

# Sistema de detección de instrumentos musicales en señales de audio

Autor: Marc Juan Monguillot  
Tutora: Verónica Vilaplana Besler  
Profesor: David García Solórzano

Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació  
Aplicacions multimèdia basades en processament del senyal

Junio de 2020

# Créditos/Copyright



Esta obra esta sujeta a la Licencia Reconocimiento-NoComercial-CompartirIgual

[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	Sistema de detección de instrumentos musicales en señales de audio
<b>Nombre del autor:</b>	Marc Juan Monguillot
<b>Nombre del colaborador/a docente:</b>	Verónica Vilaplana Besler
<b>Nombre del PRA:</b>	David García Solórzano
<b>Fecha de entrega (mm/aaaa):</b>	06/2020
<b>Titulación o programa:</b>	Grau en Enginyeria de Tecnologies i Serveis de Telecomunicació
<b>Área del Trabajo Final:</b>	Aplicacions multimèdia basades en processament del senyal
<b>Idioma del trabajo:</b>	Español
<b>Palabras clave</b>	Machine learning, Procesamiento digital de audio, Instrumentos musicales

### Resumen del Trabajo (máximo 250 palabras):

En este trabajo se ha diseñado un sistema para la detección de instrumentos musicales desarrollado con el requisito innovador de tener cierta capacidad para la identificación simultánea de diversas categorías superpuestas en una misma señal, sin restricción alguna respecto a las posibles combinaciones. El enfoque empleado se basa en que el contenido frecuencial de las notas musicales se limita al entorno de la frecuencia fundamental y sus armónicos, lo que permite bajo ciertas condiciones poder separar de forma aproximada el espectro de las distintas notas musicales superpuestas en un extracto de señal.

Con este objetivo, se propone un conjunto de características calculadas a partir del espectro separado de cada nota, de forma que se puedan extraer varios vectores de un mismo segmento de señal. También se ha desarrollado un algoritmo que permite la identificación de las distintas notas superpuestas mediante la detección precisa de sus frecuencias fundamentales. Para la clasificación de los vectores se ha optado por una máquina de vectores de soporte (SVM) con Kernel cúbico tras realizar varias pruebas de comparación y optimización con distintas técnicas de machine learning. Finalmente, se ha desarrollado una App de MATLAB a modo de prototipo de pruebas del sistema. Este modelo ha sido entrenado para 26 instrumentos distintos obteniendo unas tasas aproximadas de precisión del 90% para señales con un único instrumento y del 60% en señales conteniendo de 2 a 4 instrumentos.

### Abstract (in English, 250 words or less):

This project presents the design of a musical instrument detection system developed with the innovative requirement of having some ability to distinguish between different categories mixed simultaneously in the same signal, without restrictions regarding possible combinations. The approach followed is based on the fact that the spectrum of musical notes is constrained to the regions near the fundamental frequency and its harmonics. This allows, under certain conditions, to separate the approximate spectrum of different notes mixed in a signal excerpt.

For this purpose, a set of features calculated from the separated spectrum of each note is proposed so that several vectors can be extracted from the same signal excerpt. An algorithm has also been developed to identify the different overlapping notes by precisely detecting their fundamental frequencies. To classify the vectors, a support vector machine (SVM) with cubic Kernel has been chosen after carrying out various comparison and optimization tests with different machine learning techniques.

Finally, a MATLAB App has been developed as a prototype to test the system. This model has been trained for 26 different instruments, obtaining approximate accuracy rates of 90% for signals with a single instrument and around 60% for signals containing between 2 and 4 instruments.

*A mis padres.*

«L'étude approfondie de la nature est la source  
la plus féconde des découvertes mathématiques.»  
JEAN-BAPTISTE JOSEPH FOURIER

Quisiera agradecer a todo el equipo de gestión, profesores y compañeros de los estudios de Informática, Multimedia y Telecomunicación de la UOC por su inestimable ayuda durante todos estos años y, en especial, a mi tutora Verónica Vilaplana por presentarme el mundo del *machine learning* y guiar mis primeros pasos a través de él.

También quiero agradecer el apoyo y los ánimos que me ha brindado mi familia durante toda mi carrera formativa y profesional.

Y, por último, gracias a Rodrigo y a mis amigos por los buenos momentos y la compañía durante todos estos años y los que vienen.

Finalmente, quisiera agradecer a los centenares de músicos que han compartido sus interpretaciones musicales en la plataforma online YouTube y que, de esta forma, han creado una gran biblioteca pública virtual de grabaciones de sonidos con los que se ha podido entrenar y probar el sistema presentado en este trabajo. Este trabajo hubiera sido mucho más complicado de realizar sin ellos.

# Abstract

This project presents the design of a musical instrument detection system developed with the innovative requirement of having some ability to distinguish between different categories mixed simultaneously in the same signal, without restrictions regarding possible combinations. The approach followed is based on the fact that the spectrum of musical notes is constrained to the regions near the fundamental frequency and its harmonics. This allows, under certain conditions, to separate the approximate spectrum of different notes mixed in a signal excerpt.

For this purpose, a set of features calculated from the separated spectrum of each note is proposed so that several vectors can be extracted from the same signal excerpt. An algorithm has also been developed to identify the different overlapping notes by precisely detecting their fundamental frequencies. To classify the vectors, a support vector machine (SVM) with cubic Kernel has been chosen after carrying out various comparison and optimization tests with different machine learning techniques.

Finally, a MATLAB App has been developed as a prototype to test the system. This model has been trained for 26 different instruments, obtaining approximate accuracy rates of 90% for signals with a single instrument and around 60% for signals containing between 2 and 4 instruments.

## Keywords

Machine learning, Digital audio processing, Musical Instruments, Pattern Recognition

# Resumen

En este trabajo se ha diseñado un sistema para la detección de instrumentos musicales desarrollado con el requisito innovador de tener cierta capacidad para la identificación simultánea de diversas categorías superpuestas en una misma señal, sin restricción alguna respecto a las posibles combinaciones. El enfoque empleado se basa en que el contenido frecuencial de las notas musicales se limita al entorno de la frecuencia fundamental y sus armónicos, lo que permite bajo ciertas condiciones poder separar de forma aproximada el espectro de las distintas notas musicales superpuestas en un extracto de señal.

Con este objetivo, se propone un conjunto de características calculadas a partir del espectro separado de cada nota, de forma que se puedan extraer varios vectores de un mismo segmento de señal. También se ha desarrollado un algoritmo que permite la identificación de las distintas notas superpuestas mediante la detección precisa de sus frecuencias fundamentales. Para la clasificación de los vectores se ha optado por una máquina de vectores de soporte (SVM) con Kernel cúbico tras realizar varias pruebas de comparación y optimización con distintas técnicas de machine learning.

Finalmente, se ha desarrollado una App de MATLAB a modo de prototipo de pruebas del sistema. Este modelo ha sido entrenado para 26 instrumentos distintos obteniendo unas tasas aproximadas de precisión del 90% para señales con un único instrumento y del 60% en señales conteniendo de 2 a 4 instrumentos.

## Palabras clave

Machine learning, Procesamiento digital de audio, Instrumentos musicales, Reconocimiento de patrones

# Índice

<b>1. Introducción.....</b>	<b>10</b>
1.1. Introducción .....	10
1.2. Descripción .....	10
1.3. Objetivos generales .....	11
1.3.1. Objetivos principales.....	11
1.3.2. Objetivos secundarios .....	12
1.4. Metodología y proceso de trabajo.....	12
1.5. Planificación.....	16
1.6. Presupuesto .....	17
1.7. Estructura del resto del documento .....	18
<b>2. Análisis de mercado .....</b>	<b>19</b>
2.1. Público objetivo.....	19
2.2. Estado del arte (Marco teórico) .....	19
2.2.1. Situación general.....	19
2.2.2. Análisis y extracción de características .....	20
2.2.3. Sistemas de clasificación .....	22
2.2.4. Resultados y propuestas innovadoras .....	24
<b>3. Propuesta.....</b>	<b>26</b>
3.1. Objetivos del sistema .....	26
3.2. Especificaciones del prototipo .....	27
<b>4. Diseño.....</b>	<b>28</b>
4.1. Arquitectura general del sistema .....	28
4.2. Preprocesado de la señal .....	29
4.3. Conceptos de teoría musical .....	31
4.4. Segmentación y inventariado .....	33
4.5. Detección de notas y obtención de su espectro específico .....	37
4.6. Cálculo de los vectores de características .....	43
4.7. Clasificación.....	49
<b>5. Implementación.....</b>	<b>54</b>
5.1. Programación de una MATLAB App con interfaz gráfica para pruebas .....	54

---

<b>5.2. Requisitos de instalación .....</b>	<b>54</b>
<b>5.3. Instrucciones de instalación.....</b>	<b>54</b>
<b>6. Demostración y resultados .....</b>	<b>55</b>
<b>6.1. Instrucciones y ejemplo de uso del sistema.....</b>	<b>55</b>
<b>6.2. Modo de análisis rápido .....</b>	<b>56</b>
<b>6.3. Medidas de desempeño del sistema.....</b>	<b>57</b>
6.3.1. Tasas de aciertos de categorías en señales con un instrumento.....	57
6.3.2. Tasas de aciertos de categorías en señales con varios instrumentos .....	58
6.3.3. Matriz de confusión del sistema por vectores extraídos .....	58
6.3.4. Tasas de sensibilidad y especificidad por vectores extraídos.....	59
6.3.5. Respuesta del sistema a categorías desconocidas.....	59
<b>7. Conclusiones y líneas de futuro .....</b>	<b>60</b>
7.1. Conclusiones .....	60
7.2. Líneas de futuro.....	61
<b>Bibliografía.....</b>	<b>62</b>
<b>Anexos.....</b>	<b>64</b>
<b>Anexo A: Corpus de audios de entrenamiento del sistema.....</b>	<b>64</b>
<b>Anexo B: Corpus de audios de test y respuestas proporcionadas por el sistema ...</b>	<b>67</b>
<b>Anexo C: Código MATLAB de extracción de vectores de características.....</b>	<b>72</b>
<b>Anexo D: Código MATLAB del prototipo e interfaz gráfica .....</b>	<b>86</b>

# Figuras y tablas

## Índice de figuras

Figura 1. Variabilidad de los armónicos emitidos en un extrado de audio de violín .....	14
Figura 2: Diagrama de Gantt de la planificación del TFG .....	16
Figura 3. Esquema habitual de los detectores de instrumentos musicales propuestos en la bibliografía .....	19
Figura 4. Esquema básico de funcionamiento del sistema planteado.....	26
Figura 5. Diagrama de flujo del sistema diseñado .....	28
Figura 6. Filtros de preprocesado probados en el desarrollo del sistema .....	29
Figura 7. Relación entre amplitud RMS de una señal y el número de vectores extraídos .....	31
Figura 8. Espectro teórico de una nota de 440 Hz .....	32
Figura 9. Comparación de sistemas de enventanado para segmentos cortos de 20 ms .....	34
Figura 10. Comparación de sistemas de enventanado para segmentos largos de 50 ms .....	35
Figura 11. Relación temporal entre ventanas largas y cortas .....	36
Figura 12. Propagación del error al estimar la frecuencia fundamental con métodos clásicos .....	37
Figura 13. FFT de un segmento con dos notas y ruido gaussiano.....	39
Figura 14. Picos significativos detectados en la FFT del segmento.....	39
Figura 15. Amplitudes de los armónicos detectadas en el segmento .....	42
Figura 16. Espectros específicos obtenidos a partir de un segmento con dos notas simultáneas.....	43
Figura 17. Método de cálculo de la importancia de las características .....	46
Figura 18. Archivo ejecutable del instalador de la MATLAB App con que se ha implementado el sistema .....	54
Figura 19. Icono de inicio de la MATLAB APP desarrollada una vez instalada .....	54
Figura 20. Interfaz gráfica de la APP desarrollada.....	55
Figura 21. Botón e interfaz de selección de archivos .wav .....	55
Figura 22. Barras de progreso durante el análisis de un archivo .....	56
Figura 23. Interfaz gráfica mostrando los resultados de la clasificación de un archivo .....	56
Figura 24. Selector del número de instrumentos a detectar.....	56
Figura 25. Selector de tipo de análisis .....	57
Figura 26. Cuadro de diálogo de cambio de tipo de análisis.....	57

## Índice de tablas

Tabla 1. Presupuesto estimado de desarrollo del sistema .....	17
Tabla 2. Instrumentos incluidos en el prototipo con sus códigos de identificación.....	27
Tabla 3. Comparación rendimiento de los distintos filtros de preprocesado .....	30
Tabla 4. Características de los dos procesos de segmentación utilizados .....	36
Tabla 5. Primer listado de posibles frecuencias fundamentales en segmento de ejemplo (15 primeras filas) .....	40
Tabla 6. Segundo listado de posibles frecuencias fundamentales en segmento de ejemplo (15 primeras filas) ..	41
Tabla 7. Tercer listado de posibles frecuencias fundamentales en segmento de ejemplo.....	41
Tabla 8. Listado final de frecuencias fundamentales detectadas en segmento de ejemplo.....	41
Tabla 9. Componentes del vector de características preliminar y su importancia.....	46
Tabla 10. Componentes del vector de características definitivo y su importancia .....	49
Tabla 11. División del corpus de entrenamiento en 5 grupos de <i>cross-validation</i> .....	50
Tabla 12. Tasas de acierto de los algoritmos <i>machine learning</i> (primera selección) .....	50
Tabla 13. Tasas de acierto de los algoritmos <i>machine learning</i> (segunda selección).....	51
Tabla 14. Tasas de aciertos con SVM Cuadrático optimizado .....	52
Tabla 15. Tasas de aciertos con SVM Cúbico optimizado .....	52
Tabla 16. Tasas de aciertos con Red neuronal optimizada .....	53
Tabla 17. Comparación de los tres sistemas <i>machine learning</i> optimizados .....	53
Tabla 18. Tasas de aciertos en instrumentos detectados para señales con un único instrumento.....	57
Tabla 19. Tasas de aciertos en instrumentos detectados para señales con varios instrumentos simultáneos .....	58
Tabla 20. Matriz de confusión en vectores (valores en %).....	58
Tabla 21. Tasas de sensibilidad y especificidad en vectores extraídos .....	59
Tabla 22. Respuesta del sistema a algunas fuentes desconocidas .....	59
Tabla 23. Corpus de audios de entrenamiento del sistema .....	64
Tabla 24. Corpus de audios de test y respuestas proporcionadas por el sistema .....	67

# 1. Introducción

## 1.1. Introducción

El oído humano tiene la capacidad de reconocer y diferenciar, con mayor o menor exactitud, la fuente emisora de muchos sonidos. Un ejemplo cotidiano se encuentra en el reconocimiento de voz, ya que al identificarse un familiar por teléfono o interfonía, es habitual que utilice simplemente la oración “Soy yo”, dando por sentado implícitamente la capacidad que el ser humano tiene de identificar el timbre de voces familiares. Por otro lado, cualquier persona con unos conocimientos mínimos de música, sabe distinguir perfectamente entre la música interpretada por un violinista de la de un saxofonista.

