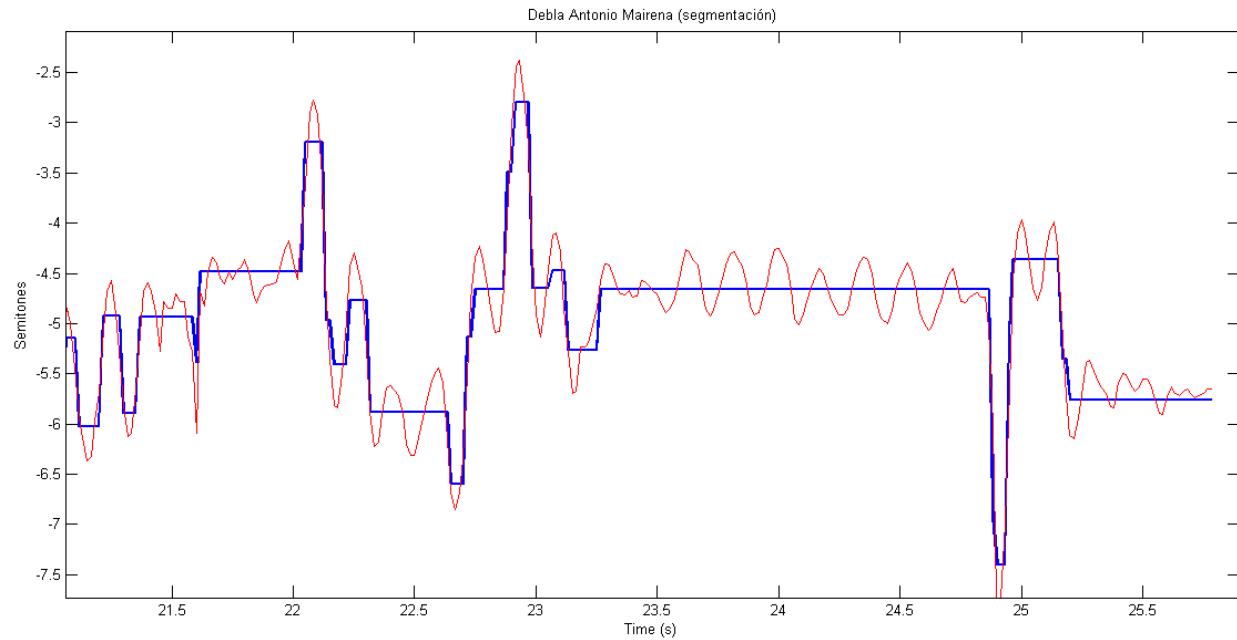


Figura 26: Detalle de la segmentación con $\alpha = 1$.

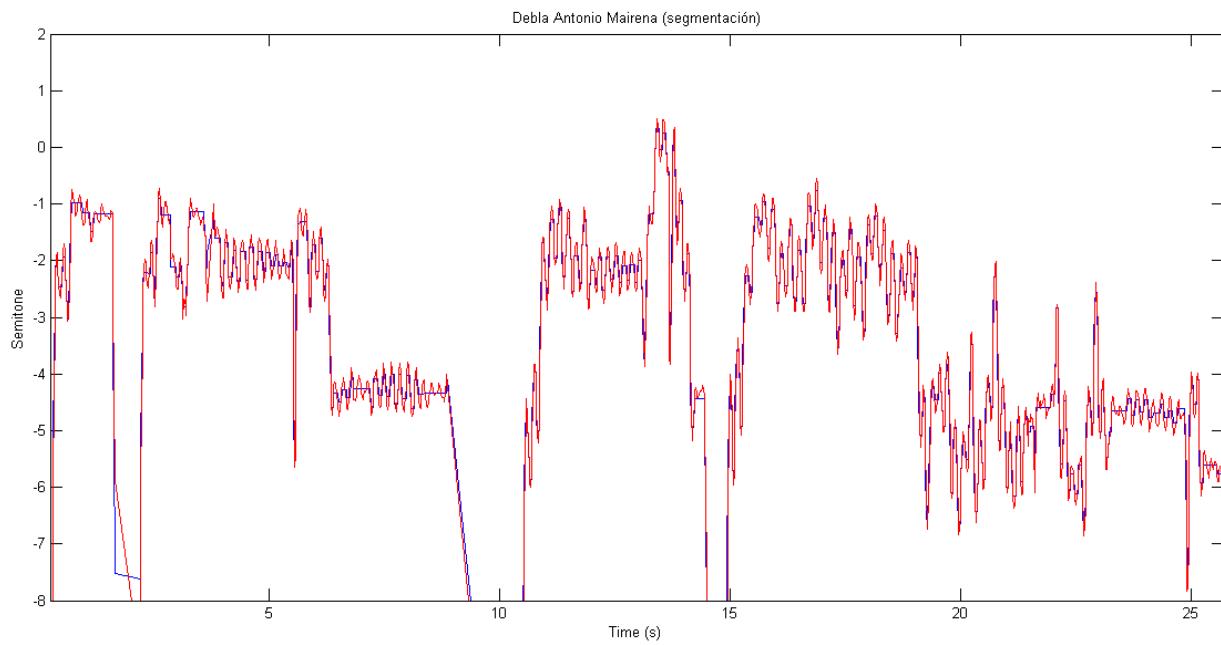


Figura 27: Arriba: Segmentación completa $\alpha = 0.5$.

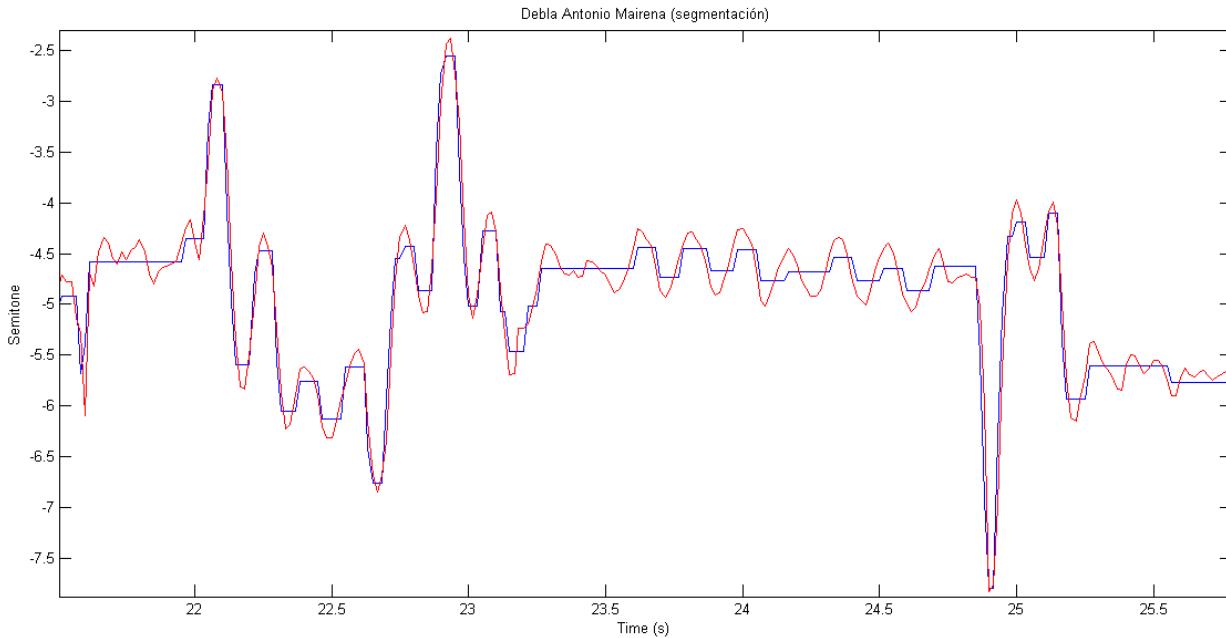


Figura 28: Detalle de la segmentación completa $\alpha = 0,5$.

α	Nº de componentes
2	70
1	174
0,5	398

Tabla 4: Variación del nº de componentes en función del parámetro.

- PASO 5: Aplicación del algoritmo DTW y obtención de la matriz de distancias:

En este paso de la estrategia se llega al último paso del algoritmo SDTW, que es la aplicación del algoritmo DTW clásico sobre el contorno segmentado de los audios que forman el corpus de cantes elegidos, y así poder construir la matriz de distancias entre los distintos cantes.

Como se vio en el Capítulo 2, la medida de similitud melódica elegida basada en el DTW es el *global warp cost*, que depende de los elementos del camino óptimo y del número de elementos del mismo, así que el aplicar el algoritmo DTW no bastará con obtener el valor $D(I, J)$ de la matriz de costos D , si no que habrá que tener en cuenta el valor de todos los puntos que definen el camino óptimo.

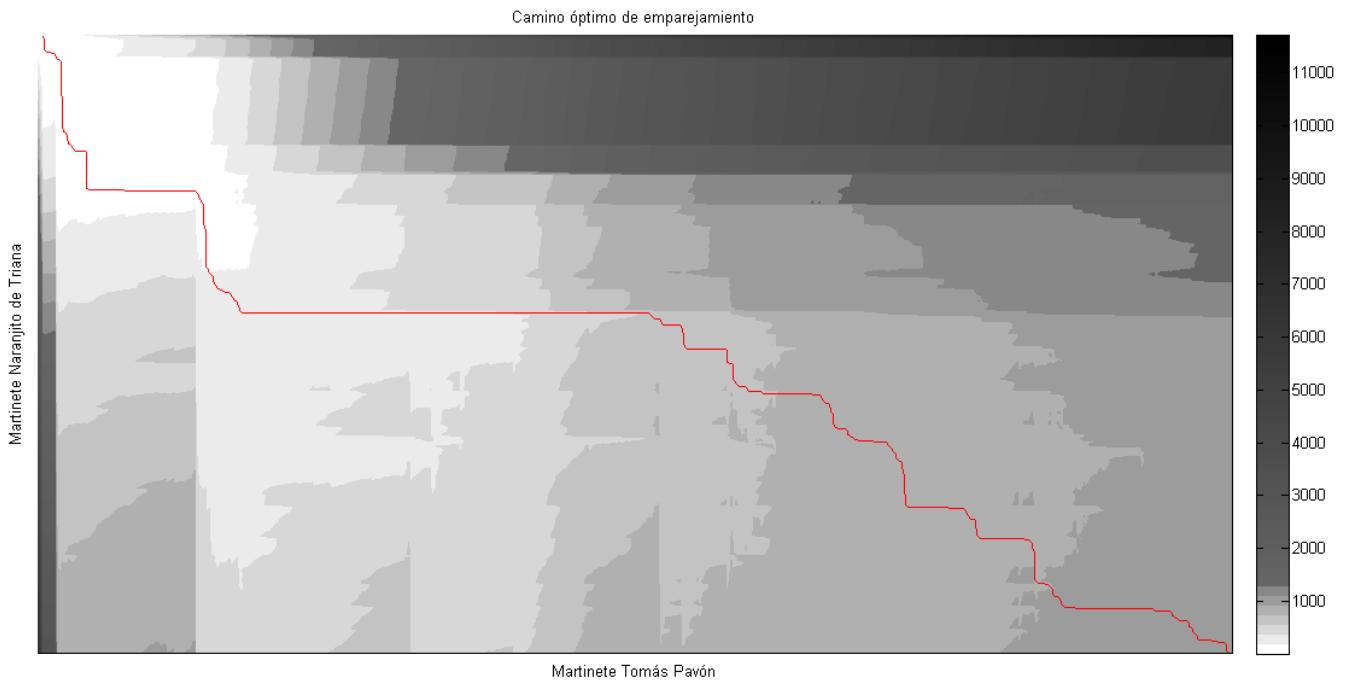


Figura 29: Representación del camino óptimo (rojo) sobre la matriz de costos D (escala de grises) para el estudio de similitud entre un martinete de Naranjito de Triana y otro de Tomás Pabón. $GC = 662,66$.

Como se puede ver en las Figuras 29 y 30, donde las zonas más oscuras representan las celdas de mayor coste y las más claras las de menor, el algoritmo de búsqueda del camino óptimo (línea de color rojo) hace que una vez construida la matriz de costos D según el método de construcción fijado por el algoritmo DTW, el camino óptimo se construya desde el punto $D(I,J)$ hasta el $D(0,0)$ eligiendo los elementos adyacentes de menor $D(i,j)$. También se puede observar que en ambas figuras las trayectorias de los caminos óptimos tienden a la diagonal de la matriz, este hecho será de interés en el próximo capítulo donde se comprobará que el uso de restricciones de trayectoria no penalizará en exceso la clasificación de los cantes.

El coste de emparejamiento entre dos cantes i y j se define como $GC_{i,j}$. Teniendo en cuenta las propiedades del global warp cost, la matriz de distancias de un corpus formado por n cantes se construye como se ve en la Tabla 5. La implementación del global warp cost se encuentra incluida en el archivo *Dtw_Sakoe_GC.m* del Anexo.

Una vez se ha creado la matriz de distancias entre todos los componentes del corpus a analizar, se puede dar por terminada la estrategia de resolución del algoritmo SDTW.

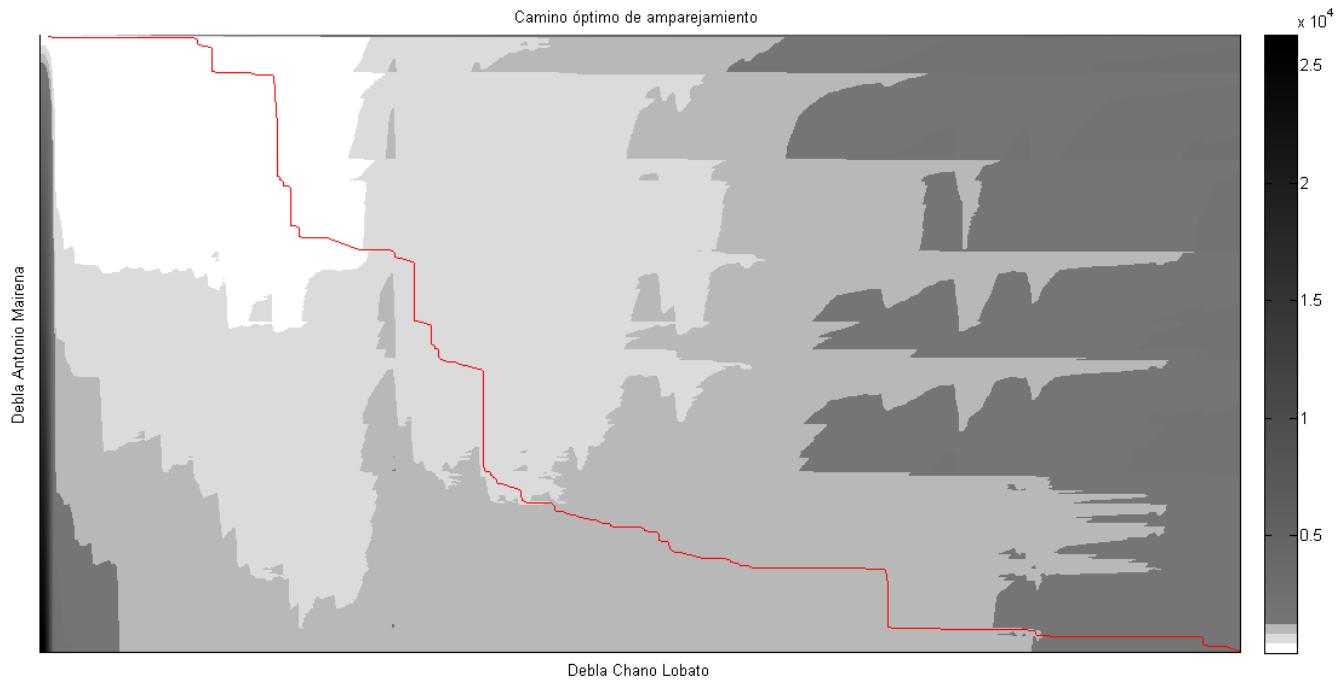


Figura 30: Representación del camino óptimo de una debla de A. Mairena frente a una de C. Lobato. $GC = 521,17$.