No obstante, en aquellas situaciones en las que el individuo desconoce o no se ha familiarizado con las fuentes, no es posible la identificación fisiológica de las mismas. La capacidad de diferenciación también se resiente en el caso de piezas en las que intervengan muchos instrumentos de forma simultánea o, entre instrumentos cercanos dentro de una misma familia instrumental, como es el caso del violín y la viola<sup>1</sup>. En estos casos, solo músicos o melómanos con experiencia pueden tener la capacidad de detectarlos. No obstante, ni siquiera los oídos entrenados pueden reconocer todos los instrumentos con la misma facilidad. En una investigación<sup>2</sup> realizada en 2014, se determinó que los timbres del piano y la guitarra eran los mejor identificados entre alumnos de estudios superiores de música en conservatorios de la Comunidad de Madrid, mientras que, por el contrario, el fagot fue el instrumento con más dificultad en su reconocimiento.

Con la proliferación de la domótica y teléfonos inteligentes durante la última década, ha surgido una gran variedad de aplicaciones en las que se utilizan técnicas de procesamiento de sonido para realizar diferentes tareas de reconocimiento de patrones. Por ejemplo, mediante la aplicación Shazam es posible detectar una canción que esté sonando en nuestro entorno. Además, los algoritmos de reconocimiento del habla se han mejorado de tal forma que hablar con un altavoz inteligente está empezando a consolidarse como una rutina cotidiana.

No obstante, el reconocimiento de instrumentos musicales no cuenta con el mismo grado de interés comercial<sup>3</sup>. De hecho, en las plataformas de distribución de aplicaciones para los sistemas operativos más utilizados (Android y iOS) no se ha encontrado ninguna APP que identifique un instrumento musical a partir de una señal de sonido.

## 1.2. Descripción

Este Trabajo de Final de Grado (TFG) tiene como finalidad desarrollar un sistema de detección de instrumentos musicales en señales de audio a partir de la investigación de las características más

---

<sup>1</sup> (Maersch, Rohde, Seiffert, & Singer, 1994) y (González, 2017)

<sup>2</sup> (Lago Castro, Ponce de León Barranco, & Balo, 2014)

<sup>3</sup> (Eronen & Klapuri, Musical instrument recognition using cepstral coefficients and temporal features, 2000)

útiles para este cometido, así como el análisis y comparación de los diferentes algoritmos de clasificación. Aunque no existen aplicaciones de este tipo en el mercado, el objetivo de este trabajo no es la programación de un *software* comercial sino el de analizar, desarrollar y probar un modelo que pueda establecer la base teórica para desarrollar aplicaciones de este tipo en el futuro. Aún así, este TFG incluye la programación de una sencilla interfaz gráfica ejecutable dentro del entorno MATLAB, a modo de prototipo, para poder probar el sistema desarrollado y comprobar su utilidad. Este prototipo está excluido de cualquier finalidad comercial ya que su sistema clasificador se ha entrenado a partir de señales de sonido extraídas a partir de vídeos disponibles en la plataforma online YouTube de los que no se dispone de licencia de uso comercial.

A pesar de no ser una necesidad habitual para el público general; el desarrollo de un sistema de identificación de instrumentos no solo tiene la utilidad didáctica de dar a conocer la fuente de los sonidos, sino que puede constituir un punto de partida para el posible desarrollo de herramientas profesionales para la conversión de audios multi-instrumentales en partituras. De la misma forma, sus resultados pueden ser interesantes para el desarrollo de sistemas de reconocimiento de otros tipos de fuentes de sonido, así como para complementar los sistemas biométricos de reconocimiento del hablante existentes.

Aunque no existen aplicativos en el mercado, sí se han publicado durante las últimas dos décadas varios artículos de investigación que abordan el problema desde diferentes estrategias y cuyos funcionamientos y resultados se resumen en el apartado 2.2 (Estado del arte). Estos artículos constituyen el punto de partida de este TFG, que se complementará con la inclusión de nuevas características propuestas por el autor, así como la experimentación con el conjunto de algoritmos de clasificación proporcionados por el entorno MATLAB.

Además, el diseño del sistema de selección y extracción de características de la señal que se propone en este TFG se ha desarrollado desde un primer momento con el requisito de tener la capacidad de detectar instrumentos que suenen de forma simultánea sin restricciones ni incompatibilidades respecto a los posibles grupos de instrumentos ni a sus tésituras<sup>4</sup>, lo cual representa un aspecto novedoso en relación a los artículos de investigación publicados.

## **1.3. Objetivos generales**

### **1.3.1. Objetivos principales**

Los objetivos principales de este TFG son:

- Diseñar un sistema de detección de instrumentos musicales en señales de audio mediante un clasificador basado en técnicas de *machine learning*.

---

<sup>4</sup> Rango de frecuencias que es capaz de emitir un instrumento musical

- Idear y diseñar una arquitectura del sistema de extracción de características con la capacidad de detectar varios instrumentos sonando simultáneamente en la misma señal.
- Analizar y medir la utilidad de las diferentes características propuestas en los artículos de investigación, así como proponer otros valores para incluir en los vectores.
- Comparar el rendimiento de los diferentes algoritmos de clasificación, así como escoger y optimizar el sistema de *machine learning* que mejor generalice las categorías a clasificar.
- Medir el rendimiento del sistema mediante la obtención de la matriz de confusión y tasas de precisión y especificidad en la clasificación de los vectores de características extraídos de los archivos de test que contienen un único instrumento.
- Medir el rendimiento del sistema mediante la obtención de las tasas de acierto en la clasificación de archivos de test conteniendo uno o varios instrumentos.
- Diseñar una aplicación ejecutable en el entorno MATLAB con una interfaz gráfica sencilla que permita a cualquier usuario probar el sistema con sus propios archivos de sonido.
- Introducirse a las diferentes técnicas de *machine learning* y explorar las posibilidades que el entorno MATLAB dispone para el entrenamiento y utilización de dichos sistemas.

### 1.3.2. Objetivos secundarios

Los objetivos secundarios de este TFG son:

- Programar en MATLAB las diferentes funciones de procesamiento de señales, extracción de características y clasificación de los vectores y publicar el código generado.
- Comprobar el comportamiento del sistema al introducir una señal de audio procedente de un instrumento o fuente desconocido.
- Analizar las dificultades encontradas y los puntos débiles del sistema y proponer posibles soluciones para su mejora.

## 1.4. Metodología y proceso de trabajo

Para desarrollar un sistema de estas características es imprescindible disponer de un corpus de archivos de sonido con un único instrumento, de los que poder extraer un conjunto de vectores de características perfectamente identificados para entrenar los distintos sistemas y medir su rendimiento. Aunque la primera idea fue utilizar preferentemente archivos de sonido procedentes de bases de datos públicas, debido a la ausencia de recursos abiertos de estas características se ha recurrido finalmente a extraer el sonido de vídeos subidos a la plataforma online YouTube, procurando que la calidad de su sonido sea aceptable, sin presencia de ruido ni reverberaciones evidentes. En el Anexo A se ha publicado la lista de los videos consultados para la extracción del corpus de entrenamiento. En este corpus se han rechazado cualquier audio procedente de sintetizadores ya que probablemente presenten unas características distintivas mucho más simples y fáciles de reconocer que la de un instrumento real.

La primera parte del trabajo consiste en el diseño del sistema de extracción de los vectores de características, así como la determinación de la información contenida en los mismos. Esta fase empieza con la toma de decisiones en cuanto a los procesos de preprocesado, segmentación y enventanado de las señales a analizar. Posteriormente, se procede a la definición de un vector de características provisional para realizar las primeras pruebas de clasificación. Estas características serán sometidas a un análisis individual de su eficacia en los que se determinará cuales tienen una mayor influencia para clasificar correctamente los instrumentos y cuales son superfluas. A partir de estos datos, se diseñará el vector de características definitivo eliminando aquella información que no sea útil para el clasificador y añadiendo nuevas características que, a raíz del análisis anterior, se prevea que puedan contener nueva información relevante.

En el desarrollo de esta primera parte tiene un papel muy importante el requisito novedoso de poder distinguir instrumentos sonando simultáneamente y, dado que cada vector de características solo se puede clasificar en un único instrumento, este sistema tendrá la particularidad de que podrá extraer varios vectores de características de un mismo segmento de señal.

Aunque la separación de distintas fuentes mezcladas en una señal de audio es, en general, una tarea muy complicada, la estrategia aplicada en este TFG se basa en que los instrumentos seleccionados emiten fundamentalmente notas con un tono perceptible; y por ello, su contenido espectral se limita al entorno de la frecuencia fundamental y su tren de armónicos. Así pues, el hecho de que cada nota musical solo ocupe unas regiones delimitadas en el espectro hace posible que, si se detectan las distintas frecuencias fundamentales simultáneas, entonces se pueda obtener un vector de características para cada una de ellas asociando al mismo únicamente la información contenida en el espacio ocupado en el espectro por la nota e ignorando el resto del dominio frecuencial.

Evidentemente, se pueden dar situaciones de solapamiento de armónicos de distintas notas que tendrán que ser solventadas mediante un clasificador con suficiente capacidad de generalización como para no clasificar erróneamente los vectores con algunos armónicos distorsionados. También pueden darse situaciones puntuales de instrumentos distintos interpretando simultáneamente la misma nota, de manera que el contenido frecuencial de todo el tren de armónicos no correspondería genuinamente a una sola categoría, pero estos casos se deben considerar puntuales y confiar el éxito de la clasificación en el correcto funcionamiento para los vectores extraídos cuando las notas no coinciden. Al fin y al cabo, si dos instrumentos tocan continua y simultáneamente la misma melodía, a efectos perceptivos se trataría de un timbre distinto al que proporciona cada instrumento por separado.

La segunda parte del sistema consiste en encontrar el mejor algoritmo de clasificación para el sistema diseñado. Aunque la teoría musical más elemental indica que la distinción entre notas de distintos instrumentos reside en las diferencias en la intensidad de los diferentes armónicos, si se analiza la evolución de éstos en el tiempo se observa que lejos de mantener un comportamiento constante, estos presentan una gran variabilidad con pequeños cambios en la frecuencia fundamental y la amplitud de

la señal. Para ejemplarizar este comportamiento, en la Figura 1 se muestra un extracto de 5 segundos de una señal de audio de violín, con la frecuencia fundamental emitida y la evolución de los 6 primeros armónicos.

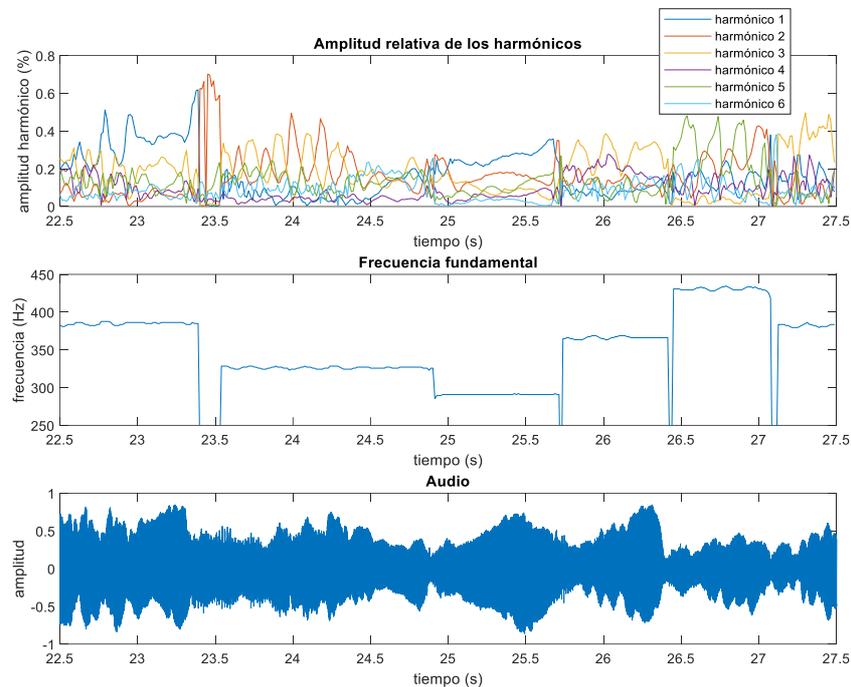


Figura 1. Variabilidad de los armónicos emitidos en un extracto de audio de violín

Debido a este comportamiento tan inestable de los armónicos y, tal como proponen la totalidad de artículos de investigación en esta área, es necesario recurrir a un sistema de *machine learning* para poder clasificar los vectores extraídos. Para probar los distintos algoritmos en MATLAB se utilizará la aplicación *Neural Network Pattern Recognition* y las funciones `patternnet` y `train` para el caso de redes neuronales y la aplicación *Classification Learner* para otros 24 algoritmos disponibles en dicho entorno y que incluye la mayoría de los utilizados en la bibliografía existente, como k-vecinos más cercanos (kNN), máquinas de vectores de soporte (SVM) o *Random Forest*.

Dada la gran variedad de algoritmos disponibles es necesario hacer inicialmente algunos procesos de descarte para poder focalizar los esfuerzos en los sistemas que mejores resultados ofrezcan. Esta medida es necesaria ya que en el sistema diseñado se extraen miles de vectores de características para cada archivo, lo que ralentiza enormemente el entrenamiento y prueba de muchos algoritmos con un PC doméstico, pudiendo alargarse algunas pruebas concretas durante varios días si no se limita fuertemente el número de vector utilizados. De hecho, los tiempos de computación han constituido el principal problema en el desarrollo de este proyecto obligando a hacer la totalidad de los procesos de entrenamiento de clasificadores, así como de optimización de hiperparámetros mediante una pequeña selección estratificada de vectores de características.

Finalmente, una vez identificado el sistema óptimo de clasificación se procede a realizar el entrenamiento definitivo y a efectuar las pruebas finales para medir su rendimiento. Dado que el

sistema clasifica los archivos utilizados en el entrenamiento con una tasa de aciertos mucho más alta que en audios totalmente desconocidos, es necesario proveerse de un nuevo corpus de archivos, distinto al de entrenamiento, para poder medir el rendimiento del sistema en las condiciones reales de utilización en las que los archivos a clasificar no han formado parte del desarrollo del mismo. En el Anexo B se ha publicado la lista de los videos consultados para la extracción de este corpus de test, conjuntamente con la respuesta facilitada por el sistema para cada una de las señales.

La programación completa del sistema, así como el entrenamiento del clasificador y la interfaz gráfica ejecutable se realizarán en el entorno MATLAB. Todo el código de las funciones generadas en MATLAB, y que se referencian a lo largo del trabajo, se adjunta debidamente comentado, en los Anexos C y D.

Por último, a efectos experimentales, se analizará la respuesta del sistema para varias señales conteniendo sonidos de instrumentos no incluidos en el clasificador o de otras fuentes de origen natural y se comprobará si el clasificador es capaz de ofrecer una respuesta cuyo timbre perceptivo se parezca al de los sonidos introducidos.

## 1.5. Planificación

La planificación de este TFG se resume en el siguiente diagrama de Gantt:

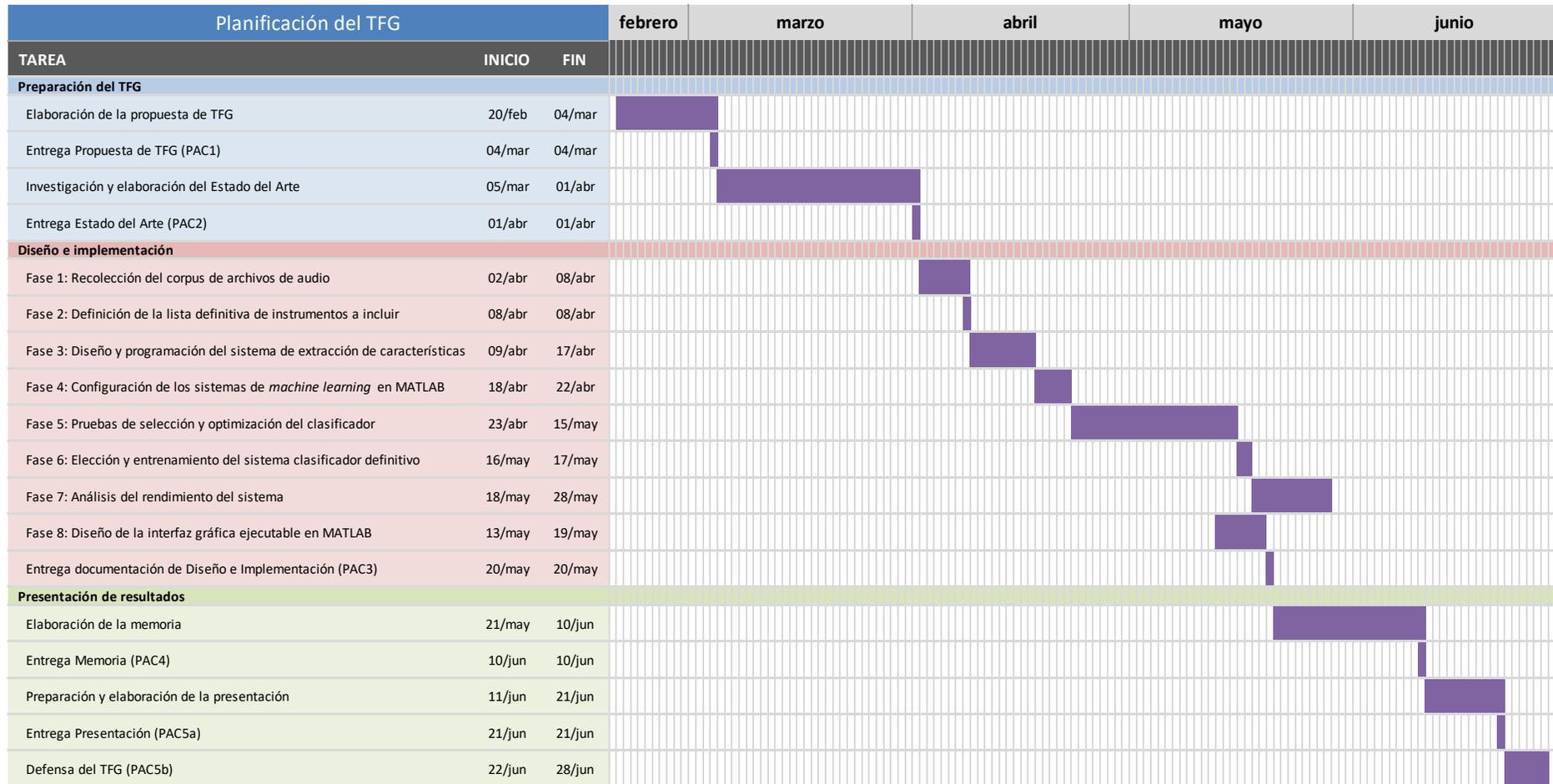


Figura 2: Diagrama de Gantt de la planificación del TFG

## 1.6. Presupuesto

A continuación, se expone un presupuesto aproximado para la realización de este proyecto con precios a fecha de 31 de mayo de 2020. La utilización de un servidor de *cloud computing* no es obligatoria, pero permite acortar de forma importante los tiempos de computación utilizados para extraer los vectores del corpus y entrenar los sistemas de clasificación durante las pruebas, además de ser útil para obtener un sistema con mejores prestaciones al permitir la realización de los entrenamientos con un número de vectores muy superior al que permite un PC doméstico en tiempos razonables.

CANTIDAD	DESCRIPCIÓN ELEMENTO	PRECIO POR UNIDAD	TOTAL ELEMENTO
200	Hora ingeniero técnico de telecomunicaciones	30,00 €	6.000,00 €
1	Licencia anual MATLAB uso industrial	800,00 €	800,00 €
1	Licencia anual <i>Statistics and Machine Learning Toolbox</i> de MATLAB uso industrial	400,00 €	400,00 €
1	Licencia anual <i>Signal Processing Toolbox</i> de MATLAB uso industrial	400,00 €	400,00 €
100	Hora servicio <i>cluster</i> online para <i>cloud computing</i> Amazon EC2 con 16 CPU virtuales con SO Windows	1,35 €	135,00 €
<b>SUBTOTAL</b>			7.735,00 €
<b>IVA (21%)</b>			1.624,35 €
<b>TOTAL</b>			<b>9.359,35 €</b>

Tabla 1. Presupuesto estimado de desarrollo del sistema

No obstante, como se indica en la presentación de este TFG, el prototipo del sistema desarrollado tiene el objetivo de validar el diseño propuesto, pero no puede utilizarse con fines comerciales debido a que el clasificador se ha entrenado con videos publicados en la plataforma YouTube de los que se carece de licencia de uso comercial.

En caso de que se desee desarrollar una aplicación de uso comercial, será necesario adquirir los derechos de suficientes grabaciones de alta calidad de los instrumentos para entrenar el clasificador. Si no se pudiera conseguir estos recursos, será necesario proceder a la grabación de señales de entrenamiento *ad hoc*, lo que implicaría la contratación de músicos y alquiler de un estudio o equipos de grabación. Finalmente, la programación de un *software* de calidad que cumpla los estándares del mercado implicará recurrir a servicios de ingeniería informática.

## 1.7. Estructura del resto del documento

En el capítulo 2 (**Análisis de mercado**) se explica brevemente la utilidad del sistema propuesto y la ausencia de aplicaciones disponibles en el mercado. A continuación, se expone el Estado del arte sobre la detección de instrumentos musicales mediante la explicación de las diferentes técnicas propuestas en distintos artículos de investigación publicados.

En el capítulo 3 (**Propuesta**) se explican las características básicas, alcance y limitaciones del sistema que se ha desarrollado en este trabajo. Asimismo, se presentan las 26 categorías que se han seleccionado para el entrenamiento y uso del prototipo con el que se validará la eficacia de la propuesta.

En el capítulo 4 (**Diseño**) se explica de forma detallada la arquitectura del sistema propuesto en todas sus fases, los problemas encontrados y decisiones de diseño tomadas, así como las diferentes pruebas de distintos componentes del sistema, como es el caso del filtro de énfasis, la composición del vector de características, la selección del algoritmo de *machine learning* empleado como clasificador y su optimización.

En el capítulo 5 (**Implementación**) se introduce la aplicación de MATLAB desarrollada como prototipo de pruebas del sistema y se indican los requisitos e instrucciones de instalación.

En el capítulo 6 (**Demostración y resultados**) se explica el funcionamiento de la aplicación diseñada con un ejemplo real. Posteriormente, se presentan las medidas de desempeño del prototipo que se ha desarrollado a partir del entrenamiento con las señales indicadas en el Anexo A y la prueba con las listadas en el Anexo B, detallándose las tasas de error en instrumentos detectados, así como la matriz de confusión, tasas de sensibilidad y tasas de especificidad de los vectores de características extraídos.

Finalmente, en el capítulo 7 (**Conclusiones y líneas de futuro**) se valoran los resultados mostrados en el capítulo 6 y se explican las lecciones aprendidas. Asimismo, se plantean diversas líneas de actuación que podría ser útil explorar para mejorar el rendimiento del sistema.

## 2. Análisis de mercado

### 2.1. Público objetivo

Dado que la finalidad principal del diseño de este sistema es fundamentalmente académica, este trabajo está orientado para profesionales y estudiantes de los campos relacionados con el procesamiento de audio e inteligencia artificial, así como cualquier persona interesada en estas áreas.

Cualquier aplicación de uso comercial que pudiera desarrollarse a partir de este trabajo tendría como público objetivo cualquier persona que necesitara por motivos de ocio o profesionales identificar los instrumentos que suenan en un audio, sin ningún tipo de limitación demográfica, económica o psicográfica<sup>5</sup>.

### 2.2. Estado del arte (Marco teórico)

#### 2.2.1. Situación general

Como se ha comentado, no existe actualmente en el mercado<sup>6</sup> ninguna aplicación de reconocimiento de instrumentos musicales. Desde el punto de vista de la investigación, todavía no existe un consenso en relación a las características a extraer y a los sistemas de clasificación para identificar con mejor eficiencia la fuente del sonido, por lo que en los siguientes apartados se realizará un repaso a las diferentes técnicas propuestas. La gran mayoría de ellas sigue el siguiente esquema:

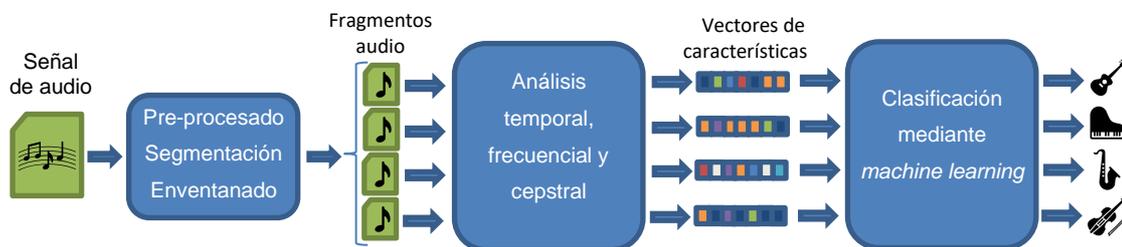


Figura 3. Esquema habitual de los detectores de instrumentos musicales propuestos en la bibliografía

Así, el primer paso suele consistir en dividir la señal en pequeños segmentos para su análisis individualizado. En estos segmentos, las principales propiedades de la señal (amplitud, frecuencia fundamental) son estables lo que permite calcular de forma muy precisa las características de respuesta en tiempo y frecuencia de un instrumento para dichas condiciones. La longitud de los extractos varía en las diferentes propuestas, entre los 20 y los 40 ms, con un solapamiento de entre un 25% y un 60% entre

<sup>5</sup> (Universidad ESAN, 2015)

<sup>6</sup> Se han consultado las principales plataformas de aplicaciones para *smartphones* (Google Play e iTunes) y ordenadores personales (cnet.com)

extractos consecutivos<sup>7</sup>. Para mitigar la aparición de lóbulos en el entorno de los armónicos, la segmentación se suele hacer mediante una ventana de tipo Hann<sup>8</sup> o Hamming<sup>9</sup>.

Dado que la mayor parte de la información relativa al timbre se encuentra en los armónicos secundarios de menor energía; para facilitar la clasificación, en algunos sistemas se realiza un preprocesado de la señal consistente en un filtro paso-alto previo a la segmentación<sup>10</sup>.

## 2.2.2. Análisis y extracción de características

La siguiente etapa común en la mayoría de los sistemas de identificación, consiste en el análisis de cada uno de los extractos con el objetivo de obtener, para cada segmento, un conjunto de valores que lo caracterice completamente y, a su vez, permita su clasificación posterior en función del instrumento que emitió el sonido. Estas características se extraen a partir del análisis temporal, frecuencial y cepstral de cada extracto. A continuación, se comentan las características más comunes que se incluyen en los vectores<sup>11</sup>:

### Análisis temporal

- **Tiempo y pendiente de crecimiento** (*onset time/slope*): corresponde al periodo transcurrido desde que se inicia la interpretación de una nota, en la que la amplitud de la señal va, por lo general, aumentando hasta que se estabiliza.
- **Tiempo y pendiente de decaimiento** (*decay time/slope*): corresponde al periodo final de ejecución de una nota, desde que termina su fase estable hasta que se extingue, en la que la amplitud decrece.
- **Pendiente y error cuadrático medio de la recta ajustada a la curva de media cuadrática en la fase estable** (*rms-energy slope*). Esta curva mide el valor eficaz de la amplitud de una señal y se obtiene como:

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N x^2[n]}$$

- **Factor de amplitud** (*crest factor*) cociente entre la amplitud de la señal y su media cuadrática.
- **Señal de autocorrelación**: se obtiene al sumar los productos muestra a muestra de una señal por una versión retardada de sí misma:

$$R[\tau] = x[n] * x[-n] = \frac{1}{N} \sum_{n=1}^N x[n] \cdot x[n + \tau]$$

<sup>7</sup> (Eronen & Klapuri, Musical instrument recognition using cepstral coefficients and temporal features, 2000), (Eronen, Comparison of features for musical instrument recognition, 2001) y (Maliki & Sofiyannudin, 2018)

<sup>8</sup> (Azarloo & Farokhi, 2012)

<sup>9</sup> (Eronen, Comparison of features for musical instrument recognition, 2001) y (Maliki & Sofiyannudin, 2018)

<sup>10</sup> (Eronen, Comparison of features for musical instrument recognition, 2001) y (Maliki & Sofiyannudin, 2018)

<sup>11</sup> (Eronen & Klapuri, Musical instrument recognition using cepstral coefficients and temporal features, 2000), (Eronen, Comparison of features for musical instrument recognition, 2001), (Maliki & Sofiyannudin, 2018) y (Azarloo & Farokhi, 2012)

Esta señal tiene el doble de muestras que la original, pero solo se incluyen en el vector de características los primeros  $N_R$  valores ( $0 \leq \tau \leq N_R$ ), en función del sistema de clasificación empleado<sup>12</sup>.