1	n
1	0	$GC_{1,j}$	$GC_{1,n}$
...	$GC_{i,1}$	0							...
	...		0						
				0					
					0				
						0			
							0		...
...	...							0	$GC_{n,n-1}$
n	$GC_{n,1}$	$GC_{n,n-1}$	0

Tabla 5: Matriz de distancias entre cantes.

Capítulo 4:

EVALUACIÓN

- 1. Corpus: Tonás.**
- 2. Medidas de rendimiento.**
- 3. Evaluación empírica: Precisión y eficiencia.**

CAPÍTULO 4: EVALUACIÓN

4.1 Corpus: Tonás.

4.1.1 Introducción

El corpus objeto de estudio en este proyecto está formado por un grupo grabaciones de dos palos flamencos: las deblas y los martinetes, pertenecientes al grupo de las tonás. A continuación, se va a realizar un breve estudio descriptivo sobre dicho grupo y sobre los palos que se van a analizar.

El flamenco se nutre de fuentes muy variadas entre las que destaca el repertorio de romances, tonás y otros cantes "a palo seco" (sin acompañamiento de guitarra). La debla y el martinete pertenecen al grupo de las tonás. Toná proviene del término español tonada, que significa "cantable" o "fragmento melódico". Se cree que muchas de estas tonadas, tonás en andaluz, provienen de antiguos romances y son partes desgajadas de los mismos que se conservan en la memoria del pueblo. Otros estilos que pertenecen al grupo de las tonás son las carceleras, la debla y, la matriz, la gama de las tonás.

El concepto de tonás no fue bien definido en el cante flamenco hasta la obra de Ricardo Molina y Antonio Mairena "Mundo y formas del Cante Flamenco" (Molina & Mairena, 1963). Hasta ese momento, estos cantes se denominan como martinetes, en el caso de Jerez, Cádiz o Málaga; como tonás; indistintamente como martinetes o tonás en Triana y El Puerto; o reservando el nombre de martinete a alguna de sus variedades y llamando tonás al resto.

A principios del s. XX el nombre de toná cae en desuso y en los discos de pizarra no aparece la palabra toná, pero sí martinete y, a veces, carcelera. En este contexto, el prestigio del que goza Antonio Mairena influye en que se adopte su propuesta relativa al concepto y la denominación de toná de modo generalizado.

Molina y Mairena definen la toná como coplas de cuatro versos octosílabos, rimados los pares de forma asonante e interpretadas a capella, sin acompañamiento de guitarra y sin marcar externamente el compás con palmas, bastón o pié. Algunos de los cantes con estas características tienen una denominación específica, como es el caso de los martinetes y las deblas. El nombre de carceleras, en cambio, puede aplicarse igual, a un estilo determinado de toná que a martinetes con letras de contenido alusivo a la vida en prisión.

Para Molina y Mairena el cante por tonás es el género, y los martinetes y deblas las variedades o especies de tonás.

En la actualidad se conservan varios tipos de martinetes, destacando el conocido como “martinete trianero”, desarrollado en los ambientes fragueros del barrio de Triana, Sevilla, que será el estilo elegido aquí para su estudio. Al contrario que las tonás, que usan como tonalidad la cadencia andaluza, los martinetes se realizan en tono mayor. Con respecto al ritmo, se cantan, como todas las tonás, sin un compás determinado. Sin embargo, Antonio Ruiz El Bailarín realizó una versión bailable eligiendo el compás de la seguiriya para realizarlo y desde entonces se ha generalizado la interpretación del martinete sobre este metro rítmico. Finalmente, respecto a la copla, es una cuarteta octosílaba (romance) y se canta sin repetición de ningún verso o con repeticiones cuando se hace redoblado. Suele iniciarse el martinete con un onomatopéyico tran-tran, imitando el sonido del martillo sobre el yunque.

Para una descripción más profunda de algunos cantes del grupo de tonás, se pueden consultar los trabajos: (Vega & Ruiz, 1990), (Lefranc, 2000), (Castro, Los cantes sin guitarra en el flamenco, antecedentes musicales y modalidades., 2010), (Castro, 2010) y (Mora, Gómez, Gómez, Escobar-Borrego, & Díaz-Bañez, 2010).

4.1.2 Los cantes por Tonás para un estudio computacional

Quizá sean los *cantes por tonás* de los cantes que ofrecen más dificultades a la hora de estudiarlos computacionalmente, debido a que las diferentes variedades existentes dependen más de la idiosincrasia del cantaor que de variables locales o de estructuras musicológicas bien definidas. La inexistencia del armazón armónico proporcionado por la guitarra y las irregularidades de afinación de muchos intérpretes entorpecen la identificación de los modos musicales y otro tanto cabe decir de una estructura rítmica que va *ad libitum* el intérprete puede variar el tempo, la velocidad a la que interpreta el cante, como lo deseé. Por todo ello, consideramos que representan la esencia del cante flamenco.

Las tonás son cantes a capella, sin instrumentación, en algunos casos con alguna percusión tales como nudillos o golpes de martillo en el caso de martinetes, en el argot flamenco son denominados *cantes a palo seco*.

Tradicionalmente se les ha considerado cantes de las fraguas. En efecto, algunos intérpretes históricos han sido herreros, y ciertas letras aluden a las labores de fragua. Lo mismo ocurre con las letras de tema carcelario, que justifican la denominación de carceleras que reciben algunas de las tonás.

La casi total ausencia de grabaciones de tonás en cilindros y discos de pizarra influye en las dificultades para su estudio. De los 5.500 títulos de la discografía en 78 rpm recogida en el Diccionario de Blas Vega y Ríos Ruiz (Vega & Ruiz, 1990) solo hay 6 grabaciones de tonás, realizadas entre 1922 y 1933 por Centeno, Tenazas, Cepero, El Gloria, Mazaco y El Cuacua. Esto confirma lo dicho por Demófilo (Machado y Álvarez, 1975 reedición de 1881) sobre el olvido en que habían caído estos cantes.

Sobre el número de tipos diferentes de tonás existentes se ha hablado mucho, y existen diferentes versiones. Por citar un ejemplo, Demófilo (Machado y Álvarez, 1975 reedición de 1881) enumera treinta y un tipos diferentes. Pierre Lefranc (Lefranc, 2000), sin embargo, identifica catorce. El repertorio de tonás diferentes merece un estudio independiente, que permita una clasificación basada más en criterios musicológicos que de filiación, local o de atribución, como hasta ahora se ha hecho.

A grandes rasgos, dentro del género de las tonás pueden distinguirse los siguientes cantes diferenciados:

- Martinetes.
- Carceleras.
- Deblas.
- Saetas¹.
- Tonás propiamente dichas.

Conviene indicar que el grupo de las saetas presenta características que pueden aconsejar estudiarlas de manera independiente al resto de tonás. Para un estudio musicológico de las saetas se puede consultar. (Berlanga, 2001) y (Kramer & Plenckers, 1998).

Respecto al grupo de tonás propiamente dichas, se define por exclusión: pertenecen a él los cantes que no se incluyen dentro de algunos de los grupos restantes.

4.1.3 Descripción y características musicales: debla y martinete.

Para nuestro estudio se van a seleccionar dos estilos dentro del grupo tonás: la debla que popularizó Tomás Pavón y el martinete trianero conocido como “martinete de Juan Pelao de Triana”. Un estudio musicológico de estos cantes se puede consultar en el trabajo (Mora, Gómez Martín, Gómez, Escobar-Borrego, & Díaz-Báñez, 2010). Hacemos aquí un breve resumen de tal artículo por completitud.

○ DEBLAS:

La debla es una variante o especie de las tonás. En general, se caracteriza por su gran ornamentación melismática, más abrupta que el resto de las canciones de este estilo, que hace muy exclusiva su melodía. Las deblas se caracterizan por un contorno melódico en particular.

La versión actual de la debla se debe a una aportación de Tomás Pabón en 1948, que la había aprendido de su suegro Antonio El Baboso. Es indudable que se trata de una

¹ En las conocidas como “saetas flamencas”, se hace uso de melodías y ornamentación que pertenecen a ciertas tonás.

toná trianera, pero algunos clásicos dudan que fuera la verdadera debla de la que hablaba Demófilo (Machado y Álvarez, 1975 reedición de 1881).

La letra que graba Pabón y que se ha difundido es insustancial. Ha sido modificada primero por Antonio Mairena y después por otros cantaores para hacer alusiones a persecuciones sufridas por los gitanos o a figuras trianeras legendarias.

Como otras tonás, se trata de un cante de cuatro versos que sigue el esquema 8-/8a/8-/8a. Seguramente por influjo de las características vocales de Pabón, este cante se presta a la aparición de melismas abundantes y prolongados, ejecutados con pocas respiraciones.

Las variables musicales que caracterizan las diferentes variantes dentro del estilo debla son las siguientes según el trabajo (Mora, Gomez Martin, Gómez, Escobar-Borrego, & Díaz-Báñez, 2010):

- I. **Comienzo por la palabra *iAy!*:** *iAy!* es una interjección de dolor, característica de fuerte idiosincrasia en la música flamenca.
- II. **Vinculación de *iAy!* con el resto de los versos:** El *iAy!* inicial puede ser unido con el resto de los versos o simplemente separado de ellos.
- III. **La nota inicial:** Hace referencia a la primera nota del terceto. Normalmente es el sexto grado de la escala (VI), aunque también puede aparecer el quinto (V).
- IV. **Tendencia de la melodía en el primer hemistiquio:** (un hemistiquio es la mitad o el fragmento de un verso, dividido por una pausa en la entonación) La tendencia puede ser ascendente (appoggiatura rápida en V, a continuación, la progresión VI-IV), simétrica (III-VI-IV) o ascendente.
- V. **Repetición del primer hemistiquio:** La repetición puede ser del hemistiquio completo o de parte de él.
- VI. **Cesura:** La cesura (del latín caesura: cortadura) es el espacio o pausa dentro de un verso separando dos partes.
- VII. **Tendencia de la melodía en el segundo hemistiquio:** La tendencia se define igual que el primer hemistiquio.
- VIII. **Mayor grado en el segundo hemistiquio:** Es el grado más alto encontrado en el segundo hemistiquio. Por lo general, se alcanza el séptimo grado, aunque pueden aparecer quinto y sexto grado.

○ MARTINETES:

Como se ha visto antes, los martinetes también son considerados una variante de las tonás. Se diferencian de las deblas en la temática de sus letras y en su estructura melódica que siempre termina en modo mayor. La temática de las deblas es usualmente triste, cantadas sin acompañamiento de guitarra, como el resto de tonás. Sin embargo, los martinetes se suelen acompañar con el sonido de un martillo golpeando un yunque.

Los aspectos musicales que caracterizan el martinete trianero son (Mora, Gomez Martin, Gómez, Escobar-Borrego, & Díaz-Báñez, 2010):

- I. **Repetición del primer hemistiquio:** Como en las deblas, la repetición puede ser completa o parcial.
- II. **Clivis/flexa (movimiento alto/bajo) en el final del primer hemistiquio:** Normalmente, caída IV-III o IV-IIIb. El final más común para un terceto es el cuarto grado, cuyo sonido es sostenido hasta alcanzar la cesura. Algunos cantes utilizan a modo de cierre el III o IIIb.
- III. **Uso del mayor grado en los dos hemistiquios:** La práctica habitual es llegar hasta el cuarto grado, y algunos cantantes llegan al quinto grado.
- IV. **Nota final del segundo hemistiquio:** El segundo hemistiquio termina cayendo al segundo grado.

Se eligen estos dos estilos porque tienen un contorno melódico claramente diferenciado como se puede ver en la transcripción de la Figura 31. Por tanto, estamos abordando un estudio inter-estilo, esto es, definiendo una medida de similitud que discrimina estilos diferentes de cantes. En otros contextos, estudios intra-estilo, podríamos estudiar las distintas variantes de martinetes que interpretan distintos cantaores.

Una alternativa sería diseñar medidas de similitud combinadas en las que se involucra a demás del contorno melódico otras otros parámetros de carácter musical, como se propone en el artículo (Mora, Gomez Martin, Gómez, Escobar-Borrego, & Díaz-Báñez, 2010).

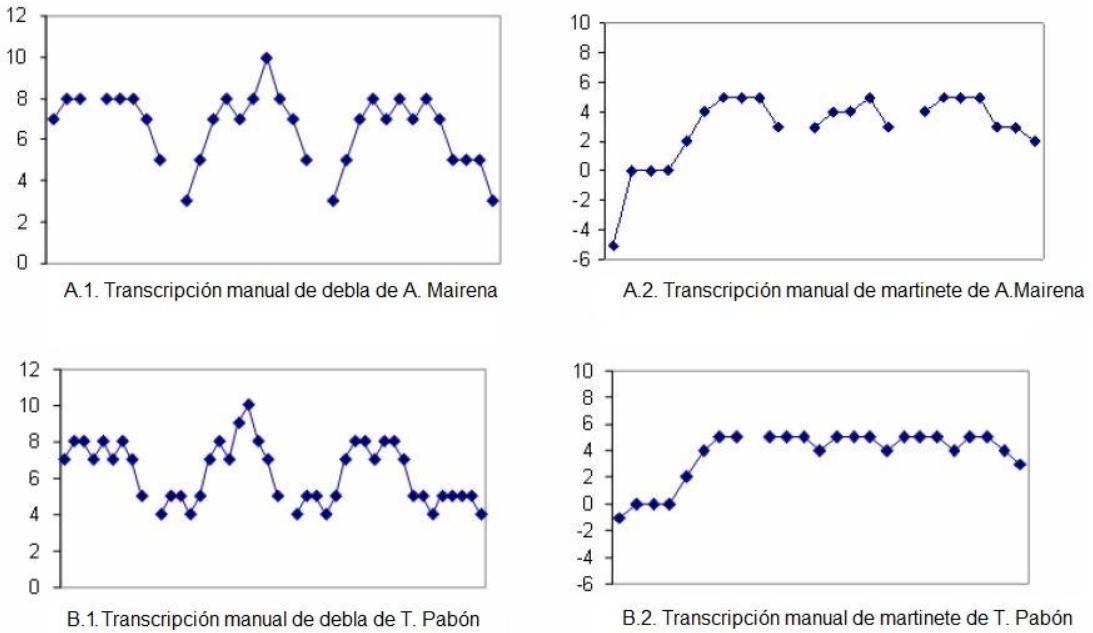


Figura 31: Transcripciones manuales realizadas por el Dr. Joaquín Mora.

4.1.4 El Corpus Tonas. Un metadato de acceso libre.

El corpus (Tonás) fue creado en el contexto de un estudio sobre similaridad y clasificación de estilos de cantes flamencos a capella (tonás) por el grupo COFLA. Los 72 fragmentos son monofónicos y su duración media es de 30 segundos. También se ofrece en el corpus una transcripción melódica manual, como se explica a continuación:

- TRANSCRIPCIÓN MELÓDICA SEMIAUTOMÁTICA:

Tres sujetos participaron en el proceso de transcripción: un músico con un conocimiento limitado de la música flamenca, un experto en la música flamenca y un cantaor flamenco. El músico realizó primero anotaciones manuales detalladas. Como su conocimiento de la música flamenca era muy limitado, se esperaba que no usara el conocimiento implícito en el estilo, para así no condicionar la transcripción manual y la realizar de una forma objetiva. Algunos ejemplos fueron corregidos por un experto en flamenco para establecer un criterio que se aplicó además a refining manual de transcripciones. Por último, el cantaor verificaría de forma independientemente las transcripciones manuales.