- **Zero-crossing rate:** corresponde al número de veces que la amplitud de la señal cambia de signo:

$$ZCR = \frac{1}{2} \sum_{n=1}^N |\text{sgn}(x[n]) - \text{sgn}(x[n-1])|$$

## Análisis frecuencial

- **Frecuencia fundamental,  $f_0$ :** corresponde a la frecuencia más baja de la serie de armónicos de una nota ( $f_0, 2f_0, 3f_0, 4f_0, \dots$ ) y que define el tono (*pitch*) que percibimos. No se debe definir como la frecuencia más baja del espectro ya que la presencia de ruido y de cierta modulación en amplitud en algunas notas puede hacer aparecer componentes de muy baja frecuencia. Tal como se puede observar en la Figura 1, en ocasiones, su intensidad en el espectro es débil en comparación con el resto de armónicos lo que provoca muchos errores de cálculo con las técnicas más sencillas (primer pico fuera de origen en la autocorrelación o en el espectro). Esto ha inducido a la publicación de artículos para abordar su cálculo<sup>13</sup> así como de sistemas de reconocimiento ideados específicamente para evitar tener en cuenta este parámetro y todos los que se derivan del mismo<sup>14</sup>. La tesitura de un instrumento se puede definir como el conjunto de frecuencias fundamentales que es capaz de emitir, y supone un buen parámetro para su clasificación.
- **Desviación estándar de la frecuencia fundamental.**
- **Modulación en amplitud (AM): frecuencia e intensidad,** lo que permite medir el rizado en amplitud (de entre 4 y 40 Hz) que aparece en las notas de algunos instrumentos.
- **Amplitud, valor medio y desviación típica del centroide espectral (*spectral centroid*).** El centroide espectral corresponde al centro de masas del espectro:

$$SC = \frac{\sum_{k=0}^{N-1} X[k] \cdot f[k]}{\sum_{k=0}^{N-1} X[k]}$$

donde  $X[k]$  es la DFT del extracto y  $f[k]$  la frecuencia asociada a la muestra  $k$  de la DFT. Se utilizan tanto las medidas obtenidas a partir de frecuencias absolutas como de frecuencias normalizadas ( $f/f_0$ ).

## Análisis cepstral

El análisis cepstral se comenzó a desarrollar en los años sesenta para extraer la información implícita en las redundancias y periodicidades del espectro de una señal y es, hoy en día, una herramienta fundamental para el funcionamiento de los sistemas de reconocimiento del habla<sup>15</sup>. La definición original del cepstro,  $\hat{x}[m]$ , de una señal  $x[n]$  corresponde a la transformada inversa de Fourier del logaritmo del valor absoluto del espectro:

$$\hat{x}[m] = \mathcal{F}^{-1}\{\log|X[k]|\} = \mathcal{F}^{-1}\{\log|\mathcal{F}\{x[n]\}|\}$$

<sup>12</sup> (Azarloo & Farokhi, 2012)

<sup>13</sup> (Klapuri, 1999)

<sup>14</sup> (Kitahara, Goto, Komatani, Ogata, & Okuno, 2006)

<sup>15</sup> (Oppenheim & Schaffer, 2004)

No obstante, los valores del cepstro no se suelen incluir directamente en el vector de características, si no que se resumen en una serie de coeficientes que condensan su información teniendo en cuenta el funcionamiento perceptivo del oído humano. Estos coeficientes son:

- **Mel Frequency Cepstral Coefficients (MFCC):** Se obtienen aplicando, a la potencia espectral de la señal,  $|X[k]|^2$ , un banco de  $N_F$  filtros triangulares paso-banda espaciados a escala Mel (que refleja de manera más precisa que la escala lineal la percepción de los tonos por el oído humano). Para cada filtrado se suman todas las muestras obtenidas obteniendo  $N_F$  valores que contienen información de la energía contenida en cada banda de Mel. Finalmente, aplicando una transformada de coseno discreta (DCT) a estos valores, se obtienen los coeficientes MFCC, de los cuales se suelen usar solamente los 13 primeros. El cálculo detallado de estos coeficientes se puede consultar en la bibliografía<sup>16</sup>.
- **Delta-MFCC y Doble-delta-MFCC:** Corresponden a la primera y segunda derivada en el tiempo de los coeficientes MFCC y se pueden calcular a partir de los coeficientes MFCC de los extractos anteriores, actual y posteriores. Permiten tener en consideración la evolución temporal del cepstro.

### 2.2.3. Sistemas de clasificación

Una vez obtenido el vector de características de cada extracto es necesario proceder a su clasificación, es decir, determinar qué fuente de sonido es más probable que lo originara a partir de un conjunto de instrumentos conocido y analizado previamente. Dado el gran tamaño de los vectores, la heterogeneidad de los datos que contienen y la ausencia de reglas sencillas de distinción, la mayoría de las propuestas publicadas recurren al aprendizaje automático (*machine learning*) para esta tarea. El *machine learning* es un área de la inteligencia artificial que consiste en un conjunto muy variado de algoritmos matemáticos y estadísticos que, a partir de unos vectores de entrenamiento, determina como se deben transformar y/o analizar las muestras para poder realizar predicciones con el mínimo error. De entre la gran variedad de técnicas, las que se han utilizado en algunos de los artículos de investigación publicados<sup>17</sup> son:

- **Clasificador  $k$ -vecinos más cercanos ( $k$ -nearest neighbors, kNN):** es el sistema más sencillo. Suponiendo que los vectores tienen una dimensión  $n$ , entonces los datos de entrenamiento (cada uno asociado a un instrumento) pueden representarse en el hiperespacio  $\mathbb{R}^n$ . Para clasificar un nuevo vector de origen desconocido, se obtienen los  $k$  vectores más cercanos al mismo según la distancia euclidiana u otra métrica. Finalmente, el vector se clasifica de acuerdo al instrumento mayoritario asociado a los  $k$  vectores más cercanos. El valor de  $k$  es elegido por el diseñador. Este sistema tiene el problema de que requiere que el número de vectores de entrenamiento esté equilibrado para todos los posibles casos ya que, si tenemos muchas más muestras de un instrumento en comparación con otro similar, el sistema kNN tenderá a clasificar según el instrumento más representado en el entrenamiento.

<sup>16</sup> (Sreenivasa Rao & Vuppala, 2014)

<sup>17</sup> (Eronen & Klapuri, Musical instrument recognition using cepstral coefficients and temporal features, 2000), (Ghosh, Pal, Sil, & Palit, 2018), (Maliki & Sofiyannudin, 2018) y (Azarloo & Farokhi, 2012)

- **Clasificador gaussiano:** Este sistema calcula las funciones de densidad de probabilidad en el hiperespacio  $\mathbb{R}^n$  para cada uno de los instrumentos a clasificar, asumiendo que cada uno de los valores de los vectores sigue una distribución gaussiana. Así, cada muestra es clasificada según el instrumento que tiene mayor densidad de probabilidad para el vector  $n$ . Este sistema evita los errores por sobrerepresentación en los test del clasificador kNN, pero muchas veces las variables incluidas en los vectores no siguen una distribución gaussiana.
- **Máquinas de vectores de soporte (Support Vector Machines, SVM).** Este sistema realiza una operación de transformación sobre los valores de los vectores de test en  $\mathbb{R}^n$  de manera que queden agrupados según su clasificación en un hiperespacio de dimensión superior  $\mathbb{R}^m$  ( $m > n$ ), o incluso de infinitas dimensiones, de forma que las diferentes clases puedan separarse fácilmente mediante hiperplanos en  $\mathbb{R}^m$  con el mínimo error. Así, la clasificación de un nuevo vector quedará definida por el instrumento asociado a la región de  $\mathbb{R}^m$  en la que se sitúe después de realizar las transformaciones.
- **Redes neuronales artificiales (Artificial Neural Networks, ANN).** Este sistema simula el comportamiento de las neuronas humanas y, por lo tanto, constituye una herramienta muy útil en el reconocimiento de patrones difíciles de definir de forma sencilla a partir de los datos. Una red neuronal consta de  $N$  entradas, una por cada dimensión del vector de características y  $M$  salidas, una por cada clase distinta a clasificar. Entre las capas de entrada y salida se sitúan  $K$  capas de  $J_1, \dots, J_K$  neuronas. Los valores de  $K$  y  $J_k$  son a elección del diseñador. Cada neurona realiza una operación matemática que es función lineal corresponde a la suma de un valor constante único para cada neurona más una combinación lineal de los valores de salida de las neuronas de la capa anterior, en que los coeficientes de la combinación son también únicos para cada conexión entre neuronas.

$$f(a_{1,k-1}, a_{2,k-1}, \dots, a_{J_{k-1},k-1}) = \omega_{j,k,0} + \sum_{h=1}^{J_{k-1}} \omega_{j,k,h} \cdot a_{h,k-1}$$

Los coeficientes  $\omega$  se llaman pesos, y al iniciar el sistema toman valores aleatorios. Durante la fase de entrenamiento, se espera que para cada vector de entrada  $x$  los valores de salida de la última capa sean  $y = [0,0, \dots, 0,1,0, \dots, 0]$  siendo todos nulos excepto el asignado a la categoría de  $x$  que se desea que tome valor 1. Al principio, como los pesos son aleatorios, la salida también será aleatoria, pero a partir del error de la misma respecto al resultado deseado se procede a actualizar el valor de los pesos para minimizarlo. Después de entrenar el sistema y corregirlo mediante la introducción de todos los vectores de test en varias tandas<sup>18</sup> se llega a un punto en que el error no se minimiza más. Una vez está la red neuronal entrenada, se clasifica un vector nuevo según la categoría correspondiente a la salida de mayor valor de la red.

- **Modelos ocultos de Markov (Markov Hidden Model, HMM).** Se trata de un clasificador dinámico que tiene en cuenta las transiciones entre vectores consecutivos. Este sistema de clasificación toma los diferentes vectores de características como estados y obtiene una matriz de probabilidades de transición entre ellos. De esta forma, se puede calcular la probabilidad de que una determinada

<sup>18</sup> Las tandas de entrenamiento reciben el nombre de épocas y pueden ser miles.

secuencia de vectores haya sido emitida por cierto instrumento, y la clasificación se realiza asignando el instrumento con mayor probabilidad.

- **Árbol de decisión** (*Decision Tree*). Consiste en un diagrama en el que se evalúan de forma sucesiva diferentes condiciones referentes a valores de los vectores de características. En función de si se cumple o no cada condición se accede a una nueva rama del diagrama; de ahí su nombre. La elección de las mejores condiciones a evaluar en cada paso se realiza de forma sistemática a partir del análisis de los vectores de entrenamiento. Los diversos caminos del árbol de decisión terminan con la asignación del instrumento más probable en función de las condiciones verificadas.
- **Random Forest** (Bosque aleatorio). Constituye una sofisticación del árbol de decisión. En este sistema, se construyen un gran número de árboles de decisión diferentes, en los que las condiciones a evaluar, en vez de ser escogidas sistemáticamente, se seleccionan de forma aleatoria. Una vez construido el bosque, se procesa cada vector de características por todos los árboles, obteniendo una predicción por cada uno de ellos. Finalmente, la clasificación definitiva del vector de características se asigna a la categoría que más predicciones ha obtenido.

#### 2.2.4. Resultados y propuestas innovadoras

Resulta difícil comparar directamente los diferentes sistemas propuestos en la bibliografía ya que cada uno de ellos utiliza un conjunto distinto de categorías a clasificar. En el artículo de Eronen & Kapluri de 2000<sup>19</sup> se consigue una tasa de aciertos del 80% con 30 instrumentos utilizando clasificadores gaussianos y kNN. En el artículo de Eronen de 2001<sup>20</sup> se clasifican señales conteniendo una única nota obteniendo una tasa de aciertos del 35% con 29 instrumentos y clasificadores kNN. En el artículo de Maliki & Sofiyanudin<sup>21</sup> no se hace una clasificación por instrumentos sino por familias, consiguiendo una tasa del 89% utilizando SVM. Por otro lado, en el artículo de Azarloo & Farokhi<sup>22</sup> se intenta distinguir entre 8 categorías de instrumentos individuales o conjuntos concretos consiguiendo un 98% de aciertos con red neuronal y un 94% con kNN.

Dado que la tasa de aciertos solo es muy alta para conjuntos reducidos de categorías a clasificar, algunos investigadores han publicado propuestas muy diferentes a las explicadas y en las que se incluyen técnicas ajenas al procesamiento de sonido. Así, el artículo de Kitahara y otros<sup>23</sup> presenta una herramienta llamada instrograma en la que, mediante el análisis de las señales de entrenamiento, se representa la probabilidad de que un instrumento  $\omega_i$  suene para cada frecuencia fundamental y tiempo desde el origen de la pieza  $p(\omega_i; t, f_0)$ , así para cada audio procesado se obtiene la probabilidad de pertenencia a un instrumento determinado. La tasa de aciertos varía entre el 78% y el 97% utilizando solo un conjunto de 4 instrumentos, pero con la posibilidad de mezclarlos en el mismo audio siempre que las frecuencias fundamentales de los distintos instrumentos se mantengan en bandas distintas.