A fin de reunir las anotaciones manuales, se dotó a los sujetos de un interfaz para visualizar la forma de onda y la frecuencia fundamental (f_0) en centésimas de tono (en una representación de teclado de piano). Los sujetos podían escuchar la forma de onda y la transcripción sintetizada, durante la edición de los datos melódicos hasta que

quedaron satisfechos con la transcripción. Se observó que todavía había un grado de subjetividad en cuanto a la diferenciación entre los ornamentos y los cambios de tono.

Los criterios para la transcripción manual de la colección de Tonás se referencian en (Gómez & López, 2013).

○ ARCHIVOS INCLUIDOS:

- Archivos de audio: 72 archivos de audio monofónicos en 16 bit. Formato WAV y frecuencia de muestreo 44.1 kHz.
- Estructura de datos: TONAS - _____.txt: Nombre del archivo (Primera columna), título (segunda columna) y cantaos (tercera columna) cada archive en formato Unicode UTF-8 para preservar los acentos y la ñ.
- Transcripción melódica: 3x72 archivos. Por cada uno de los 72 archivos de audio se crean los siguientes archivos de transcripción:
 - Frecuencia fundamental (*nombre del archivo.f0.Corrected*): Estimación de la f0 basada en (Gómez & Bonada) donde se han corregido manualmente algunos errores de la f0. También contiene valores cuantificados de note.f0 a escala referenciados a la escala temperada. Cada una de las 4 columnas del archivo corresponden a:
time_seconds energy estimatedf0_Hz estimatednotef0_Hz
 - Transcripción segmentación Notes (*nombre del archivo.notes.Corrected*): Contiene una línea por cada nota, incluyendo la siguiente información:
onset_seconds, duration_seconds, pitch_MIDI, energy

4.2 Medidas de rendimiento.

A continuación se describen las herramientas que usaremos para comparar el rendimiento de los distintos procedimientos de similitud considerados en este proyecto.

4.2.1 Análisis filogenético: Generación de grafos.

Con objeto de estudiar las relaciones existentes entre las diferentes interpretaciones que forman el corpus que se va a analizar, se va a hacer uso de una metodología perteneciente al campo de la Biología Computacional o Bioinformática, lo que se conoce como análisis filogenético. El análisis filogenético es la disciplina que se dedica al estudio de las relaciones evolutivas entre las distintas especies, reconstruyendo la historia de su diversificación desde el origen de la vida en la Tierra hasta la actualidad. Gracias al análisis filogenético se puede realizar una clasificación de los organismos,

analizando normalmente cuán diferentes son dos materiales genéticos dados, como podemos ver en la Figura 32.

En nuestro caso, se va a sustituir el material genético por las diferentes interpretaciones de cantes que forman el corpus, y de ese modo se podrán observar diferentes comportamientos que hagan que se planteen diversas cuestiones sobre la evolución de los cantes, la influencia de unos cantaores en otros o la clasificación de los diferentes estilos.

Para reflejar gráficamente esos resultados, vamos a hacer uso de técnicas de Bioinformática (Huson, 1998) que son capaces de reconstruir un árbol que refleja las relaciones entre sus miembros. Un análisis muy similar a éste se realizó por primavez en un artículo sobre la evolución rítmica del Flamenco (Díaz Bañez, Farigu, Gómez, Rappaport, & Toussaint, 2005) en el que se utilizaron dos distancias de similitud, la distancia cronotónica y la distancia de permutación dirigida, obteniendo los siguientes árboles filogenéticos ver Figura 33. En ese caso, el “material genético” que se utiliza son los diferentes patrones rítmicos ternarios que se encuentran en el Flamenco.

A partir de las matrices de distancias obtenidas con las diferentes medidas de similitud estudiadas, se van a generar lo que se conoce como árbol filogenético. Para ello, se hará uso de la herramienta Splitstree versión 4.8 (Huson & Bryant), que permitirá introducir manualmente una matriz de distancias, generando un grafo plano con la propiedad de que la distancia en el dibujo entre dos nodos refleja, tanto como es posible, la verdadera distancia entre las dos melodías correspondientes en la matriz de distancias.

La técnica está basada en algoritmo UPGMA (Unweighted Pair Group Method using Arithmetic averages) (Sneath & Sokal, 1973) que utiliza la distancia evolutiva, y va agrupando las “parejas más cercanas” de forma iterativa. Por otro lado, la herramienta calcula también el índice de descomposición, LSFit, una medida de la bondad del ajuste del grafo completo.

Phylogenetic Tree of Life

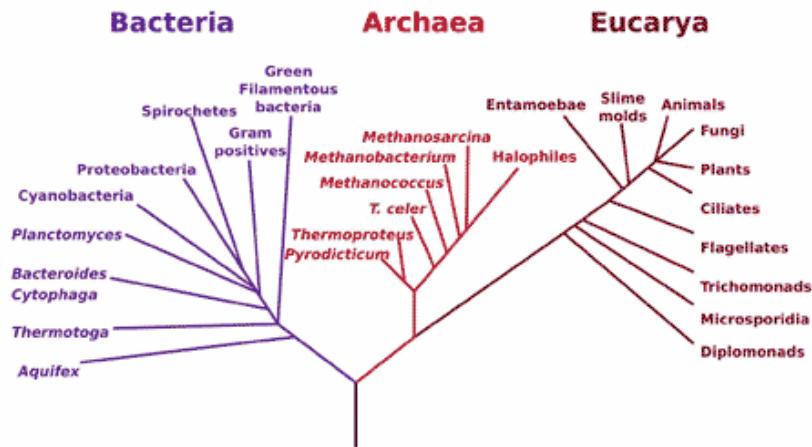


Figura 32: Árbol filogenético de la vida.

Un valor bajo de este índice, que se calcula dividiendo la suma de todas las distancias aproximadas en el grafo por la suma de todas las distancias originales en la matriz de distancias, indicaría que el grafo obtenido no refleja fielmente los valores obtenidos en la matriz de distancias.

Como se verá en capítulos posteriores, el uso de esta herramienta de Bioinformática resulta de una gran utilidad, pues permite visualizar los resultados que hemos obtenido a lo largo de todo el análisis computacional y representarlos en forma de árbol o grafo, lo que permitirá hacer una primera validación de resultados atendiendo a los *cluster* o grupos que se formen. Así mismo, y en el contexto de un posible estudio musicológico más completo, estos grafos podrían ayudar a extraer conclusiones sobre la proximidad de las interpretaciones, gracias a la facilidad para ser visualizados.

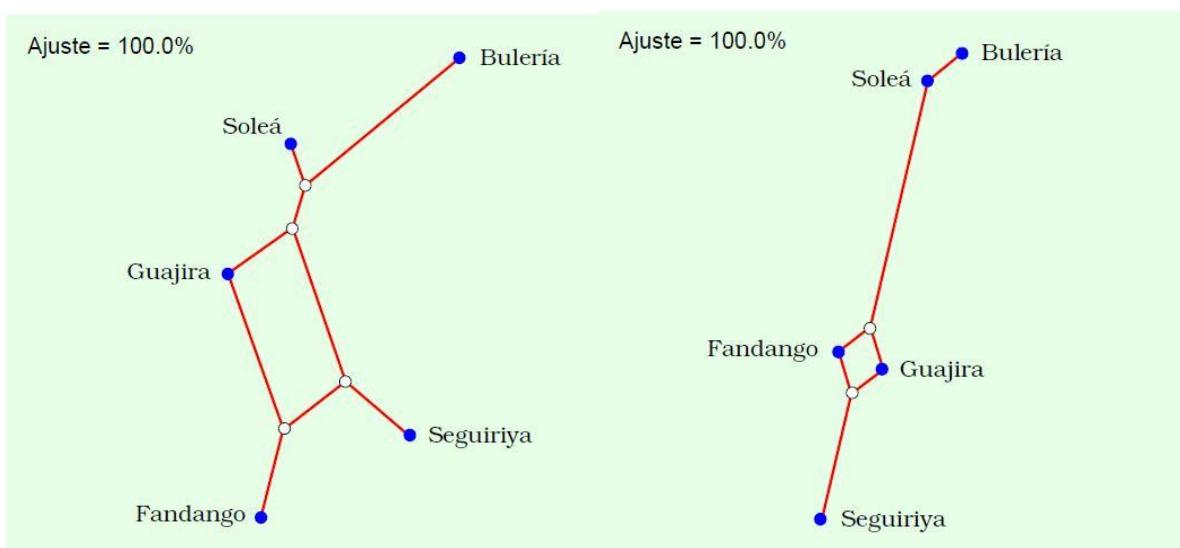


Figura 33: Arriba: Árbol filogenético con la distancia de permutación dirigida. Abajo: Árbol filogenético con la distancia cronotónica.

4.2.2 Test de Mantel.

Con el fin de evaluar los distintos métodos de segmentación, se realizará el test de Mantel (Mantel & Valand, 1970) para analizar la correlación entre las matrices de distancia obtenidas con las distintas segmentaciones y la matriz de distancia obtenida sobre el contorno melódico sin segmentación.

El test de Mantel estima el grado de correlación existente entre dos matrices X e Y. La hipótesis nula de esta técnica (H_0) postula que las distancias/similitudes entre las variables de la matriz respuesta Y no están linealmente correlacionados con las correspondientes distancias/similitudes en la matriz modelo X. Se trata, por tanto, de evaluar si la asociación (positiva o negativa) es más robusta de lo que cabría esperar por puro azar.

El estadístico del test de Mantel (Z_M) se calcula mediante la suma de los productos cruzados de los valores de las dos matrices de similitud/distancia, excluyendo la diagonal principal que sólo contiene valores triviales (0 en el caso de las matrices de distancias y 1 en el caso de las matrices de similitudes).

$$Z_M = \sum_{i,j=1}^n X_{ij}Y_{ij}$$

Donde X_{ij} e Y_{ij} son los elementos de las matrices X e Y, respectivamente.

El estadístico Z_M se expresa en unidades arbitrarias y sus implicaciones son difíciles de entender, por lo que suele utilizarse el coeficiente estandarizado de Mantel. Para esto se calcula el producto cruzado de las dos matrices dividido por $[(n(n-1)/2)-1]$ cuyo rango está entre -1 y +1, y se acerca a uno en caso de fuerte correlación. Se comporta como un coeficiente de correlación, y se denomina r_M .

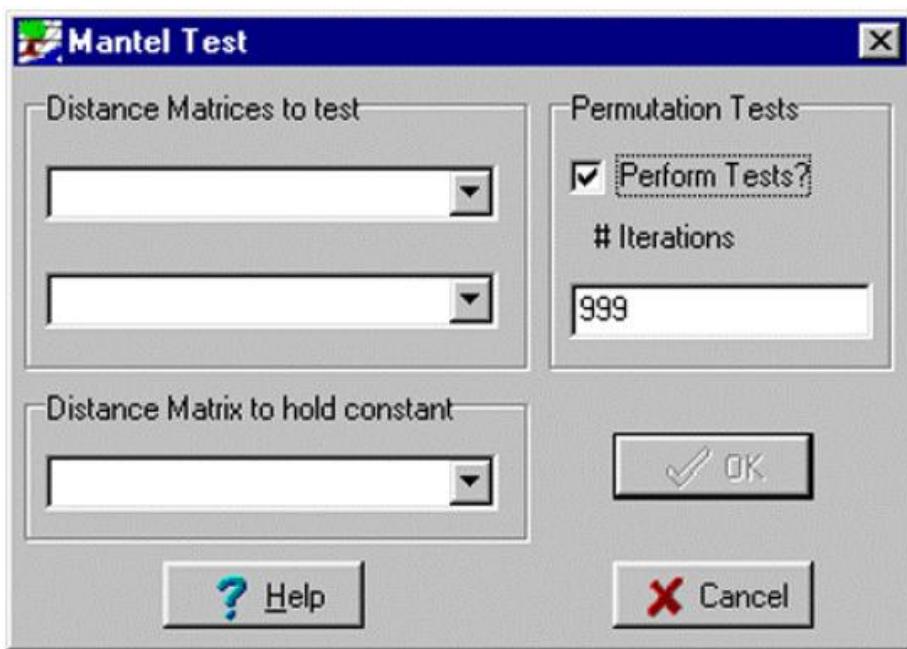


Figura 34: Ventana de test de Mantel.

Para el cálculo del estadístico de Mantel y su coeficiente estandarizado, se ha utilizado el programa de análisis de patrones *PASSaGE v2.0* (Rosenberg & Anderson, 2013). Al cual se le introducen las dos matrices (ver Figura 34) de distancias a comparar (una de ellas siempre es la que no proviene de una segmentación) y se obtienen información del tipo:

Observed Z = 6068490.0200
Correlation = 0.6955

4.2.3 Precisión y efectividad.

Con el objeto de cuantificar la bondad del algoritmo propuesto en este proyecto se llevará a cabo un estudio de precisión y efectividad del mismo en relación a la clasificación de los cantes que realiza. La precisión y la efectividad, conocidas por su nombre en inglés *Precision & Recall* (Powers, 2011) es una métrica empleada en la medida del rendimiento de los sistemas de reconocimiento de patrones.

Para tareas de clasificación, la precisión para una clase C es el número de verdaderos positivos (TP) (es decir, el número de elementos pertenecientes a la clase C que se clasifican correctamente en la clase C) dividido por el número total de elementos etiquetados como pertenecientes a la clase C (es decir, la suma de los verdaderos positivos y falsos positivos (FP), que son elementos incorrectamente etiquetados como pertenecientes a la clase C). La efectividad para una clase C se define como el número de verdaderos positivos dividido por el número total de elementos que en realidad pertenecen a la clase C (es decir, la suma de los verdaderos positivos y falsos negativos).

(FN), que son elementos que no fueron etiquetados como pertenecientes a la clase C pero deberían haber sido).

- Precisión para una clase C: $P(C) = \frac{TP}{(TP+FP)}$
- Efectividad para una clase C: $R(C) = \frac{TP}{(TP+FN)}$

Una medida que combina precisión y eficiencia es la llamada *F-measure* (F) (van Rijsbergen, 1979) que se entiende como la media harmónica entre precisión y eficiencia.

$$F = 2 \cdot \frac{P(C) \cdot R(C)}{P(C) + R(C)}$$

Finalmente se utilizará una medida de proporción de resultados verdaderos en la clasificación total, llamada *Accuracy* (Olson & Delen, 2008), entendiendo como resultados verdaderos tanto los verdaderos positivos como los verdaderos negativos.

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN}$$

El código de las distintas métricas de precisión se puede ver en el archivo *PR.m* del Anexo.

En una clasificación, una puntuación de precisión de 1,0 para una clase C significa que todos los elementos etiquetados como pertenecientes a la clase C, pertenecen a la clase C (pero no dice nada sobre el número de elementos de la clase C que no estaban etiquetados correctamente), mientras que una eficiencia de 1,0 significa que cada elemento de la clase C fue etiquetado como perteneciente a la clase C (pero no dice nada acerca de cuántos otros artículos fueron incorrectamente también etiquetados como pertenecientes a la clase C). Suele existir una relación inversa entre la precisión y la eficiencia, donde es posible aumentar uno a costa de la reducción de la otra. Los términos verdadero positivo, falso positivo, verdadero negativo y falso negativo, comparan los resultados del clasificador utilizado frente a un juicio externo, verdad subyacente o también conocida como *Ground truth*. Los términos positivos y negativos se refieren a la predicción del clasificador (a veces conocido como la expectativa), y los términos verdaderos y falsos se refieren a si esa predicción se corresponde con el juicio externo o verdad subyacente (a veces conocido como la observación).

<u>Categoría C</u>	JUICIO EXTERNO O GROUND TRUTH	
PREDICCIÓN DEL CLASIFICADOR	VERDADERO POSITIVO	FALSO POSITIVO
	FALSO NEGATIVO	VERDADERO NEGATIVO

Tabla 6: Términos de clasificación.