<sup>19</sup> (Eronen & Kapluri, Musical instrument recognition using cepstral coefficients and temporal features, 2000)

<sup>20</sup> (Eronen, Comparison of features for musical instrument recognition, 2001)

<sup>21</sup> (Maliki & Sofiyanudin, 2018)

<sup>22</sup> (Azarloo & Farokhi, 2012)

<sup>23</sup> (Kitahara, Goto, Komatani, Ogata, & Okuno, 2006)

En el artículo de Zlatintsi & Maragos<sup>24</sup> se aplica el concepto matemático de dimensión fractal sobre la amplitud de la señal de audio obteniendo un conjunto de coeficientes que contienen información relativa a la evolución temporal y desviaciones típicas para cada instrumento y nota. En el artículo se muestra que para 4 de los 8 instrumentos analizados, estos coeficientes aumentan ligeramente la tasa de aciertos que se consigue solo con los MFCC.

Finalmente, en el artículo de Ghosh y otros<sup>25</sup> se convierte la señal de sonido de cada nota en una imagen 2D de tamaño  $\sqrt{N} \times \sqrt{N}$ , donde  $N$  es la longitud del audio (aplicando previamente *zero-padding* hasta que  $\sqrt{N}$  sea entero). Una vez obtenida la matriz, se calcula un vector de características con técnicas de procesamiento de imagen y mediante clasificación utilizando *Random Forest* se consigue una tasa de aciertos que varía desde el 41% para la viola al 100% del oboe, siendo 9 el número de instrumentos distintos clasificados.

---

<sup>24</sup> (Zlatintsi & Maragos, 2013)

<sup>25</sup> (Ghosh, Pal, Sil, & Palit, 2018)

## 3.Propuesta

### 3.1. Objetivos del sistema

El objetivo principal del sistema a desarrollar en este trabajo es la detección de instrumentos musicales en señales de audio con el requisito de poder diferenciar varios instrumentos distintos sonando separada o simultáneamente en la misma señal y sin que existan incompatibilidades en cuanto a las posibles combinaciones, lo que constituye una innovación respecto a las soluciones propuestas en los artículos publicados.

La arquitectura del sistema será flexible para poderse aplicar en la categorización de cualquier conjunto de instrumentos que se desee, siempre que se cuente con un corpus de señales de audio de los mismos suficientemente completo para un buen entrenamiento del clasificador. No obstante, deben excluirse los instrumentos de percusión de tipo “no afinado” ya que para el correcto funcionamiento de la aplicación es requisito imprescindible que los instrumentos emitan notas musicales de tono perceptible y no sonidos fruto de impactos cuyo contenido frecuencial se encuentra disperso de forma continua en el espectro.

A partir del análisis de la señal de audio se extraerán un conjunto de vectores de características que el sistema procederá a clasificar individualmente, mostrando un gráfico con el porcentaje de vectores clasificados en cada categoría. Si el usuario indica que solo suena un instrumento en el archivo, la predicción (respuesta) del sistema corresponderá a la categoría con más vectores clasificados.

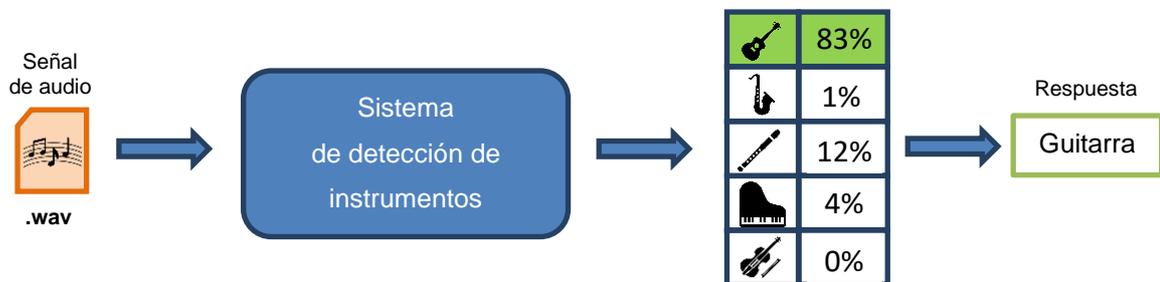


Figura 4. Esquema básico de funcionamiento del sistema planteado

El sistema no tendrá la capacidad de determinar por sí mismo el número de instrumentos distintos que suenan simultáneamente en la señal de audio, por lo que el usuario deberá introducir el número  $N_i$  de instrumentos a detectar, proporcionando entonces el sistema una respuesta con las  $N_i$  categorías con mayor número de vectores clasificados. Tanto el prototipo como los experimentos realizados se han limitado a  $N_i \leq 4$  ya que las tasas de acierto no son perfectas y a mayor número de instrumentos más posibilidades hay de proporcionar una respuesta errónea. No obstante, si el archivo contuviera más instrumentos, o bien, si el usuario desconociera su número, podrá utilizar la representación gráfica de los porcentajes de vectores clasificados en cada categoría como referencia para conjeturar la posible respuesta.

### 3.2. Especificaciones del prototipo

La interfaz gráfica diseñada como prototipo para probar el sistema planteado tiene las siguientes especificaciones.

- Las categorías aceptadas incluyen todos los instrumentos de cuerda y metal habituales en las orquestas incluyendo piano y clavicémbalo. También se han añadido instrumentos populares no orquestales como la guitarra, el ukelele, el acordeón y el órgano. Además, se ha incluido de forma experimental un instrumento de percusión de tipo afinado: el xilófono. El listado completo de instrumentos, agrupados por subfamilias instrumentales es el siguiente:

<b>Cuerda frotada</b> <ul style="list-style-type: none"> <li>• Violín <b>VN</b></li> <li>• Viola <b>VA</b></li> <li>• Violoncelo <b>VC</b></li> <li>• Contrabajo <b>CB</b></li> </ul>	<b>Cuerda pulsada</b> <ul style="list-style-type: none"> <li>• Arpa <b>AR</b></li> <li>• Clavicémbalo <b>CV</b></li> <li>• Guitarra <b>GT</b></li> <li>• Ukelele <b>UK</b></li> </ul>	<b>Cuerda percutada</b> <ul style="list-style-type: none"> <li>• Piano <b>PI</b></li> </ul>
<b>Viento Bisel</b> <ul style="list-style-type: none"> <li>• Flautín <b>FN</b></li> <li>• Flauta travesera <b>FT</b></li> <li>• Flauta dulce <b>FD</b></li> </ul>	<b>Viento Lengüeta Simple</b> <ul style="list-style-type: none"> <li>• Clarinete <b>CL</b></li> <li>• Clarinete Bajo <b>CJ</b></li> <li>• Saxofón <b>SX</b></li> </ul>	<b>Viento Lengüeta Doble</b> <ul style="list-style-type: none"> <li>• Oboe <b>OB</b></li> <li>• Corno Inglés <b>CI</b></li> <li>• Fagot <b>FA</b></li> <li>• Contrafagot <b>CF</b></li> </ul>
<b>Viento Metal</b> <ul style="list-style-type: none"> <li>• Trompeta <b>TP</b></li> <li>• Trompa <b>TA</b></li> <li>• Trombón <b>TN</b></li> <li>• Tuba <b>TB</b></li> </ul>	<b>Viento Lengüetas Libres</b> <ul style="list-style-type: none"> <li>• Acordeón <b>AC</b></li> <li>• Órgano <b>OR</b></li> </ul>	<b>Percusión</b> <ul style="list-style-type: none"> <li>• Xilófono <b>XI</b></li> </ul>

Tabla 2. Instrumentos incluidos en el prototipo con sus códigos de identificación

Junto a cada instrumento aparece el código de 2 caracteres que, para mejorar la legibilidad y presentación, se utilizará a lo largo de todo este trabajo para identificar la categoría en las distintas tablas.

- El prototipo admite para su análisis archivos de sonido en formato WAVE (.wav) tanto mono como estéreo y con cualquier frecuencia de muestreo. El sistema trabaja internamente con  $f_s = 44,1$  kHz y señales mono, por lo que, de forma previa a la extracción de características, la aplicación realiza las transformaciones necesarias para convertir la señal a esta configuración.

# 4. Diseño

## 4.1. Arquitectura general del sistema

El siguiente diagrama de flujo muestra la arquitectura de funcionamiento del sistema desarrollado y que se detallará en los siguientes apartados.

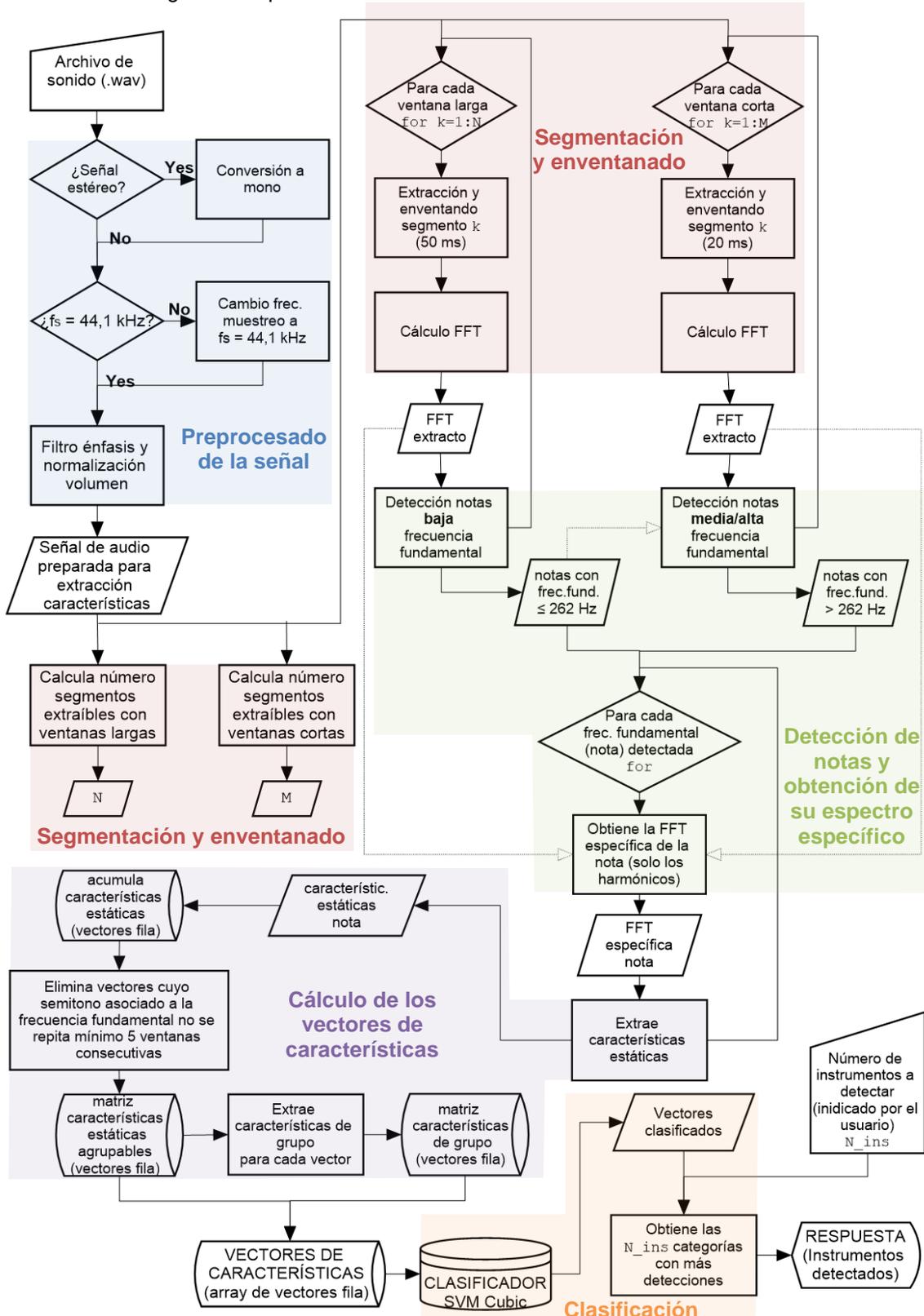


Figura 5. Diagrama de flujo del sistema diseñado

## 4.2. Preprocesado de la señal

El sistema de extracción de características diseñado trabaja exclusivamente con señales de audio en mono y con una frecuencia de muestreo de  $f_s = 44,1$  kHz. El uso de un único formato es muy útil para simplificar la programación de las funciones de procesamiento, así como para asegurar que las mediciones calculadas sobre la señal sean homogéneas. En caso de no cumplir la señal de entrada alguna de estas especificaciones se aplican las siguientes transformaciones:

- Conversión de estéreo a mono. Se implementa generando una señal de un único canal fruto de la suma de los canales izquierdo y derecho.
- Cambio de frecuencia de muestreo a  $f_s$  (función `interpolacion_fs`) que se basa fundamentalmente en la función `resample` de MATLAB.

Una vez obtenida la señal con un formato homogéneo, y tal como sucede en la mayoría de las propuestas revisadas en el Estado del arte, se somete la misma a un filtrado de énfasis para altas frecuencias. El motivo es conseguir enfatizar los armónicos de orden superior de las notas, puesto que estos constituyen un rasgo descriptivo del timbre de cada instrumento. Para seleccionar el filtro a específico a utilizar, durante la fase de desarrollo del sistema se ha experimentado con 3 opciones:

- No utilizar filtro de énfasis,  $h[n] = 1$
- Utilizar el filtro de énfasis sugerido en algunos de los artículos publicados<sup>26</sup> cuya respuesta impulsional es:  $h[n] = \delta[n] - 0.95\delta[n - 1]$
- Utilizar un filtro de énfasis de altas frecuencias con menor atenuación en baja frecuencia, cuya respuesta impulsional es:  $h[n] = 1.2\delta[n] - 0.8\delta[n - 1]$ .

El motivo de realizar estas pruebas es que se observa que en la respuesta en frecuencia del filtro de énfasis habitual las frecuencias inferiores a 1000 Hz quedan fuertemente atenuadas (amplificación inferior a 0,2); lo que plantea la hipótesis de que podría tener efectos contraproducentes para los instrumentos con tesituras más graves como el contrabajo o el contrafagot, con frecuencias fundamentales que pueden ser del orden de hasta 50 Hz, de forma que todos sus armónicos quedarán muy atenuados.