La verdad subyacente o *Ground truth* utilizada para determinar los términos de clasificación será el etiquetado correspondiente al corpus (Tonás), ya que como se ha visto en apartados anteriores, es una clasificación realizada por expertos en flamenco, por lo que se utilizará como juicio externo.

Como método de clasificación se usará el método del *Vecino más cercano* conocido por sus siglas en inglés *k-NN* (*k-Nearest Neighbor*) (Altman, 1992), el cual se utilizará como *método de clasificación supervisado*, es decir, se conoce de ante mano las clases. Éste es un método de clasificación no paramétrico, que predice “valores” o “pertenencias” a una determinada clase basándose en la clase de k elementos más cercanos, donde k es un número entero, por ejemplo, si $k=1$, los elemento a clasificar pertenece a la clase del primer elemento más cercano a él.

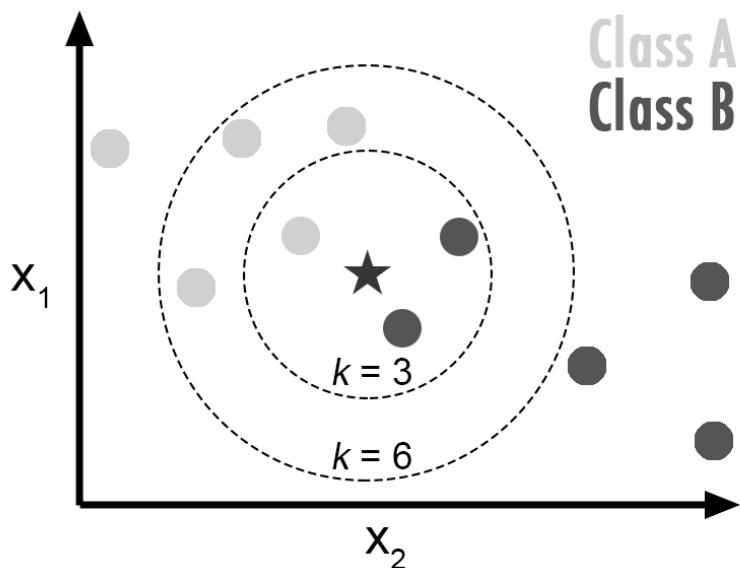


Figura 35: Ejemplo de clasificación *k-NN*, con $k=3$ y $k=6$.

El *k-NN* es un método de clasificación basado en la distancia. En general, existen distintas variables métricas usadas con este método, entre las que destacan:

- Distancia euclídea y euclídea al cuadrado.
- Distancia de Minkowski
- Distancia de Manhattan

En este caso la variable métrica utilizada será la medida de similitud melódica basada en el *DTW*, *GC* (*global warp cost*) a la que se ha hecho referencia en capítulos anteriores. Por tanto se entenderá que a mayor cercanía un elemento de otro, menor *GC*, por lo que consecuentemente más similar son ambos elementos.

Uno de los factores más determinantes del método *k-NN* es la elección del parámetro k el cual fija el número de vecinos más cercanos que hay que tener en cuenta para la clasificación del elemento. La mejor opción de k depende del número de elementos a clasificar, por lo general, mayores valores de k reducen el efecto del ruido sobre la

clasificación, pero hacen que los límites entre las clases difusos, es decir hace que las clases sean menos distintas. En teoría, para un número infinito de elementos a mayor k mejor clasificación. Un buen k se puede seleccionar mediante diversas técnicas heurísticas (por ejemplo hiperparámetro), por lo que la elección del valor tiene que marcar un equilibrio entre la definición de las clases y la precisión y efectividad de la clasificación.

Al ser un tema de compleja solución, que no es objeto de estudio en este proyecto, se opta por la elección de la llamada *Regla de Oro* (Veksler, 2002) que establece que:

$$k = \sqrt{n}$$

Donde n es el número de elementos a clasificar, el desarrollo de este método de clasificación se puede ver el archivo *knn.m* del Anexo.

En nuestro caso, al contar con 12 elementos a clasificar de cada clase, el k elegido variará entre 3 y 4. Para elegir definitivamente el número de vecinos evaluaremos las distintas métricas con los dos valores de k propuestos. En la Tabla 7, se relacionan los valores de medidas obtenidos para los contornos melódicos de los 24 cantes de deblas y matinetes.

k	PRECISION	RECALL	F-MEASURE	ACCURACY
3	0.9412	0.6667	0.7805	0.3542
4	0.9333	0.5833	0.7179	0.3125

Tabla 7: Evaluación de k.

Como consecuencia de los resultados en la tabla, tomaremos como el número de vecinos más cercanos que hay que tener en cuenta para la clasificación $k=3$. Conviene decir que en las pruebas estadísticas de rendimiento la elección de los parámetros varían según los experimentos sean considerados “de laboratorio” o “de campo”. Debido a que las grabaciones del corpus carecen de homogeneidad en cuanto a calidad y temporalidad podemos considerar nuestro experimento como de campo y quizás el valor del parámetro k sea “exigente” en nuestro caso. De cualquier forma, no existen trabajos previos sobre grabaciones de flamenco en los que se indique el valor más adecuado del parámetro.

Las medidas de precisión y eficiencia se obtendrán tanto para cada una de las clases, es decir, para deblas y martinetes, como para el conjunto total de cantes.

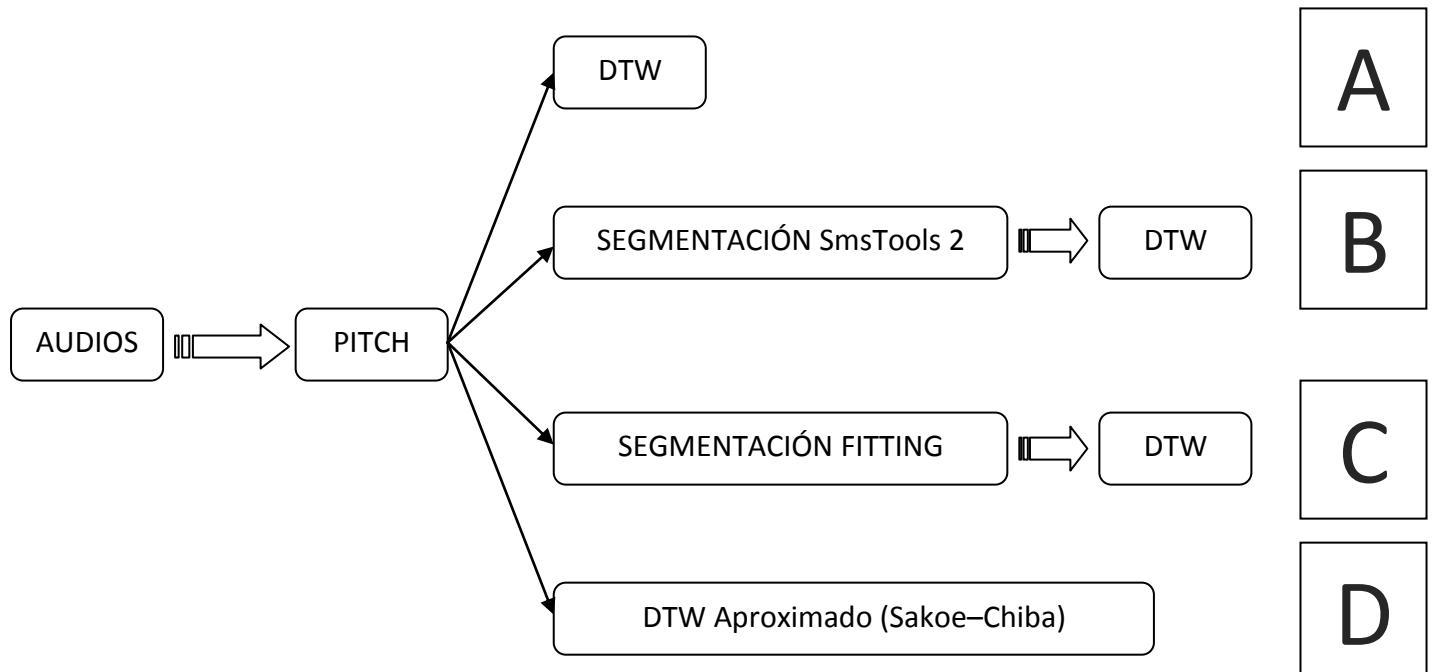
4.3 Evaluación empírica: Precisión, efectividad y eficiencia.

En esta sección se evaluará nuestra estrategia *SDTW* propuesta en este proyecto, comparándola frente a otros posibles métodos y estrategias, con el objeto de cuantificar su precisión, efectividad y eficiencia en el cálculo de la similitud melódica. Para el cálculo de la precisión y efectividad se hará uso de las distintas métricas y herramientas propuestas en la Sección 4.3. con objeto de evaluar gráficamente las relaciones existentes entre los elementos, test de Mantel para cuantificar la bondad de la segmentación del contorno frente al contorno no segmentado y finalmente las métricas *Precision & Recall* y *F-measure* para medir el rendimiento de la clasificación.

Con respecto a la eficiencia, se medirá con el tiempo que tarda la computadora en completar el algoritmo de similitud, para lo cual todas las pruebas han sido efectuadas en la misma computadora con objeto de que la capacidad del procesador no altere la medida de eficiencia.

Los distintos experimentos a realizar se harán en base a los siguientes procedimientos:

- **Procedimiento A:** Algoritmo DTW sobre contorno no segmentado.
- **Procedimiento B:** Algoritmo DTW sobre contorno segmentado usando SmsTools 2.
- **Procedimiento C:** Aproximación del algoritmo DTW de Sakoe-Chiba sobre contorno no segmentado para distintos anchos de banda (r).
- **Procedimiento D:** Algoritmo DTW sobre contorno segmentado usando Fitting para distintos valores de tolerancia (α).



4.4.1 PROCEDIMIENTO A: DTW sobre contorno no segmentado.

- MEDIDAS DE EFICACIA Y PRECISIÓN

A) Análisis filogenético:

Tras realizar los cálculos de matriz de similitud y el algoritmo UPGMA para visualizar el agrupamiento, obtenemos el árbol de la Figura 35, que muestra una perfecta discriminación entre los dos estilos tratados, salvo el caso de la Debla de Chocolate, que de hecho, es razonable su localización porque el cantaor realiza una interpretación alejada del cánón de la Debla.

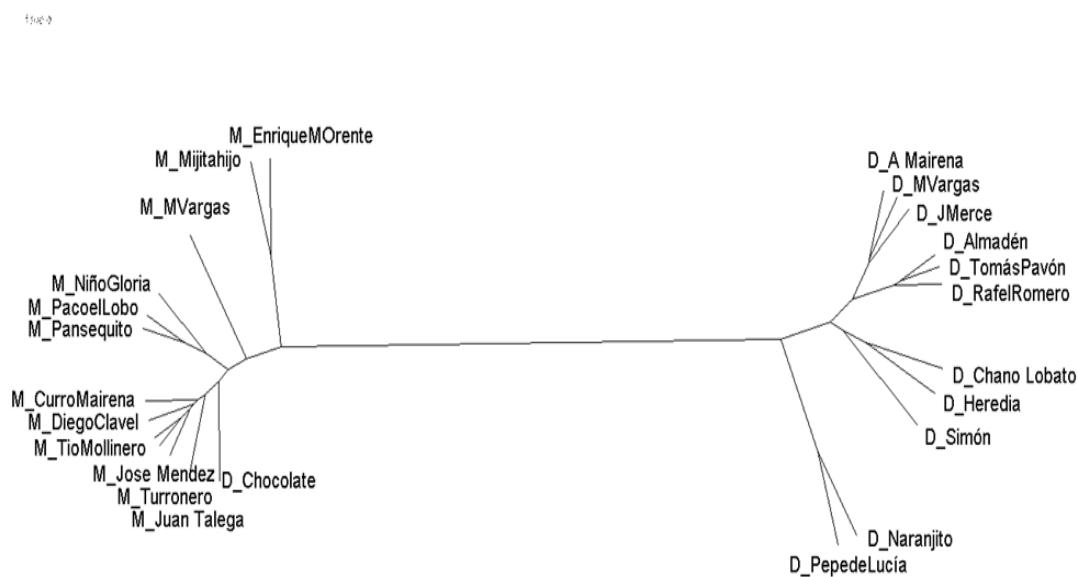


Figura 36: Árbol filogenético Procedimiento A

B) Test de Mantel:

Como se comentó en la Sección 4.2.2, el Test de Mantel Mantel (Mantel & Valand, 1970) se usará para con el fin de evaluar los distintos métodos de segmentación, para así poder analizar la correlación entre las matrices de distancia obtenidas con las distintas segmentaciones y la matriz de distancia obtenida sobre el contorno melódico sin segmentación. Por lo que en esta prueba no tiene sentido aplicar dicha medida de precisión, ya que sería comparar una matriz consigo misma.

C) Métricas de precisión y efectividad:

- Precisión (P):

Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
VP	FP	Precisión	VP	FP	Precisión	VP	FP	Precisión
6	0	1	10	1	0.91	16	1	0.94

Tabla 8

- Efectividad (R):

Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
VP	FN	Efectividad	VP	FN	Efectividad	VP	FN	Efectividad
6	6	0.5	10	2	0.83	16	1	0.67

Tabla 9

- F-measure y Accuracy:

Clase DEBLA		Clase MARTINETE		Clasificación TOTAL	
F-measure	Accuracy	F-measure	Accuracy	F-measure	Accuracy
0.67	0.25	0.87	0.46	0.78	0.35

Tabla 10

Los datos de las tablas 8,9 y 10 reflejan que este método proporciona alto valor de precisión, no tanto de efectividad y resultados análogos ocurren con las medida F-measure y Accuracy. Recordamos que aquí se utiliza el contorno melódico tal como se extrae con el programa SmsTools. A continuación, la tabla 11 de eficiencia muestra que el método requiere mucho tiempo de ejecución, lo que sugiere algún método más eficiente como el método aquí propuesto.

- MEDIDA DE EFICIENCIA:

TIEMPO COMPUTACIONAL		
Segmentación (s)	DTW (s)	TOTAL ESTRATEGIA(s)
NO SEGMENTADO	2.29×10^3	2.29×10^3, aprox. 39 minutos

Tabla 11

4.4.2 PROCEDIMIENTO B: DTW sobre contorno segmentado usando SmsTools 2.

- MEDIDAS DE PRECISIÓN Y EFECTIVIDAD:

A) Análisis filogenético:

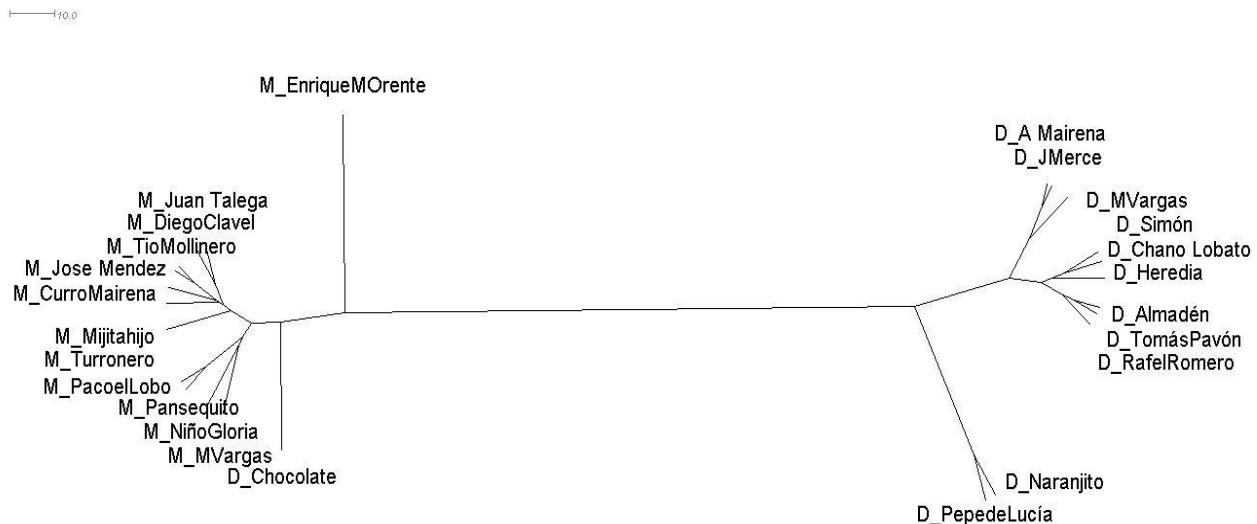


Figura 37: Árbol filogenético Procedimiento B

Ahora observamos en el árbol que la clasificación es similar al procedimiento anterior, Chocolate aparece mal ubicado y resulta destacable la situación de enrique Morente, también con una interpretación que puede decirse “particular”. Este hecho sugiere un estudio musicológico particular.