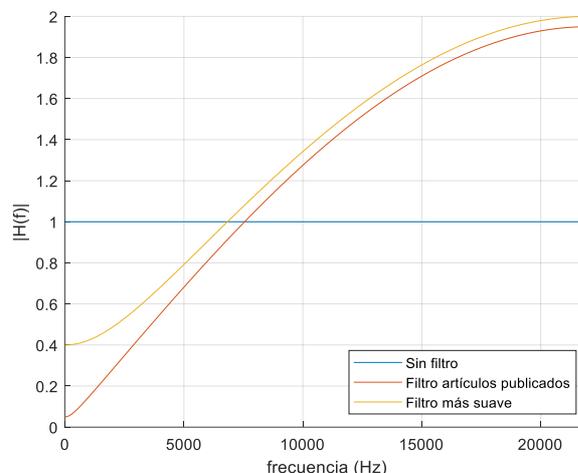


Figura 6. Filtros de preprocesado probados en el desarrollo del sistema

<sup>26</sup> (Eronen, Comparison of features for musical instrument recognition, 2001) y (Maliki & Sofiyannudin, 2018)

En una fase posterior de desarrollo del sistema, se procedió a realizar una serie de pruebas con algunos sistemas de *machine learning* para comprobar si los vectores de características propuestos contenían información suficiente para que los algoritmos pudieran distinguir las diferentes categorías. En estos experimentos se obtuvo la tasa de aciertos en la clasificación de vectores utilizando tan solo subconjuntos de 4 y 10 instrumentos distintos. Cada prueba se repitió con los vectores obtenidos con los diferentes filtros de preprocesado propuestos para así comprobar cual era la opción más idónea. En la siguiente tabla se observan los resultados obtenidos con los sistemas de *machine learning* que obtuvieron mejores tasas. Para cada experimento se muestra la tasa de aciertos general, así como la tasa correspondiente a la categoría con menor ratio de aciertos.

Experimentos ARCHIVOS UTILIZADOS	Tipo filtro preprocesado	Sistema <i>machine learning</i>			
			Red Neuronal <sup>27</sup>	Random Forest	Decision Tree
4 archivos instrumentos distintos FN2, VC3, CL4, TB2	Ninguno	<b>Aciertos</b> 96,5% Peor tasa 82,0%	<b>99,1%</b> 91,3%	<b>92,7%</b> 57,5%	
	Filtro artículos publicados	<b>Aciertos</b> 98,0% Peor tasa 87,5%	<b>99,1%</b> 84,6%	<b>95,4%</b> 63,7%	
	Filtro suave	<b>Aciertos</b> 96,7% Peor tasa 82,4%	<b>99,2%</b> 91,9%	<b>94,5%</b> 40,3%	
4 archivos instrumentos agudos VN6, VA3, FN3, FT1	Ninguno	<b>Aciertos</b> 92,4% Peor tasa 90,1%	<b>98,7%</b> 98,0%	<b>85,3%</b> 80,6%	
	Filtro artículos publicados	<b>Aciertos</b> 94,9% Peor tasa 92,5%	<b>99,4%</b> 98,7%	<b>91,2%</b> 83,6%	
	Énfasis suave	<b>Aciertos</b> 93,9% Peor tasa 91,4%	<b>99,4%</b> 99,2%	<b>87,7%</b> 81,5%	
4 archivos instrumentos graves VC3, CB4, FA3, CF5	Ninguno	<b>Aciertos</b> 96,4% Peor tasa 93,2%	<b>99,2%</b> 97,9%	<b>92,1%</b> 85,9%	
	Filtro artículos publicados	<b>Aciertos</b> 99,8% Peor tasa 99,1%	<b>99,7%</b> 96,9%	<b>98,5%</b> 96,9%	
	Énfasis suave	<b>Aciertos</b> 97,2% Peor tasa 94,8%	<b>99,2%</b> 98,0%	<b>91,8%</b> 82,9%	
20 archivos de 10 instrumentos distintos VA1, VA2, CB1, CV2, AR1, AR2, FN1, FN2, OB1, OB2, CJ1, CJ2, FA1, FA2, TN1, TN2, TB1, TB2, OR1, OR2	Ninguno	<b>Aciertos</b> 83,4% Peor tasa 66,9%	<b>98,1%</b> 93,5%	<b>62,5%</b> 19,8%	
	Filtro artículos publicados	<b>Aciertos</b> 91,7% Peor tasa 75,2%	<b>98,6%</b> 95,4%	<b>74,4%</b> 95,4%	
	Énfasis suave	<b>Aciertos</b> 86,1% Peor tasa 69,2%	<b>98,1%</b> 93,2%	<b>64,7%</b> 24,9%	

Tabla 3. Comparación rendimiento de los distintos filtros de preprocesado

Se puede observar que para 10 de los 12 experimentos realizados, el filtro publicado en los artículos es el que mejor tasa de aciertos obtiene. Este resultado, además, es persistente para los 3 sistemas de clasificación en la prueba más compleja en la que hay 10 categorías involucradas. Por lo tanto, a la vista de los datos obtenidos, se descarta la hipótesis realizada en cuanto a que el filtro propuesto en los artículos podría dificultar la detección de instrumentos de tesitura media-baja. Por lo tanto, para la implementación del sistema se utiliza el filtro cuya respuesta impulsional es  $h[n] = \delta[n] - 0.95\delta[n - 1]$  y que se programa en la función `filtrado_preenfasis`.

Finalmente, antes de pasar al bloque de segmentación y inventariado, es necesario realizar una última función de preprocesado con tal de homogeneizar la intensidad (o volumen) de las señales que se van a analizar. Esta homogeneización es necesaria ya que, para evitar extraer vectores de características del ruido

<sup>27</sup> La red neuronal se ha probado con 10 neuronas para las pruebas de 4 instrumentos y con 20 neuronas para las pruebas de 10 instrumentos.

de fondo de la señal, se exigirá un nivel mínimo de energía para poder calcular un vector. Por lo tanto, si no se normaliza el volumen, el número de vectores extraído será muy desigual en señales distintas, pudiendo llegar a ser insuficiente si la señal está muy atenuada.

En un principio se había planteado realizar la normalización amplificando las señales de forma que su amplitud máxima en valor absoluto fuera 1. Sin embargo, después de realizar la primera extracción de vectores de características del corpus, se observaron tasas muy desiguales de vectores extraídos por segundo de señal. En algunos archivos de varios minutos a penas se extraían una decena de vectores, mientras que, en otros casos, se llegaban a extraer más de 100 vectores de características por segundo.

Por ello, se optó por calcular la raíz de la amplitud cuadrática media<sup>28</sup> para algunas de las señales analizadas y comprobar si tenía alguna relación con la tasa de vectores extraídos. A partir de la representación gráfica de los resultados se observa claramente que, por un lado, a mayor amplitud más vectores se extraen, como era previsible y, por otro, que la normalización imponiendo el máximo a 1 no homogeniza bien el volumen de las señales dando resultados muy dispares.

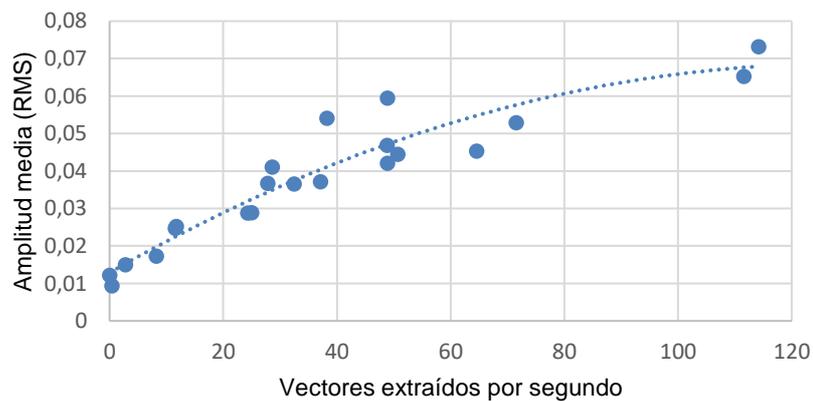


Figura 7. Relación entre amplitud RMS de una señal y el número de vectores extraídos

Así pues, se observó necesario rectificar el diseño del método de normalización de volumen para igualar la amplitud RMS media de los archivos, en vez de la amplitud máxima. Para ello se ha programado la función `normalizar_volumen` que homogeniza el volumen de las señales realizando una amplificación de forma que la amplitud media RMS resultante sea de 0,05.

### 4.3. Conceptos de teoría musical

Antes de proseguir con la descripción del resto de bloques del sistema, es necesario comentar algunos conceptos de teoría musical en torno al elemento que va a describir cada vector de características extraído: la nota. En un sistema de estas características, es de esperar que las señales de audio contengan información de una serie de notas musicales interpretadas por distintos instrumentos. Estas notas, a diferencia de otros sonidos, se caracterizan por tener una tonalidad perceptible de manera que su contenido en el dominio frecuencial se limita al entorno de la frecuencia fundamental  $f_0$  y de sus armónicos  $nf_0$  con

<sup>28</sup> También llamada amplitud RMS, del inglés *root mean square*.

$n \in \mathbb{N}^+$ . En la siguiente figura se muestra el aspecto de un espectro de la interpretación de una nota con frecuencia fundamental  $f_0 = 440$  Hz, lo que en notación musical corresponde a la nota LA<sub>4</sub>.

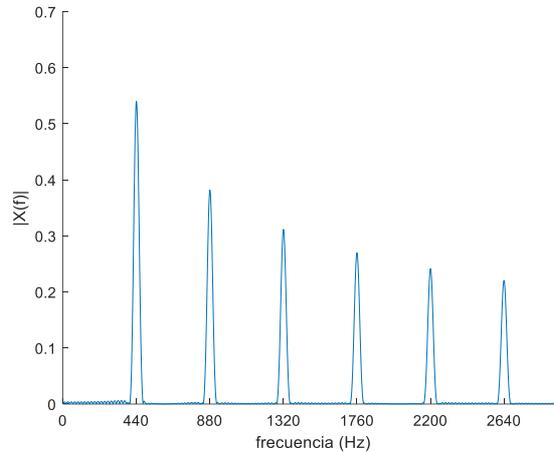


Figura 8. Espectro teórico de una nota de 440 Hz

El pilar fundamental del sistema diseñado se basa en la detección de estos trenes de armónicos en el espectro, teniendo en cuenta que pueden coexistir varios de ellos superpuestos en un mismo segmento de la señal.

En general se puede suponer que los audios a analizar proceden de instrumentos que están correctamente afinados de acuerdo a la escala musical convencional. Esto implica que las notas emitidas tienen una frecuencia fundamental siempre en el entorno de uno de los semitonos de dicha escala. La frecuencia fundamental teórica de los distintos semitonos se obtiene a partir de la expresión:

$$f(n) = 440 \cdot 2^{\frac{n}{12}} \text{ [Hz]}$$

donde  $n \in \mathbb{Z}$  es el número de semitono. La frecuencia fundamental del semitono  $n = 0$  corresponde a la nota LA<sub>4</sub> (440 Hz) y el rango de tesituras del conjunto de instrumentos más habituales abarca buena parte de las capacidades de audición humana desde los 32,7 Hz del DO<sub>1</sub> (semitono  $n = -45$ ) hasta los 4.186 Hz del DO<sub>8</sub> (semitono  $n = 39$ ).

En la Figura 8 se puede observar que los lóbulos de los armónicos ocupan cierto espacio en el espectro debido a su naturaleza finita en el tiempo. Dado que para diferenciar dos notas simultáneas es necesario detectar claramente sus respectivas frecuencias fundamentales, es importante que los lóbulos de estos primeros armónicos interfieran entre sí lo menos posible. Por ello, de cara al diseño del bloque de segmentación y enventanado se hace uso de un concepto de armonía musical que simplifica la complejidad del problema: la disonancia<sup>29</sup>, que es el fenómeno que ocurre cuando dos notas interpretadas simultáneamente suenan de forma desagradable.

<sup>29</sup> (Ball, 2011)

La teoría de armonía musical<sup>30</sup> indica que la emisión simultánea de dos notas cuya diferencia es de 1 o 2 semitonos se considera armónicamente muy disonante y, por lo tanto, no se encontrará de forma habitual en la mayoría de las piezas musicales. Así, el problema se simplifica si se tiene en cuenta que, en general, la distancia mínima entre 2 frecuencias fundamentales simultáneas será de 3 semitonos.

#### 4.4. Segmentación y enventanado

La progresión de las interpretaciones musicales hace que estas señales de sonido sean muy variables en el tiempo ya que se suceden distintos tonos, volúmenes, silencios e instrumentos. Además, las características diferenciales del timbre, como es el caso de las amplitudes de los armónicos, raramente se mantienen estables como se pudo observar en la Figura 1. Por ello, no tiene sentido extraer características del conjunto de la señal, sino que, para captar sus distintos estados, es necesario segmentar la señal en fragmentos muy cortos en los que las distintas características de la señal se mantengan casi constantes de forma que se puedan realizar mediciones suficientemente precisas para describir el estado de la señal en cada instante.

Para realizar esta segmentación, además, es muy conveniente aplicar una función de enventanado sobre cada extracto, de forma que la señal no termine abruptamente en los extremos. Si no se aplica este paso, se realiza lo que se conoce como enventanado rectangular en el que cada componente sinusoidal del espectro, que idealmente toma la forma de impulso, queda reemplazado por una función sinc, con un lóbulo principal proporcional a la amplitud del armónico y multitud de pequeños lóbulos secundarios que no proporcionan información alguna y que distorsionan la información contenida en esa región.

Dadas las necesidades de nuestro sistema, comentadas en el apartado anterior, resulta de mucho interés que los lóbulos principales de los armónicos de distintas notas no interfieran entre sí, así como que los lóbulos secundarios sean lo más pequeños posibles para no distorsionar las amplitudes de otros armónicos. Por ello, la transformación de cada armónico en una función sinc resulta perjudicial para el sistema y, por lo tanto, conviene aplicar una función de enventanado distinta a la rectangular.

En un principio, y tal como se dispone en los distintos sistemas propuestos en diferentes artículos publicados<sup>31</sup>, se pensó en utilizar un único sistema de enventanado mediante la extracción de fragmentos de duración  $T_x = 20$  ms. Este intervalo es suficientemente pequeño para poder capturar de forma precisa las características instantáneas de la señal, así como observar con más detalle su evolución.

No obstante, en el momento de comparar las diferentes funciones de enventanado posibles con esta longitud de segmento, se detectó un importante problema de diseño. Para la elección del sistema de enventanado se procedió a representar las respuestas en frecuencia de las ventanas más habituales

---

<sup>30</sup> (Zamacois, 1973)

<sup>31</sup> (Eronen, Comparison of features for musical instrument recognition, 2001) y (Maliki & Sofiyannudin, 2018)

(Rectangular, Hamming, Hann, Blackman y Kaiser con  $\beta = 2.5$ ) con ventanas de longitud  $T_x = 20$  ms y frecuencia de muestreo  $f_s = 44,1$  kHz, lo que implica que cada ventana contiene  $N_x = 882$  muestras.

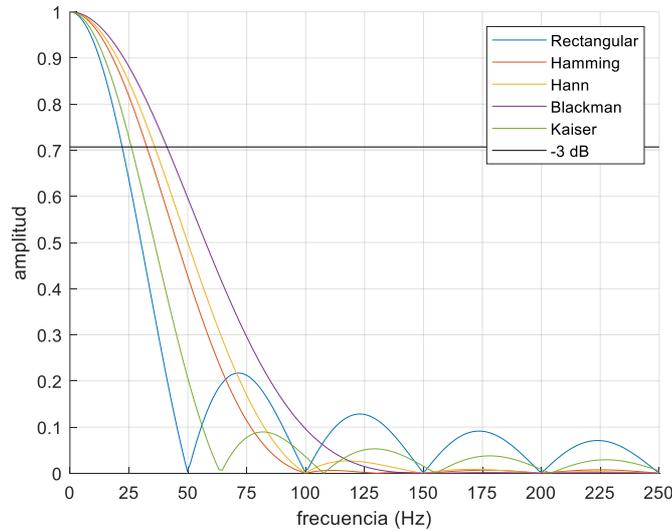


Figura 9. Comparación de sistemas de enventanado para segmentos cortos de 20 ms

Entonces, se observó que el lóbulo principal es bastante ancho para todos los métodos, lo que puede producir la interferencia de los lóbulos del primer armónico en notas concurrentes con frecuencia fundamental cercana. Las ventanas rectangular y Kaiser presentan, además, fuertes lóbulos secundarios que tienen la capacidad de alterar la amplitud de otros armónicos. Así, de las ventanas sin distorsión perceptible debido a lóbulos secundarios, la que mejor comportamiento presenta es la ventana de Hamming, pero la amplitud de su lóbulo principal solo disminuye a -3 dB respecto del pico a partir de los 32 Hz de distancia.

Así pues, considerando 32 Hz como la distancia mínima que puede distinguir este tipo de enventanado sin solapar los lóbulos principales y, tomando una distancia mínima de 2 semitonos<sup>32</sup> entre notas simultáneas, entonces, aplicando la ecuación del apartado 4.3, los semitonos  $n$  de las frecuencias fundamentales compatibles con este enventanado deberán verificar:

$$440 \text{ Hz} \cdot 2^{\frac{n+2}{12}} - 440 \text{ Hz} \cdot 2^{\frac{n}{12}} \geq 32 \text{ Hz} \rightarrow n \gtrsim -9 \rightarrow f_0 \gtrsim 261,6 \text{ Hz}$$

Por lo tanto, la ventana de duración  $T_x = 20$  ms que se ha considerado solo tiene capacidad para distinguir correctamente las frecuencias fundamentales de notas igual o superiores a 261,6 Hz (DO<sub>4</sub>). En este caso, la nota situada 2 semitonos por encima es el RE<sub>4</sub> (293,7 Hz) y su diferencia es precisamente de 32 Hz. Por debajo, la diferencia entre tonos se reduce y el lóbulo principal de la ventana de Hamming es demasiado ancho para evitar interferencias en el momento de detectar frecuencias fundamentales próximas entre sí.

Dado que muchos instrumentos emiten habitualmente notas por debajo de 261 Hz, es necesario modificar el diseño planteado para incluir dicho rango. Una solución sería utilizar una ventana más larga, lo que hace

<sup>32</sup> Como se ha comentado, se puede suponer una distancia mínima de hasta 3 semitonos.

que se reduzca la amplitud de los lóbulos, pero como contrapartida, se perdería la inmediatez de las características obtenidas ya que habría más variabilidad en los datos. Por lo tanto, para evitar aumentar la variabilidad, se opta por una solución de compromiso más flexible, aunque añadiendo complejidad al sistema, y que consiste en hacer dos procesos paralelos de segmentación. Por un lado, se mantiene la ventana escogida de longitud  $T_x = 20$  ms y  $N_x = 882$  muestras, pero limitando su uso a notas cuya frecuencia fundamental esté, como mínimo, en el entorno del semitono -9 (DO<sub>4</sub>). Por el otro, para el rango de bajas frecuencias se diseña una ventana específica más larga de longitud  $T_x = 50$  ms y  $N_x = 2.205$  muestras. La respuesta de los diferentes sistemas de enventanado para esta nueva longitud es:

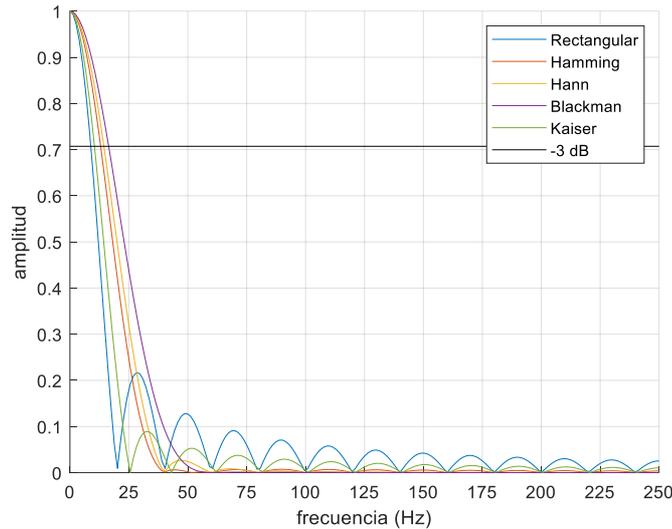


Figura 10. Comparación de sistemas de enventanado para segmentos largos de 50 ms

En este caso, el enventanamiento de Hamming vuelve a proporcionar la respuesta más estrecha sin lóbulos secundarios. Además, presenta un nivel de -3 dB a unos 12 Hz de distancia del pico. Si ahora, para abarcar el máximo rango posible, consideramos una distancia mínima de 3 semitonos entre notas simultáneas, entonces el semitono  $n$  mínimo de las frecuencias fundamentales compatibles con este enventanado debe cumplir:

$$440 \text{ Hz} \cdot 2^{\frac{n+3}{12}} - 440 \text{ Hz} \cdot 2^{\frac{n}{12}} \geq 12 \text{ Hz} \rightarrow n \geq -33 \rightarrow f_0 \geq 65,4 \text{ Hz}$$

Así, mediante la ventana larga se podrán considerar notas de tonalidades a partir del DO<sub>2</sub> (65,4 Hz) ya que la distancia a la nota situada 3 semitonos por encima, el RE#<sub>2</sub> (77,8 Hz), es de 12 Hz. El rango de frecuencias fundamentales situadas por debajo del semitono -33 queda excluido del sistema, lo que no supone un problema ya que esta banda tiene una sonoridad extremadamente grave de forma que su uso en composiciones musicales es muy poco frecuente.

Dado que al realizar la operación de enventanado de Hamming, las muestras en los extremos de cada fragmento quedan fuertemente atenuadas; con tal de no perder esa información y para mejorar el grado de seguimiento de la señal, se considera necesario solapar entre sí los diferentes segmentos. Tanto para las ventanas cortas como largas, establecemos un valor de solapamiento de tal forma que el instante de inicio

entre segmentos consecutivos sea de 10  $\mu$ s para ambos sistemas. De esta forma, ambos métodos mantienen uniformidad en el seguimiento de la evolución de las características de la señal. Y, además, se facilita el establecimiento de una relación unívoca que relacione cada ventana corta con una ventana larga coincidente en el tiempo. En el siguiente cuadro se resumen las propiedades definitivas de los dos sistemas de enventanamiento a utilizar:

Tipo ventana	Longitud temporal	Muestras segmento	Solapamiento $\sigma$	Separación segmentos	Función enventanamiento	Rango frecuencias
Corta	$T_x = 20$ ms	$N_x = 882$	50%	$\Delta T = 10$ ms	Hamming	$\geq DO_4$ $\geq 262$ Hz
Larga	$T_x = 50$ ms	$N_x = 2205$	80%	$\Delta T = 10$ ms	Hamming	$DO_2 - SI_3$ 65 – 262 Hz

Tabla 4. Características de los dos procesos de segmentación utilizados

El establecimiento de una relación unívoca entre cada ventana corta con una ventana larga es necesario debido a que los armónicos de notas de baja frecuencia detectadas en las ventanas largas pueden ser confundidos con frecuencias fundamentales de alta frecuencia en las ventanas cortas. Por ello, en el proceso posterior de detección de notas, antes de analizar los tonos de media/alta frecuencia presentes en una ventana corta se consultarán las notas de baja frecuencia halladas en una ventana larga coincidente en el tiempo. Relacionar cada ventana corta con una ventana larga es sencillo si representamos su duración en el tiempo. Para el caso hipotético de una señal de tan solo 160 ms, los segmentos que se extraerán son:

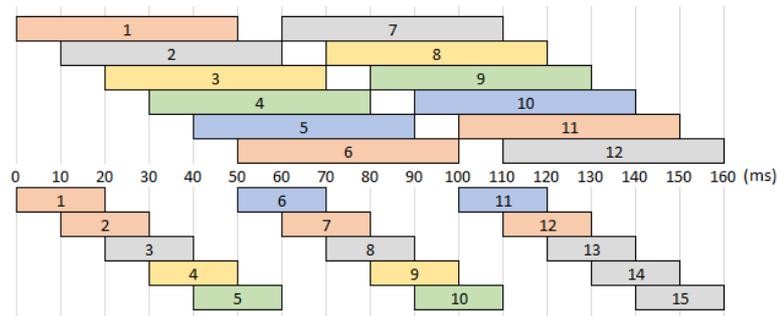


Figura 11. Relación temporal entre ventanas largas y cortas

Se puede observar que, con excepción de las ventanas extremas, la ventana corta de orden  $k$  queda completamente contenida dentro de la ventana larga de orden  $k - 1$ , de manera que se usará esta relación. En las ventanas cortas extremas (la primera y las dos últimas) la relación se asignará respectivamente a la primera y última ventana larga, tal como muestra el coloreado de la figura.

Para realizar los procesos de segmentación y enventanado se recorrerá la señal mediante un bucle `for` utilizando las funciones `extraccion_segmento` y `enventanado_hamming` en cada iteración. Con tal de cuadrar la longitud completa de la misma con el final de una ventana se realiza una operación de `zero-padding` al final del audio (función `zero_padding`).

Finalmente, el último paso de este bloque consiste en obtener la FFT<sup>33</sup> de cada segmento. En este sistema se calcula la transformación de cada extracto utilizando  $N_{FFT} = 2^{15}$  muestras con tal de obtener una resolución en frecuencia lo suficientemente pequeña para facilitar la ubicación de las frecuencias fundamentales y sus armónicos. Con dicho valor, la resolución frecuencial en el sistema es de:

$$\Delta f = \frac{f_s}{2^{15}} = 1,35 \text{ Hz.}$$

El cálculo de la FFT se programa en la función `respuesta_frecuencial_normalizada`, que basándose en la función `fft` de MATLAB, obtiene únicamente el valor absoluto de las frecuencias positivas y multiplica el resultado por  $\frac{2}{N_{FFT}}$  para que la amplitud de los lóbulos en el dominio frecuencial coincida con la amplitud de los componentes en el dominio temporal.

#### 4.5. Detección de notas y obtención de su espectro específico

Este bloque tiene el objetivo de detectar y diferenciar las distintas notas que estén contenidas en cada segmento a partir de la detección de sus frecuencias fundamentales en el espectro obtenido mediante la FFT. Aún en el caso de tener solamente una nota presente, la detección del valor exacto de la frecuencia fundamental utilizando los métodos clásicos<sup>34</sup> es muy complicada. Un motivo radica en que muchas veces el primer armónico está muy atenuado respecto armónicos de orden superior, como puede apreciarse en la Figura 1. Y, por otro lado, dado que la FFT es una función discreta, el pico de la frecuencia fundamental  $\hat{f}_0$  tendrá normalmente un error de resolución  $\varepsilon$  respecto al valor real  $f_0$ .

$$\hat{f}_0 = f_0 + \varepsilon$$

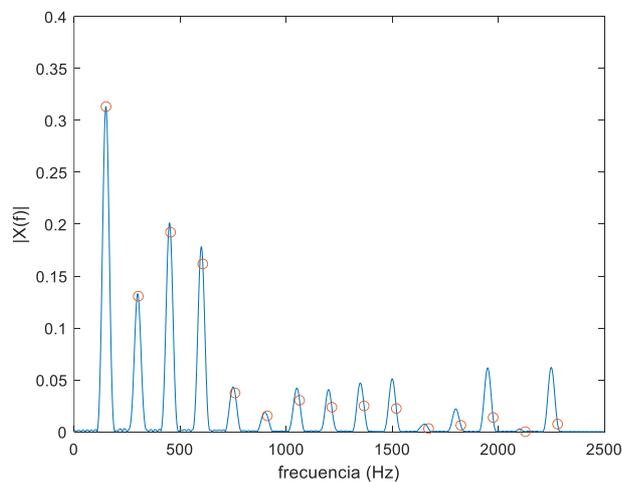


Figura 12. Propagación del error al estimar la frecuencia fundamental con métodos clásicos

Esto implica que al evaluar las amplitudes de los distintos armónicos a partir de los múltiplos del pico  $\hat{f}_0$  el error se propaga de forma creciente, pasando a ser  $n \cdot \varepsilon$ , de tal forma que los valores tomados acaban

<sup>33</sup> *Fast Fourier Transform*, algoritmo que permite obtener de forma rápida la transformada de Fourier discreta de una señal, siempre que su número de puntos sea  $2^n$  con  $n \in \mathbb{N}^+$ .