B) Test de Mantel:

Correlación r_M
f_0 vs Segmentada
0.9050

Tabla 12

C) Métricas de precisión y efectividad:

- Precisión (P):

Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
VP	FP	Precisión	VP	FP	Precisión	VP	FP	Precisión
11	0	1	12	1	0.92	23	1	0.95

Tabla 13

- Efectividad (R):

Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
VP	FN	Efectividad	VP	FN	Efectividad	VP	FN	Efectividad
11	1	0.91	12	0	1	23	1	0.95

Tabla 14

- F-measure y Accuracy:

Clase DEBLA		Clase MARTINETE		Clasificación TOTAL	
F-measure	Accuracy	F-measure	Accuracy	F-measure	Accuracy
0.95	0.45	0.96	0.54	0.95	0.5

Tabla 15

- MEDIDA DE EFICIENCIA:

TIEMPO COMPUTACIONAL		
Segmentación ² (s)	DTW (s)	TOTAL ESTRATEGIA(s)
720	7.78	727.78

Tabla 16

Los resultados mejoran en este procedimiento considerablemente, lo que indica que la transcripción automática de SmsTools es adecuada y mucho más eficiente.

² Al tratarse de una segmentación realizada con un programa externo, el cual usa un algoritmo iterativo, fijamos el tiempo consumido por la segmentación del orden de la duración del audio a segmentar. El corpus Tonás lo componen piezas de 30 segundos de duración como media.

4.4.3 PROCEDIMIENTO C: Aproximación DTW de Sakoe-Chiba sobre contorno no segmentado para distintos anchos de banda(r).

- MEDIDAS DE PRECISIÓN Y EFECTIVIDAD:

A) Análisis filogenético:

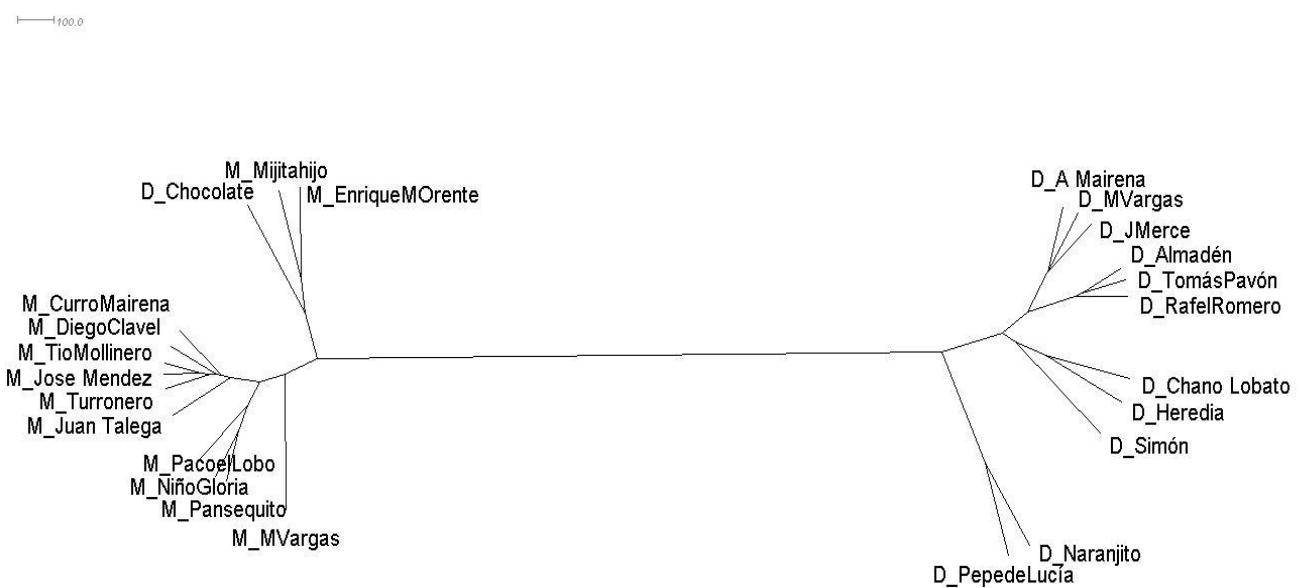


Figura 38: Árbol filogenético Procedimiento C con r=250.

Destacamos el grupo superior izquierdo, con tres elementos, Debla de Chocolate y Martinetes de Mijita y Morente, lo que sugiere un estudio musicológico de tal ubicación. Algo parecido se observa con las deblas de Pepe de Lucía y Naranjito, que si las oímos, poseen más ornamentación que las demás.

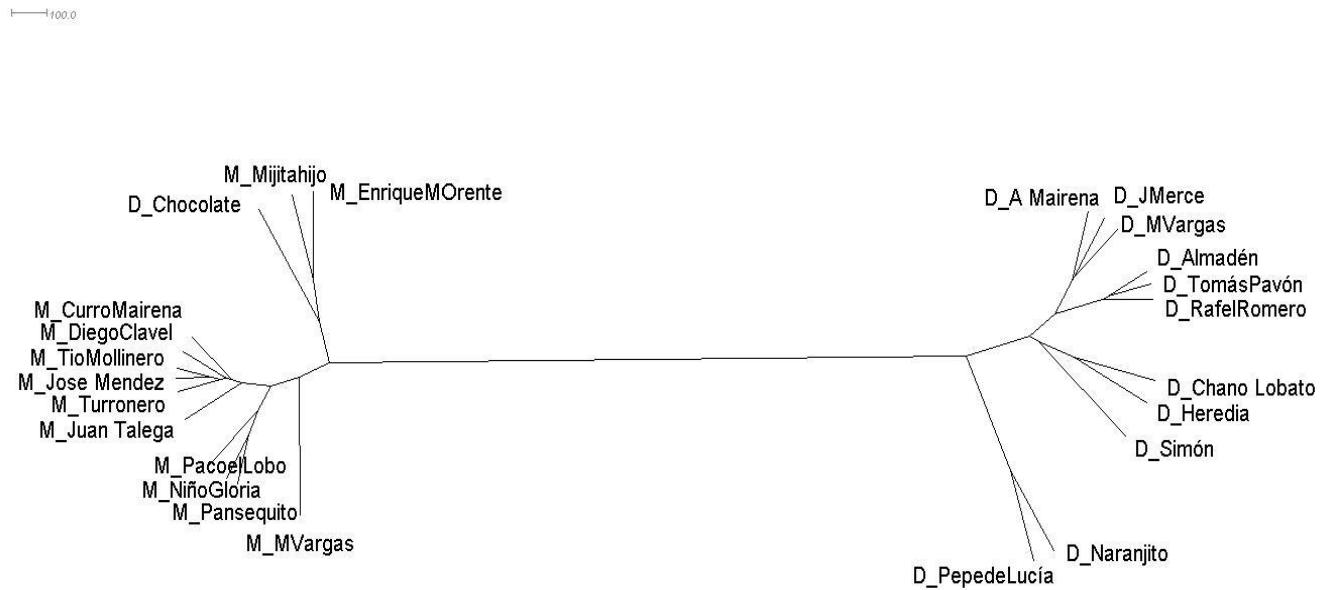


Figura 39: Árbol filogenético Procedimiento C con r=200.

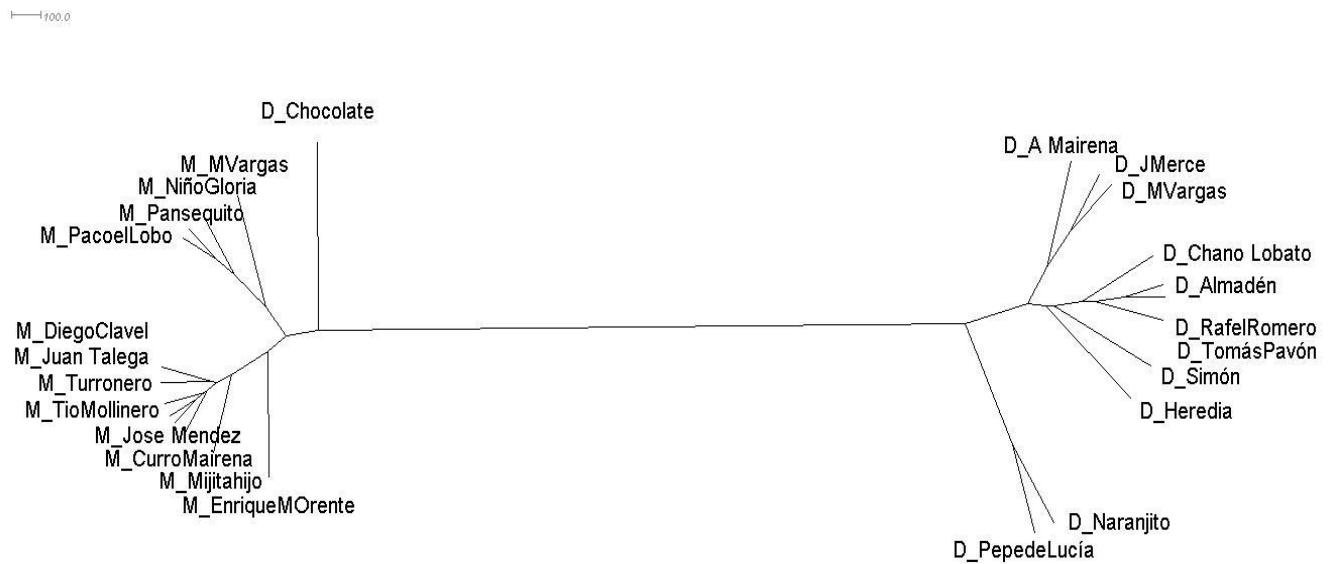


Figura 40: Árbol filogenético Procedimiento C con r=100.

100.0

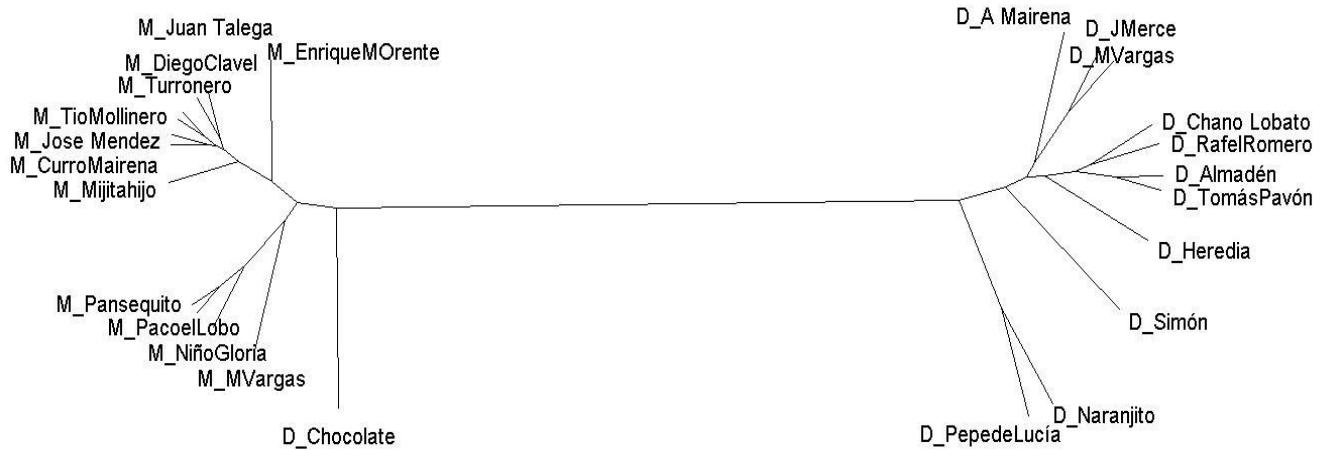


Figura 41: Árbol filogenético Procedimiento C con $r=50$.

100.0

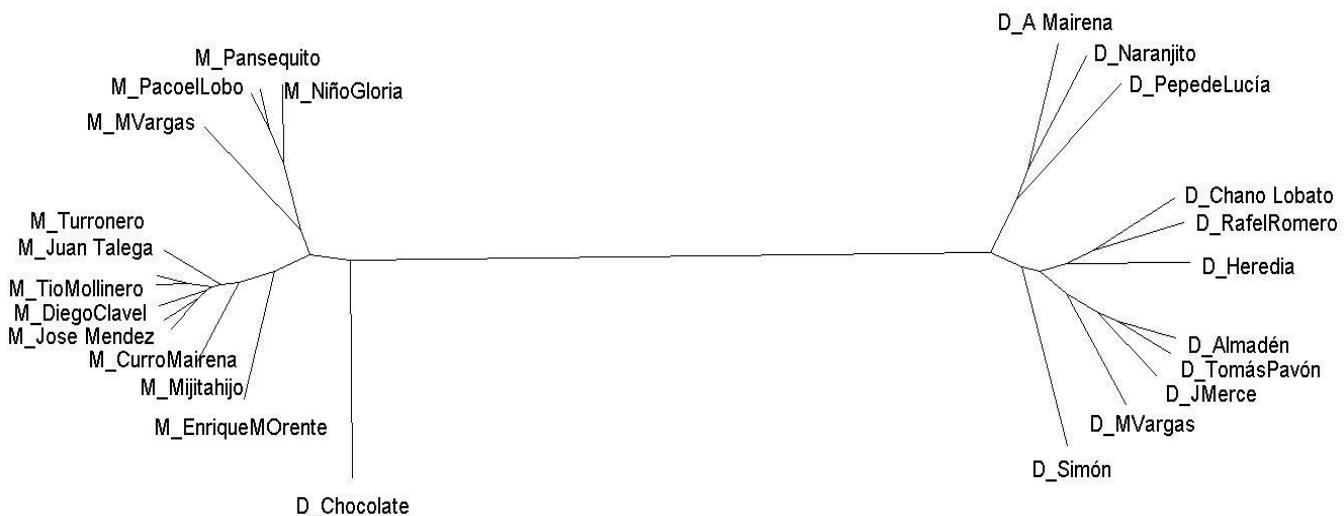


Figura 42: Árbol filogenético Procedimiento C con $r=25$.

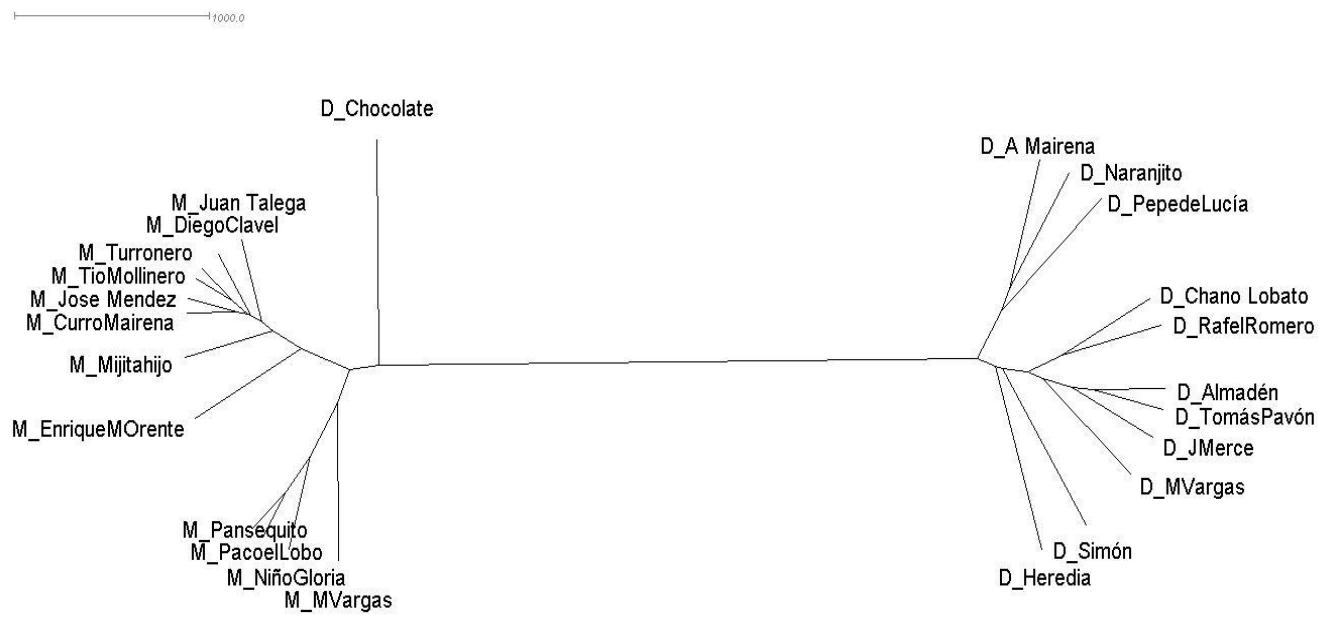


Figura 43: Árbol filogenético Procedimiento C con r=4.

B) Test de Mantel:

Correlación r_M	
r	f_0 vs Aproximación
250	0.9801
200	0.9625
100	0.9345
50	0.9137
25	0.9061
4	0.8323

Tabla 17

C) Métricas de precisión y efectividad:

- Precisión (P):

r	Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
	VP	FP	Precisión	VP	FP	Precisión	VP	FP	Precisión
250	8	0	1	10	1	0.90	18	1	0.95
200	10	0	1	11	1	0.92	21	1	0.96
100	11	0	1	12	1	0.92	23	1	0.95
50	10	0	1	11	1	0.91	21	1	0.96
25	10	0	1	12	1	0.92	22	1	0.94
4	10	0	1	12	1	0.92	22	1	0.94

Tabla 18

- Efectividad (R):

r	Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
	VP	FN	Efectividad	VP	FN	Efectividad	VP	FN	Efectividad
250	8	4	0.67	10	2	0.84	18	6	0.75
200	10	2	0.83	11	1	0.91	21	3	0.87
100	11	1	0.91	12	0	1	23	1	0.95
50	10	2	0.83	11	1	0.91	21	3	0.87
25	10	2	0.83	12	0	1	22	2	0.91
4	10	2	0.83	12	0	1	22	2	0.91

Tabla 19

- F-measure y Accuracy:

r	Clase DEBLA		Clase MARTINETE		Clasificación TOTAL	
	F-measure	Accuracy	F-measure	Accuracy	F-measure	Accuracy
250	0.91	0.44	0.90	0.45	0.91	0.44
200	0.90	0.41	0.91	0.5	0.91	0.45
100	0.95	0.45	0.96	0.54	0.95	0.5
50	0.90	0.41	0.50	0.91	0.91	0.45
25	0.90	0.41	0.96	0.54	0.93	0.47
4	0.90	0.40	0.90	0.48	0.92	0.46

Tabla 20

- MEDIDA DE EFICIENCIA:

r	TIEMPO COMPUTACIONAL		
	Segmentación (s)	DTW (s)	TOTAL ESTRATEGIA(s)
250	NO SEGMENTADO	699.09	699.09
200	NO SEGMENTADO	684.70	684.70
100	NO SEGMENTADO	578.11	578.11
50	NO SEGMENTADO	490.35	490.35
25	NO SEGMENTADO	478.56	478.56
4	NO SEGMENTADO	438.38	438.38

Tabla 21

En general, los resultados de efectividad son aceptables y los de eficiencia son un poco mejores que del procedimiento Procedimiento B. Como se comentó en la Sección 3.4 estos resultados pueden estar motivados por el hecho de que las trayectorias seguidas por los caminos óptimos de deblas y martinetes son muy similares, lo que hace que restricciones de las trayectorias no penalicen excesivamente el cálculo de la similitud.

4.4.4 PROCEDIMIENTO D: Algoritmo SDTW para distintos valores de tolerancia (α).

- MEDIDAS DE PRECISIÓN Y EFECTIVIDAD:

A) Análisis filogenético:

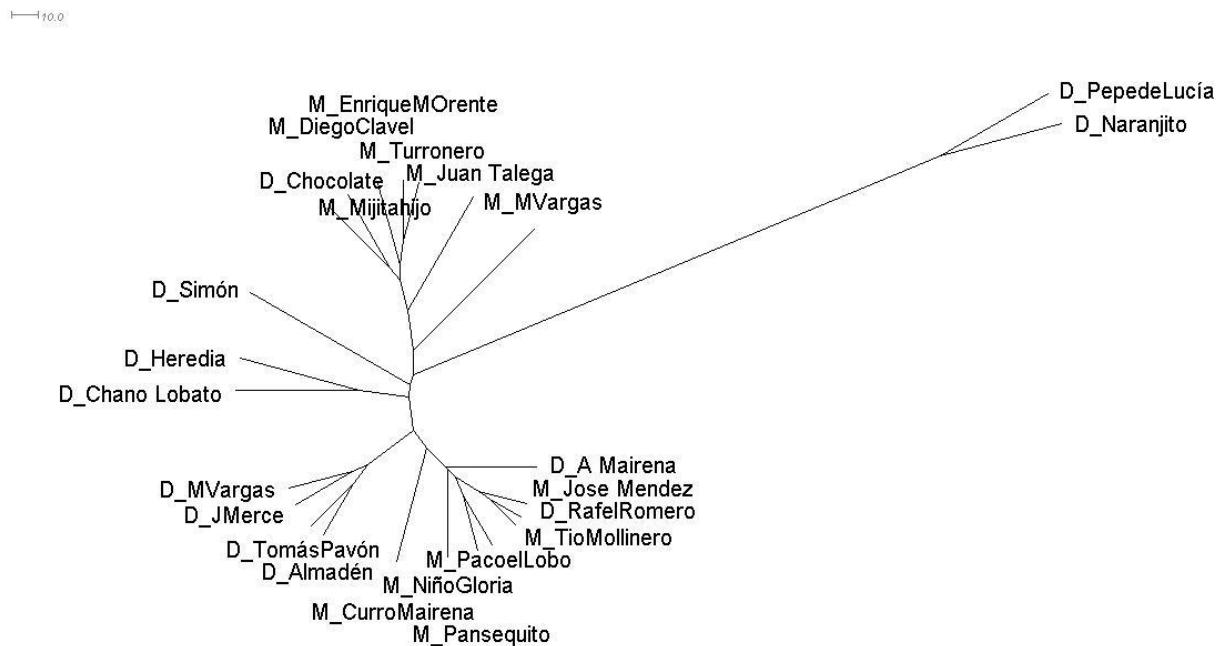


Figura 44: Árbol filogenético Procedimiento D con $\alpha=2$.

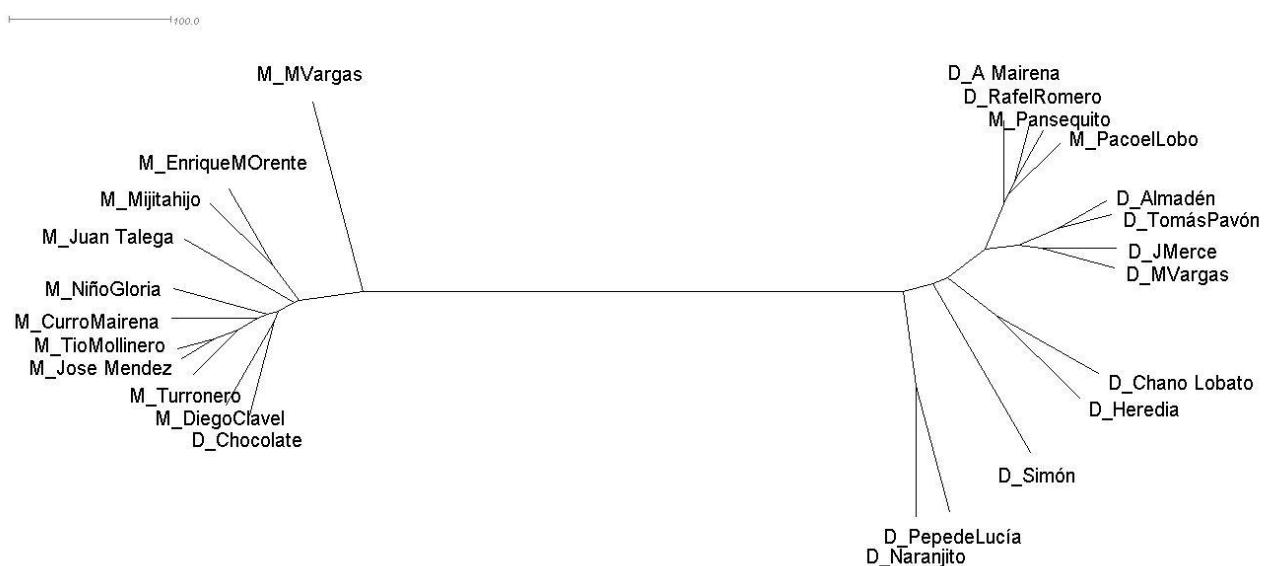


Figura 45: Árbol filogenético Procedimiento D con $\alpha=1$.

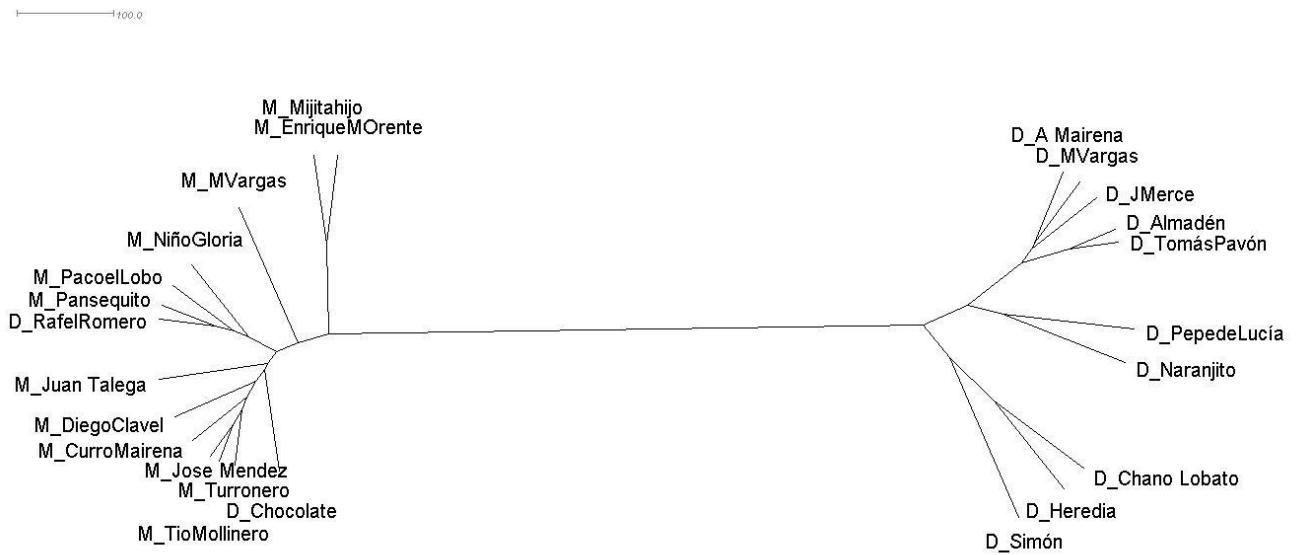


Figura 46: Árbol filogenético Procedimiento D con $\alpha=0,5$.

Para los parámetros de ventana uno y medio tono la clasificación coincide con los casos anteriores, por lo que consideramos aceptables esos valores.

B) Test de Mantel:

Correlación r_M	
α	f_0 vs Aproximación
2	0.8289
1	0.9156
0.5	0.9509

Tabla 22

C) Métricas de precisión y efectividad:

- Precisión (P):

α	Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
	VP	FP	Precisión	VP	FP	Precisión	VP	FP	Precisión
2	6	0	1	2	2	0.5	8	2	0.8
1	7	0	1	6	1	0.87	13	1	0.92
0.5	6	0	1	9	1	0.9	15	1	0.94

Tabla 23

- Efectividad (R):

α	Clase DEBLA			Clase MARTINETE			Clasificación TOTAL		
	VP	FN	Efectividad	VP	FN	Efectividad	VP	FN	Efectividad
2	6	6	0.5	2	10	0.16	8	16	0.33
1	7	5	0.58	6	6	0.5	13	11	0.54
0.5	6	6	1	9	3	0.75	15	9	0.62

Tabla 24

- F-measure y Accuracy:

α	Clase DEBLA		Clase MARTINETE		Clasificación TOTAL	
	F-measure	Accuracy	F-measure	Accuracy	F-measure	Accuracy
2	0.67	0.25	0.25	0.16	0.47	0.20
1	0.74	0.29	0.63	0.29	0.68	0.29
0.5	0.67	0.25	0.81	0.42	0.75	0.33

Tabla 25

- MEDIDA DE EFICIENCIA:

α	TIEMPO COMPUTACIONAL		
	Segmentación (s)	DTW (s)	TOTAL ESTRATEGIA(s)
2	0.12	13.65	13.78
1	0.14	69.25	69.39
0.5	0.23	246.44	246.76

Tabla 26

Destacamos la rapidez de ejecución del caso $\alpha=1$, para el cual los valores de efectividad son aceptables. También es destacable que para $\alpha=0.5$ se divide a la mitad el tiempo de ejecución y se obtiene un 0,33 (tabla 25) como valor de Accuracy frente al caso r=4 del Procedimiento aproximado C (tabla 20). En el capítulo siguiente elaboramos más los resultados y conclusiones que podemos extraer de todos los cálculos anteriores.

Capítulo 5:

CONCLUSIONES

1. Aportaciones del proyecto.
2. Posibles líneas futuras de investigación.

CAPÍTULO 5: CONCLUSIONES

5.1 Aportaciones del proyecto.

La creciente demanda de trabajos relacionados con el uso de herramientas tecnológicas en el ámbito de la música flamenca justifica la aportación de este proyecto, que pretende avanzar sobre algunas cuestiones que surgen a día de hoy en el campo de la investigación del Flamenco, y más concretamente en el ámbito de la similitud melódica y la clasificación de cantes. Por ello, dada la complejidad del tema tratado, este proyecto supone un pequeño paso adicional dentro de la que se ha llamado *Teoría Computacional del Flamenco* (COFLA).

Tras la evaluación de los distintos experimentos realizados en el anterior capítulo se pueden concluir los siguientes aspectos sobre el algoritmo *SDTW* propuesto en este proyecto. Al comparar el algoritmo *DTW* frente al *SDTW* (usándolos como medidas de similitud melódica) se obtiene que:

1. Disminuir el tiempo computacional del orden de 10 veces supone una pérdida de precisión del 5,7% en la posterior clasificación.
2. Variaciones de una unidad del parámetro de tolerancia α suponen variaciones de precisión de entorno el 10%. En cuanto a las variaciones en términos temporales se puede decir que no existe una relación lineal entre el parámetro de tolerancia y el tiempo computacional (ver figura 47).
3. Al comparar la segmentación generada por el programa Smstools 2 frente a la segmentación del algoritmo Fitting, vemos que la primera ofrece un mayor rendimiento global en la clasificación. Esto es debido a que, como se vio en la Sección 3.2, la segmentación realizada por Smstools 2 implementa ciertas características musicológicas (en el proceso de consolidación de notas). Además, es un procedimiento semiautomático, esto es, algunas correcciones del corpus TONAS son manuales. El algoritmo Fitting, por su parte, es automático y realiza una discretización simple del contorno con una precisión prefijada. Por otro lado, al comparar las estrategias en términos de eficiencia, la estrategia que usa el algoritmo Fitting requiere la tercera parte del coste temporal que la estrategia vía segmentación Smstools 2.
4. El uso del contorno melódico extraído sin segmentación no garantiza el mayor rendimiento en la clasificación. De hecho, la clasificación obtenida con la consolidación de notas ofrecida por el Smstools 2 consigue el mayor rendimiento. Esto puede ser debido a que al usar el contorno melódico completo, existan elementos que desvirtúen de algún modo la medida de similitud, mientras que al realizar consolidación de notas se eliminan esas partes de audio irrelevantes para la comparación.

Accuracy vs Eficiencia

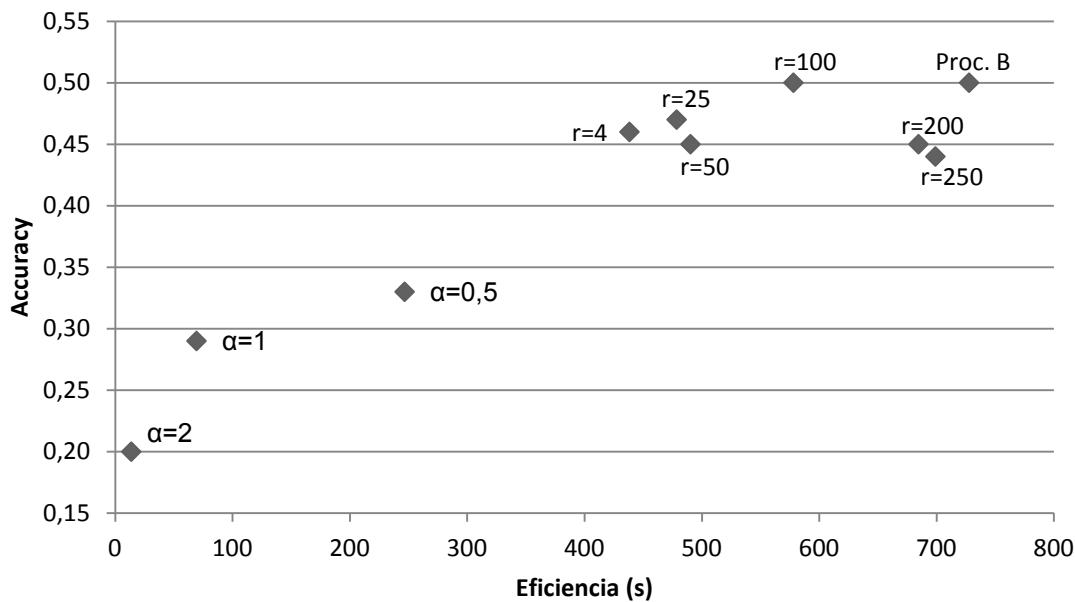


Figura 47: Tabla Accuracy vs eficiencia de los procedimientos B, C y D.

Test de Mantel

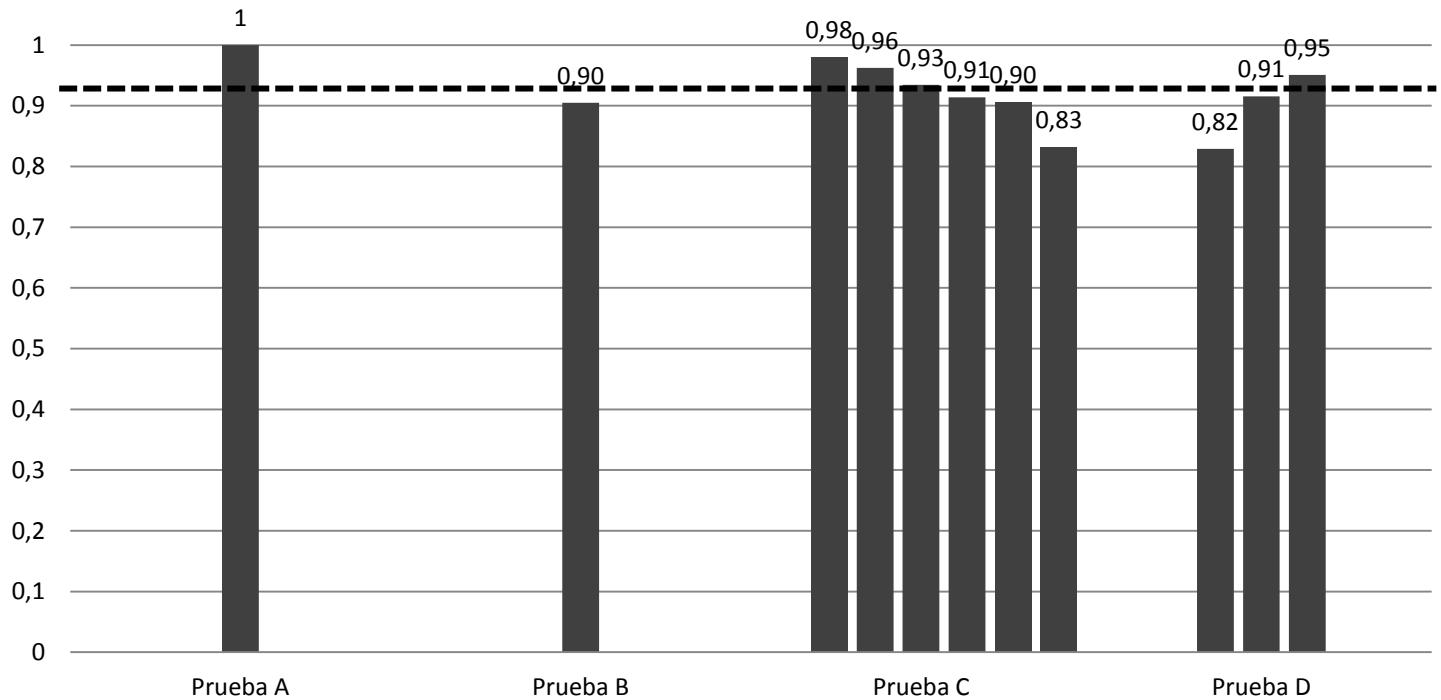


Figura 48: Test de Mantel sobre matriz de distancias.

5. El uso de la correlación ofrecida por el test de Mantel aporta una visión de cómo afecta el hecho de la segmentación del contorno a la matriz de distancias. De nuevo, se vuelve a apreciar la baja correlación entre la matriz de distancias de contorno no segmentado y la matriz de distancias generada tras la segmentación Smstools 2, por lo que se refuerza la hipótesis expuesta en el apartado 4. Véase Figura 48.

5.2 Posibles líneas futuras de investigación.

Al tratarse de un aspecto concreto dentro de un proyecto ambicioso (proyecto COFLA), las líneas futuras de investigación que pueden plantearse son muy variadas. Como ya hemos dicho, el proyecto COFLA está sentando las bases de una nueva área de investigación: *la Teoría computacional de la música flamenca*. Por tanto, podrían enumerarse como líneas naturales de mejora en el ámbito de este proyecto las siguientes:

1. *Mejora en el método de segmentación Fitting*: Una mejora del mismo que permita cierta “consolidación” de los segmentos creados o bien el “autoajuste” del parámetro de tolerancia, pudiéndolo hacer variable a lo largo del contorno a segmentar, serían sin duda clave para una mejora la de precisión y de la eficiencia de la estrategia SDTW propuesta.
2. *Ampliación del corpus de deblas y martinetes*: Si bien la colección de piezas musicales con la que hemos trabajado es representativa de estos cantes, por haber sido seleccionada cuidadosamente entre los intérpretes más característicos, no es demasiado extensa. Un siguiente paso podría ser considerar un corpus de mayores dimensiones y aplicar la misma metodología aquí seguida con objeto de validar su buen funcionamiento.
3. *Análisis de otros estilos*: Otra continuación natural de este proyecto sería aplicar el estudio aquí recogido a otros palos del flamenco. Al introducir cantes con acompañamiento musical, la complejidad de la fase de descripción automática aumentará considerablemente, si bien entendemos que la ampliación de este trabajo a otros palos diferentes a las tonás sería de gran importancia, dándole aún más valor a las aplicaciones mencionadas en el apartado anterior.
4. *Combinación de métodos de aproximación*: La combinación de distintos métodos de aproximación, como pueden ser el algoritmo Fitting y la aproximación Sakoe-Chiba pueden ofrecer resultados prometedores. Esta hibridación pudiera equilibrar los efectos de pérdida de eficiencia y ganancia de precisión al aumentar el parámetro de tolerancia, con el de uso de un

determinado ancho de banda que me permita aumentar la eficiencia sin penalizar la precisión.

5. *El uso de otros aspectos musicales:* Añadir a la melodía otros parámetros como el ritmo, la armonía o el timbre, podrían integrarse también con el objetivo de diseñar una herramienta que considere todos los ámbitos que pueden definir una obra de flamenco.

Somos conscientes del interés académico y social del proyecto, por lo que creemos que necesita una continuación que permita mejorar los resultados y resolver nuevos problemas. De esta forma, podremos avanzar en el conocimiento, divulgación y conservación de este arte con mayúsculas, el Flamenco.

Referencias:

REFERENCIAS

Aichholzer, O., Caraballo, I. E., Díaz-Báñez, J. M., Fabila-Monroy, R., Ochoa, C. & Nigsch, P. (2013). Extremal antipodal polygons and polytopes. *Proc. Mexican Conference on Discrete Mathematics and Computational Geometry, Oaxaca*, 11-15.

Aloupis, G., Fevens, T., Langerman, S., Matsui, T., Mesa, A., Nuñez, Y., & Toussaint, G. (2006). Algorithms for computing geometric measures of melodic similarity. *Computer Music Journal*, 30(3), 67-76.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175-185.

Bartók, B.; & Lord, B. (1951). Serbo-Croatian Folk Songs: Texts and Transcriptions of 75 Folk Songs from the Milman Parry Collection, and a Morphology of Serbo Croatian Folk Melodies. *Columbia University Press*.

Berlanga, M. A. (2001). *Música y religiosidad popular: Saetas y misereres en la Semana Santa andaluza*.

Cabrera, J.J., Díaz-Báñez, J.M., Escobar-Borrego, F.J., Gómez, E., Mora, J. (2008). Comparative Melodic Analysis of A Cappella Flamenco Cantes. *Fourth Conference on Interdisciplinary Musicology (CIM08)*, Thessaloniki, Greece, 2008.

Castro, G. (2010). *Las mudanzas del cante en tiempos de Silverio, análisis histórico musical de su escuela de cante*. Barcelona: Carena.

Castro, G. (2010). Los cantes sin guitarra en el flamenco, antecedentes musicales y modalidades. *Revista de Investigación sobre Flamenco: La madrugá*, (2).

Chu, S., Keogh, E. Hart, & D., Pazzani, M. (2002). Iterative deepening dynamic time warping for time series. In *Proceedings of SIAM International Conference on Data Mining*.

Cormen, T., Leiserson. C., Rivest, R. (1990). Introduction to Algorithms. MIT Press.

Cronin, C. (1998). Concepts of melodic similarity in music copyright infringement suits. *Computing in musicology: a directory of research*, (11), 187-209.

Díaz-Bañez, J. M., Farigu, G., Toussaint, G., Gómez, F., & Rappaport, D. (2005). Similaridad y evolución en la rítmica del flamenco: una incursión de la matemática computacional. *Gaceta de la Real Sociedad Matemática Española*, 8(2), 489-509.

Díaz-Báñez, J. M., & Mesa, J. A. (2001). Fitting rectilinear polygonal curves to a set of points in the plane. *European Journal of Operational Research*, 130(1), 214-222.

Díaz-Banez, J. M., Farigu, G., Gómez, F., Rappaport, D., & Toussaint, G. T. (2004). El compás flamenco: a phylogenetic analysis. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, 61-70.

Díaz-Báñez, J. M. (2013). On math problems in the study of flamenco singing (in Spanish). *La Gaceta de la Real Sociedad Matemática Española* (Vo.I 13), 513-541.

Díaz-Báñez, J. M., Korman, M., Pérez-Lantero, P., Pilz, A., Seara, C., & Silveira, R. I. (2013). New results on stabbing segments with a polygon. *Algorithms and Complexity, Lecture Notes in Computer Science* (Vol. 7878), 146-157.

Gómez, E., & Bonada, J. (2008). Automatic melodic transcription of flamenco singing. In *Proceedings of the fourth Conference on Interdisciplinary Musicology, Thessaloniki, Greece*.

Gómez, E., & Bonada, J. (2013). Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37(2), 73-90.

Gómez, E., & López, C. (2013). Criterios para la transcripción manual de la colección de tonás. Barcelona.

Gómez, E., Bonada J., & Salamon J. (2012). Automatic Transcription of Flamenco Singing from Monophonic and Polyphonic Music Recordings. *III Interdisciplinary Conference on Flamenco Research (INFLA) and II International Workshop of Folk Music Analysis (FMA)*.

Gómez, F., Pikrakis, A., Mora, J., Díaz-Báñez, J. M., Gómez, E., & Escobar, F. (2011). Automatic detection of ornamentation in flamenco music. In *4th International Workshop on Music and Machine Learning: Learning from Musical Structure, Neural Information Processing Systems Foundation, Granada*.

Gómez, F., Pikrakis A., Mora J., Diaz-Báñez J.M., Gómez E., Escobar F. (2012). Automatic Detection of Melodic Patterns in Flamenco Singing by Analyzing Polyphonic Music Recordings. *III Interdisciplinary Conference on Flamenco Research (INFLA) and II International Workshop of Folk Music Analysis (FMA)*.

Guo A., Siegelmann H. (2004). Time-warped longest common subsequence algorithm for music retrieval. In *Proceedings of the Fifth International Conference on Music Information Retrieval (ISMIR 2004, 10–15 October 2004, Barcelona)*.

Hahn, U., Chater, N., & Richardson, L. B. (2003). Similarity as transformation. *Cognition*, 87(1), 1-32.

Halmos, I. (1978). Computational Ethnomusicology in Hungary in 1978. Ann Arbor, MI: MPublishing, University of Michigan Library.

Hewlett, W. B., & Selfridge-Field, E. (1998). Melodic similarity: Concepts, procedures, and applications (Vol. 11). *The MIT Press*.

Holzapfel, A.,(2010). Similarity methods for computational ethnomusicology (*Doctoral dissertation, PhD thesis, University of Crete*).

Huson, D. H. (1998). SplitsTree: analyzing and visualizing evolutionary data. *Bioinformatics*, 14(1), 68-73.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1), 67-72.

Klapuri, A., & Davy, M. (2006). *Signal processing methods for music transcription*. Springer.

Kramer, C., & Plenckers, L. J. (1998). The structure of the saeta flamenca: An analytical study of its music. *Yearbook for traditional music*, 30, 102-132.

Kroher, N. (2013). Automatic Characterization of Flamenco Singing by Analyzing Audio Recordings. *Master thesis, Master Program in Sound and Music Computing, Universitat Pompeu Fabra*.

Lefranc, P., & Bonald, J.M.C. (2000). *El cante jondo.: Del territorio a los repertorios: tonás, siguiriyas, soleares* (Vol. 16). Universidad de Sevilla.

Lemström, K. (2000). String Matching Techniques for Music Retrieval. *PhD thesis, University of Helsinki, Department of Computer Science. Report A-2000-4*.

Lerdahl, F. A., & Jackendoff, R. S. (1983). *A generative theory of tonal music*. The MIT Press.

Machado & Álvarez, A. (1881). Colección de cantes flamencos recogidos y anotados por A. Sevilla: *El Porvenir* (reedited Madrid: Cultura Hispánica, 1975).

Mantel, N., & Valand, R.S. (1970). A technique of nonparametric multivariate analysis. *Biometrics*, 547-558.

Molina, R. & Mairena, A. (1963). Mundo y formas del Cante Flamenco». *Revista de Occidente*.

Mongeau, M., & Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, 24(3), 161-175.

Mora, J., Gómez, F., Gómez, E., Escobar-Borrego, F. J., & Díaz-Báñez, J. M. (2010). Characterization and melodic similarity of a Cappella flamenco cantes. International Society for Music Information Retrieval Conference, ISMIR.

Myers, C., Rabiner, L., & Rosenberg, A. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(6), 623-635.

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Springer 1st Edition.

Orio, N. (2006). Music retrieval: a tutorial and review (Vol. 1). *Fundations and Trends in Information Retrieval*, 1-90.

Pikrakis, A., Theodoridis, S., & Kamarotos, D. (2003). Recognition of isolated musical patterns using context dependent dynamic time warping. *Speech and Audio Processing, IEEE Transactions on*, 11(3), 175-183.

Powers, D. M. W. (2011). Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.

Rosenberg, M. S., & Anderson, C. D. (2011). PASSaGE: pattern analysis, spatial statistics and geographic exegesis. Version 2. *Methods in Ecology and Evolution*, 2(3), 229-232.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1), 43-49.

Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5), 561-580.

Seeger, S. (1966). Versions and Variants of the Tunes of “Barbara Allen”. *Selected Reports in Ethnomusicology*, 1(1).

Smith, T. F., & Waterman, M. S. (1981). Comparison of biosequences. *Advances in Applied Mathematics*, 2(4), 482-489.

Toussaint, G. T. (2010). Computational geometric aspects of musical rhythm. In *14th Annual Fall Workshop on Computational Geometry*, 47-48.

Tzanetakis, G., Kapur, A., Schloss, W. A., & Wright, M. (2007). Computational ethnomusicology. *Journal of interdisciplinary music studies*, 1(2), 1-24.

van Rijsbergen, C. J. (1974). Foundation of evaluation. *Journal of Documentation*, 30(4), 365-373.

Vega, J. B., & Ruiz, M. R. (1990). Diccionario enciclopédico ilustrado del flamenco. Ed. Cinterco.

Anexo:

CÓDIGOS MATLAB

ANEXO: CÓDIGOS MATLAB

```

%%% ARCHIVO: Dtw_Sakoe_GC.m
%% Tiene como entradas los dos contornos melódicos a comparar y como
%% salida el coste global o global cost de la comparación.
function[globalcost]=Dtw_Sakoe_GC(ref,test)
I=length(ref);
J=length(test);
for i=1:I
    for j=1:J
        %Distancia euclídea
        node_cost(i,j)=sqrt(sum((ref(:,i)-test(:,j)).^2));
    end
end
%Inicialización
D(1,1)=node_cost(1,1);
pred(1,1)=0;
for i=2:I
    D(i,1)=D(i-1,1)+node_cost(i,1);
    pred(i,1)=i-1+sqrt(-1)*1;
end
for j=2:J
    D(1,j)=D(1,j-1)+node_cost(1,j);
    pred(1,j)=1+sqrt(-1)*(j-1);
end
%Bucle principal
for i=2:I
    for j=2:J
        [D(i,j),ind]=min([D(i-1,j-1),D(i-1,j),D(i,j-1)]+node_cost(i,j));
        if(ind==1)
            pred(i,j)=(i-1)+sqrt(-1)*(j-1);
        elseif(ind==2)
            pred(i,j)=(i-1)+sqrt(-1)*(j);
        else
            pred(i,j)=(i)+sqrt(-1)*(j-1);
        end
    end%forj
end%fori

matching_cost=D(I,J);
% Algoritmo del cálculo del camino óptimo o best path
n=J;
m=I;
k=1;
w=[I J];
colormax=w;
while ((n+m) ~= 2)
    if (n-1)==0
        m=m-1;
    elseif (m-1)==0
        n=n-1;
    else
        [values,number]=min([D(m-1,n-1),D(m-1,n),D(m,n-1)]);
        switch number
        case 1
            m=m-1;
            n=n-1;
        case 2
        end
    end
end

```

```

m=m-1;
case 3
n=n-1;
end
end
k=k+1;
w=[m n; w];
end
gc=0;
globalcost=0;
f=w(:,1);
c=w(:,2);
tam=length(w); % cálculo del número de componentes del camino óptimo
for i=1:tam
gc=gc+D(f(i),c(i));
end
globalcost=gc/tam;

```

%%% ARCHIVO: dtw_banda.m

%% Tiene como entradas los dos contornos melódicos a comparar y el valor del ancho de banda r, como salida el coste global o global cost de la comparación.

```
function [globalcost,g,w,cont] = dtw_banda(A,B,r)
```

```

[L, I] = size(A);
[M, J] = size(B);

d = zeros(I, J);
for i = 1:I
    for j = 1:J
        d(i,j)=sqrt(sum((A(:,i)-B(:,j)).^2));
    end
end

% matriz de distancias
g = zeros(I,J);
g(:,:)= inf;
s = J/I;
steps = zeros(I,J);
g(1,1)=d(1,1);
for i = 2:I;
    for j = 2:J;
        if (abs(i-(j/s)) > r)
            % estamos fuera de la ventana
            continue;
        end
        [g(i,j)]=min([g(i-1,j-1),g(i-1,j),g(i,j-1)]+d(i,j));
    end
end

matching_cost=g(I,J);
% algoritmo de camino óptimo
n=J;
```

```

m=I;
k=1;
w=[I J];
colormax=w;
while ((n+m) ~=2)
    if (n-1)==0
        m=m-1;
    elseif (m-1)==0
        n=n-1;
    else
        [values,number]=min([g(m-1,n-1),g(m-1,n),g(m,n-1)]);
        switch number
        case 1
            m=m-1;
            n=n-1;
        case 2
            m=m-1;
        case 3
            n=n-1;
        end
    end
    k=k+1;
    w=[m n; w];
end

gc=0;
globalcost=0;
f=w(:,1);
c=w(:,2);
tam=length(w);
for i=1:tam
    gc=gc+g(f(i),c(i));
end
globalcost=gc/tam;

```

%%% ARCHIVO: fitting.m

```

%% Tiene como entrada el contorno melódico a segmentar y valor de la
%% tolerancia al error alfa.Como salida tiene el vector de semitonos
%% (vnote) el vector de notas (not), el vector de notas MIDI (not_MIDI),
%% y la duración y el tiempo de inicio de cada nota.
function[vnote,not(not_MIDI,duracion,inicio)=fitting(A,time,alfa)
%algoritmo de ventana
% A: Vector de semitonos del contorno melódico
% time: Vector de tiempo del contorno melódico
% alfa: Valor de la tolerancia al error
tam=length(A);
vnote=A;
not_MIDI=[];
not=[];
duracion=[];
inicio=[]; %instante de inicio de la nota en not
% uppers=[]; %vector donde almaceno el punto superior de la nota
% lowers=[]; %vector donde almaceno el punto inferior que define la
nota
% epsilon=[];
% epp=[];
i=1;
ip=1;
sup=[];

```

```

inf=[];
supfinal=[];
inffinal=[];
%contador general
%contador de rastreo superior
%contador de rastreo inferior
if A(ip+1)>=A(ip) %subida
    i=ip;
    il=ip;
    iu=ip+1;
    ip=ip+1;
else %bajada
    iu=ip;
    il=ip+1;
    ip=ip+1;
end

while i<tam
    if ip<tam
        dif1=abs(A(iu)-A(ip));
        dif2=abs(A(il)-A(ip));
        maxep=max(abs(A(iu)-A(ip)),abs(A(il)-A(ip)));
        if maxep <= alfa
            if A(ip)>=A(iu)
                %display('primer')
                iu=ip;
                ip=ip+1;

                elseif A(ip)<=A(il)
                %display('esleif')
                il=ip;
                ip=ip+1;
            else
                ip=ip+1;
            end
        else
            %%crear nota%
            sup=[sup A(iu)];
            inf=[inf A(il)];
            %display('crear nota intermedia')

[i,iu,il,ip,vnote,note_MIDI,note,dur,ini]=createventana(A,i,iu,il,ip,vnote,time);
            inicio=[inicio ini];
            duracion=[duracion dur];
            not_MIDI=[not_MIDI note_MIDI]; % notas midi
            not=[not note]; %semitones

            end
        else
            %%crear nota final%%
            supfinal=[supfinal A(iu)];
            inffinal=[inffinal A(il)];
            %display('crear nota final')

[i,iu,il,ip,vnote,note_MIDI,note,dur]=createventana(A,i,iu,il,ip,vnote,
            not_MIDI=[not_MIDI note_MIDI];
            not=[not note];
            duracion=[duracion dur];

```

```

        break
    end
end

```

%%% ARCHIVO: createventana.m

```

%% Crea el segmento y la nota sobre todo el contorno que ha entrado en
la amplitud del rango de error alfa.
function
[i,iu,il,ip,vnote,note_MIDI,note,dur,ini]=createventana(A,i,iu,il,ip,v
note,time)

% display('crear nota')
note=((A(iu)-A(il))/2) + A(il);
note_MIDI=69 + ceil(note);
dur=time(ip)-time(i);
ini=time(i);
for j=i:ip
    vnote(j)=note;
end
if ip<length(vnote)
    if A(ip+1)>=A(ip) %subida
        i=ip;
        il=ip;
        iu=ip+1;
        ip=ip+1;
    else %bajada
        i=ip;
        iu=ip;
        il=ip+1;
        ip=ip+1;
    end
else
    vnote(ip)=note;
    ip=ip+1;
    i=i+1;
end

```

%%% ARCHIVO: f0cleaner.m

```

%% Programa de pretratamiento de los vectores de f0 previo a la
comparación, elimina los ceros correspondientes a los silencios
folderName=uigetdir('.', 'Select CORPUS folder'); % Selección de la
carpeta que contiene los .txt de los f0
carpeta=dir(folderName);
tam=length(carpeta);

for t=3:tam
[time_test,energy,test,freq]=textread([folderName '\\
carpeta(t).name], '%f%f%f%f');
maximo=max(test);
size=length(test);
ceroprin=0;
cerofin=0;
i=1;
while test(i)==0 %cero del principio
    ceroprin=ceroprin+1;
    i=i+1;
end

```

```

j=size;
while test(j)==0 %cero del final
    cerofin=cerofin+1;
    j=j-1;
end
v=0;
v(1,1)=[12*log2(test(ceroprin+1)/440)];
for i=ceroprin+2:size-cerofin
    if test(i)==0
        v=[v 0];
    else
        v=[v 12*log2(test(i)/440)];
    end
end

fileID = fopen(carpetat.name, 'w');
fprintf(fileID, '%f\n', v);
fclose(fileID);
end

%% ARCHIVO: f0asemitonos.m
function
[vsemi,time,size]=f0asemitonos(folderName_origen,carpeta_origen,t)

%Devuelve el vector de semitonos de cada archivo .txt, el vector de
%tiempo
%y el número de archivos convertidos.

[time_test,x,test,y]=textread([folderName_origen '\'
carpeta_origen(t).name], '%f%f%f%f');
maximo=max(test);
size=length(test);

vsemi=[];
time=[];
i=1;
j=1;
while i<size
    if test(i)==0
        i=i+1;
    else
        vsemi(j)=12*log2(test(i)/440);
        time(j)=time_test(i);
        i=i+1;
        j=j+1;
    end
end

vsemi=vsemi';
time=time';

```

```

%%% ARCHIVO: noteasemitonos.m
%% Convierte el formato de notas aportado por la consolidación de
%% Smstools 2 a semitonos.
function [semitono_note]=noteasemitonos(folderName_noteasemi,t)

carpeta=dir(folderName_noteasemi);
tam=length(carpeta);

[ini,dur(note,energy)=textread([folderName_noteasemi '\
carpeta(t).name], '%f%f%f%f','delimiter','','');

semi=0;
for i=1:length(note)
    if note(i)<=0
        semi(i)=0;
    else
        semi(i)=note(i)-69;
    end
end

semitono_note=[ini dur semi'];

```



```

%%% ARCHIVO: knn.m
%% Implementación del método k-nn, donde la entrada es la matriz de
%% distancias, el número de vecinos a tener en cuenta y el número de
%% elementos de la clase.
function [PRECISION,RECALL,ACCURACY,FM]=knn(matriz,k,n)

% m: matriz de costos del DTW
% k: nº de vecinos que exige el k-nn
% n: nº de cantes por estilo
m=length(matriz);
class=zeros(m,k);
for (i=1:m)
    v=matriz(:,i);
    [B,P]=sort(v);
    for (j=2:(k+1)) % siempre me pone igual a cero a ella
        if(P(j)<=n)
            class(i,(j-1))=1; % Almacena un 1 si es debla % Almacena
un 0 si es martinete
        end
    end
end
%%%%% CÁLCULO DE PRESICION AND RECALL %%%%%%
[PRECISION,RECALL,ACCURACY,FM]=PR(class,k,n);

```

```

%%% ARCHIVO: PR.m
%% Implementación de las distintas métricas de clasificación
function [PRECISION,RECALL,ACCURACY,FM]=PR(class,k,n)
m=length(class);
T=sum(class,2);
VPD=0;FND=0;
VND=0;FPD=0;
for(i=1:m) %% Determinación de la predicción del clasificador.
    if (T(i)==k && i<=n)
        VPD=VPD+1;
    end
    if (T(i)~=k && i<=n)
        FND=FND+1;
    end
    if (T(i)~=k && i>n)
        VND=VND+1;
    end
    if (T(i)==k && i>n)
        FPD=FPD+1;
    end
end
PRESICION_DEBLA=(VPD/ (VPD+FPD)) ;
RECALL_DEBLA=(VPD/ (VPD+FND)) ;
ACCURACY_D=((VPD+FPD) / (VPD+VND+FND+FPD)) ;
FM_D=(2*PRESICION_DEBLA*
RECALL_DEBLA) / (PRESICION_DEBLA+RECALL_DEBLA) ;

VPM=0;FNM=0;
VNM=0;FPM=0;
for(i=1:m)
    if (T(i)==0 && i>n)
        VPM=VPM+1;
    end
    if (T(i)~=0 && i>n)
        FNM=FNM+1;
    end
    if (T(i)~=0 && i<=n)
        VNM=VNM+1;
    end
    if (T(i)==0 && i<=n)
        FPM=FPM+1;
    end
end
PRESICION_MARTINETE=(VPM/ (VPM+FPM)) ;
RECALL_MARTINETE=(VPM/ (VPM+FNM)) ;
ACCURACY_M=((VPM+FPM) / (VPM+VNM+FNM+FPM)) ;
FM_M=(2*PRESICION_MARTINETE*
RECALL_MARTINETE) / (PRESICION_MARTINETE+RECALL_MARTINETE) ;

% RENDIMIENTOS DE LA CLASIFICACIÓN TOTAL

VP=VPM+VPD;
FN=FNM+FND;
VN=VNM+VND;
FP=FPM+FPD;
PRECISION=(VP/ (VP+FP)) ;
RECALL=(VP/ (VP+FN)) ;
ACCURACY=((VP+FP) / (VP+VN+FN+FP)) ;
FM=(2*PRECISION* RECALL) / (PRECISION+RECALL) ;

```