<sup>34</sup> Como por ejemplo, la detección del primer pico presente en el espectro o en la función de autocorrelación.

no correspondiéndose con las amplitudes reales de los armónicos tal como puede verse en la Figura 12 con una nota de 150 Hz.

$$nf_0 = nf_0 + n\varepsilon$$

Por todo lo expuesto, durante el diseño de este sistema se ha ideado un algoritmo para calcular la frecuencia fundamental que parte del hecho de que todos los picos de lóbulos que se encuentran en la FFT indican la posición aproximada de un armónico de alguna frecuencia fundamental, con un error máximo correspondiente a la resolución en frecuencia, aunque desconociendo a priori el orden  $k$  del mismo.

$$k\widehat{f}_0 = kf_0 + \varepsilon$$

A partir de esta aproximación, si se consigue estimar correctamente el orden  $k$  de los armónicos, entonces se puede calcular de forma mucho más exacta su frecuencia fundamental ya que en la estimación el error queda dividido por el orden del armónico.

$$\tilde{f}_0 = \frac{k\widehat{f}_0}{k} = \frac{kf_0 + \varepsilon}{k} = f_0 + \frac{\varepsilon}{k}$$

Además, dado que  $\varepsilon$  es diferente en cada armónico, para alguno de ellos el error de resolución será mínimo propiciando una aproximación aún más exacta de la frecuencia fundamental.

Así pues, el algoritmo ideado consiste en encontrar, para cada frecuencia fundamental presente, el armónico  $k\widehat{f}_0$  que presente menor error de resolución y calcular su índice  $k$ . Para ello, se parte de la suposición de que el armónico con menor error será el que, a partir de su posición y su grado hipotético  $k$ , las posiciones del resto de armónicos estén en conjunto más alineadas con los picos y, por lo tanto, la suma de energías en la FFT para estas ubicaciones sea máxima. Como se desconoce el índice  $k$ , es necesario repetir los cálculos para todos los picos y los posibles índices  $k$ <sup>35</sup>.

En resumen, siendo  $\hat{p} = k\widehat{f}_0$  cada uno de los picos de la FFT, se desea encontrar aquel pico con menor error de resolución  $p_{opt}$  y su orden  $k_{opt}$  que se caracterizarán por maximizar la suma de las energías de los armónicos derivados del mismo.

$$(p_{opt}, k_{opt}) = \arg \max_{(p,k)} \sum_{h=1}^N \left| X^2 \left( h \frac{\hat{p}}{k} \right) \right|$$

donde  $X(f)$  es la FFT.

A partir de esta optimización, la aproximación de la frecuencia fundamental se obtendrá sencillamente como:

$$\tilde{f}_0 = \frac{p_{opt}}{k_{opt}} = \frac{k_{opt}f_0 + \varepsilon_{min}}{k_{opt}} = f_0 + \frac{\varepsilon_{min}}{k_{opt}}$$

<sup>35</sup> En este trabajo se considera que las distintas notas tienen como máximo 15 armónicos, por lo que  $k \leq 15$ .

No obstante, hay que tener presente, que este sistema se ha diseñado con el requisito de que puedan sonar varias notas a la vez, de forma que pueden coexistir diversas frecuencias fundamentales. Además, en el diseño se ha considerado que una nota tiene como máximo 15 armónicos, por lo tanto, para cada pico en la FFT se toma la hipótesis de que pueda corresponder a cualquiera de estos 15 órdenes, lo que a su vez genera 15 posibles frecuencias fundamentales hipotéticas.

Así pues, es necesario efectuar varios procesos sucesivos de descarte de las frecuencias fundamentales hipotéticas hasta obtener únicamente las correspondientes a las notas que suenen realmente en el extracto.

Para mostrar como se ha diseñado este proceso, el cual se ha programado en las funciones `frecuencias_fundamentales` y `análisis_harmonicos`, se dispone el siguiente ejemplo con un segmento con dos notas de frecuencias 500 Hz y 650 Hz más cierto ruido gaussiano. En la Figura 13 se puede observar a la izquierda el espectro de los dos tonos por separado, mientras que a la derecha se ha representado la FFT del segmento incluyendo ruido gaussiano.

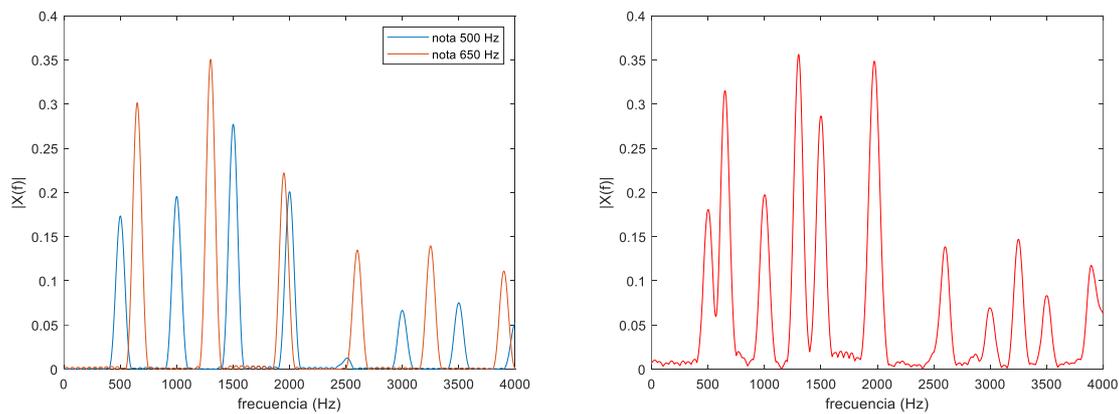


Figura 13. FFT de un segmento con dos notas y ruido gaussiano.

El primer paso es encontrar las posiciones de los picos de la FFT, pero no conviene extraer cualquier máximo local ya que hay muchos pequeños picos fruto del ruido y del entevantamiento. Por ello, mediante la función `picos_significativos` se obtiene una relación de los picos cuya amplitud sea superior a  $\mu + \frac{\sigma}{2}$  donde  $\mu$  es la media de la FFT y  $\sigma$  su desviación típica. En la Figura 14 se muestran las posiciones de los 12 primeros picos encontrados y su ubicación en el eje de frecuencias.

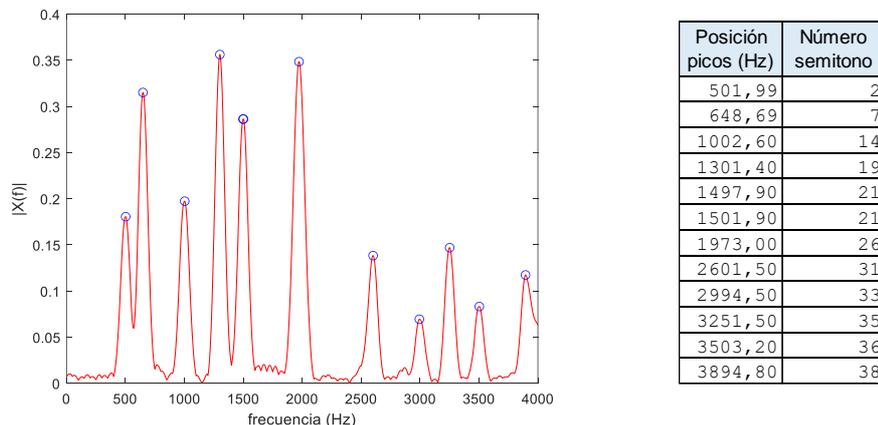


Figura 14. Picos significativos detectados en la FFT del segmento

A continuación, para cada pico detectado se plantea la hipótesis de que pueda ser un armónico de cualquier orden entre 1 y 15. Para cada posible orden se obtienen las posiciones de los otros armónicos. Para ello, se multiplica la ubicación de cada pico por la matriz  $a_{ij} = \frac{i}{j}$ , de forma que en la fila  $k$  del resultado se obtiene la posición de los 15 armónicos suponiendo que el pico corresponde al armónico  $k$ .

$$\widehat{kf_0} \cdot \begin{pmatrix} 1 & 2 & 3 & \dots & 15 \\ \frac{1}{1} & \frac{2}{1} & \frac{3}{1} & \dots & \frac{15}{1} \\ \frac{1}{2} & \frac{2}{2} & \frac{3}{2} & \dots & \frac{15}{2} \\ \frac{1}{3} & \frac{2}{3} & \frac{3}{3} & \dots & \frac{15}{3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{15} & \frac{2}{15} & \frac{3}{15} & \dots & \frac{15}{15} \end{pmatrix} = \widehat{kf_0} \cdot (a_{ij}) \quad \text{donde } a_{ij} = \frac{i}{j}$$

Entonces, para cada posición indicada en las filas de la matriz resultante, se evalúa la FFT y se suman las energías obteniendo una tabla con todas las hipótesis ordenándose de mayor a menor energía detectada, cuyos primeros 15 resultados son:

Ubicación pico (Hz)	Orden armónico	Energía total (dB)	Frecuencia fundam. (Hz)	Número semitono
1972,98	12	-3,31	164,41	-17
1301,41	6	-3,53	216,90	-12
3251,51	15	-3,53	216,77	-12
2601,48	12	-3,55	216,79	-12
648,69	3	-3,62	216,23	-12
1301,41	8	-3,69	162,68	-17
1301,41	4	-3,82	325,35	-5
3251,51	10	-3,84	325,15	-5
2601,48	8	-3,84	325,19	-5
1301,41	2	-3,87	650,71	7
3251,51	5	-3,88	650,30	7
2601,48	4	-3,88	650,37	7
1497,90	9	-3,90	166,43	-17
9097,78	14	-3,90	649,84	7
5848,96	9	-3,90	649,88	7

Tabla 5. Primer listado de posibles frecuencias fundamentales en segmento de ejemplo (15 primeras filas)

Puede parecer a priori que el método no funciona ya que los primeros resultados corresponden a frecuencias fundamentales del entorno de 163, 216 y 325 Hz. Sin embargo, se puede observar que estos tres valores tienen en común que en alguno de sus armónicos se sitúa una frecuencia fundamental real ( $163 \cdot 3 \approx 500$ ,  $163 \cdot 4 \approx 650$ ,  $216 \cdot 3 \approx 650$ ,  $325 \cdot 2 \approx 650$ ) por lo que al calcular su suma de energías se han incluido varios armónicos reales de una o varias notas.

Estos falsos resultados se pueden eliminar muy fácilmente si se toma la suposición de que el armónico fundamental de una nota tiene que presentar necesariamente un lóbulo mínimamente pronunciado y, por lo tanto, existir en su entorno un pico significativo en la FFT. Para aplicar esta regla, se considerará que existe un pico en el entorno de una posible frecuencia fundamental si alguno de los picos detectados tiene el mismo número de semitono. En la Figura 14 se detallan los números de semitonos de cada pico significativo detectado. A partir de este listado de valores, se procede a eliminar cualquier hipótesis de frecuencia fundamental cuyo número de semitono no forme parte de la relación. Una vez realizados estos descartes, los primeros 15 resultados obtenidos son:

Ubicación pico (Hz)	Orden harmónico	Energía total (dB)	Frecuencia fundam. (Hz)	Número semitono
1301,41	2	-3,87	650,71	7
3251,51	5	-3,88	650,30	7
2601,48	4	-3,88	650,37	7
9097,78	14	-3,90	649,84	7
5848,96	9	-3,90	649,88	7
8447,74	13	-3,90	649,83	7
7793,67	12	-3,93	649,47	7
3894,82	6	-3,96	649,14	7
648,69	1	-4,02	648,69	7
1972,98	3	-4,44	657,66	7
4509,86	7	-4,98	644,27	7
2994,46	6	-5,40	499,08	2
1497,90	3	-5,43	499,30	2
7000,98	14	-5,49	500,07	2
4998,39	10	-5,49	499,84	2

Tabla 6. Segundo listado de posibles frecuencias fundamentales en segmento de ejemplo (15 primeras filas)

En esta tabla ya se observan claramente las frecuencias fundamentales en los entornos de 500 Hz y 650 Hz. Para cada caso se obtienen diversos resultados, de los cuales, la hipótesis que obtiene una mayor energía se considera la aproximación más fiable. Por lo tanto, el siguiente paso consiste en dejar solamente el resultado de mayor energía para cada número de semitono. Esto reduce la tabla a únicamente 6 resultados:

Ubicación pico (Hz)	Orden harmónico	Energía total (dB)	Frecuencia fundam. (Hz)	Número semitono
1301,41	2	-3,87	650,71	7
2994,46	6	-5,40	499,08	2
1501,94	1	-10,11	1501,94	21
3251,51	1	-16,20	3251,51	35
7000,98	2	-18,44	3500,49	36
4509,86	1	-20,53	4509,86	40
4998,39	1	-22,73	4998,39	42

Tabla 7. Tercer listado de posibles frecuencias fundamentales en segmento de ejemplo

Se observa que la lista la encabezan las aproximaciones de las frecuencias fundamentales reales. No obstante, existen 5 candidaturas erróneas adicionales. Estos errores se producen al confundir armónicos secundarios con otras posibles frecuencias fundamentales. Por ejemplo 1501 Hz corresponde aproximadamente al tercer armónico de 500 Hz, de manera que en su suma de energías se contienen los armónicos de orden 3, 6, 9, 12 y 15 de la nota de 500 Hz y por eso aparece presente en la tabla. No obstante, estas sumas serán por lo general inferiores a la que se obtiene con una frecuencia fundamental real. Así, el último proceso de descarte se realiza de forma iterativa para cada fila de mayor a menor energía. En cada caso se obtiene la frecuencia fundamental  $f_0$  y se procede a eliminar cualquier resultado de menor energía cuyo semitono sea equivalente al de sus armónicos  $nf_0$ <sup>36</sup>. La tabla final resultante de este proceso es la siguiente:

Ubicación pico (Hz)	Orden harmónico	Energía total (dB)	Frecuencia fundam. (Hz)	Número semitono
1301,41	2	-3,87	650,71	7
2994,46	6	-5,40	499,08	2

Tabla 8. Listado final de frecuencias fundamentales detectadas en segmento de ejemplo

Así, se detectan pues dos frecuencias fundamentales de valores 499,08 Hz y 650,71 Hz y en las que se ha reducido en un 54% y 46% respectivamente el error absoluto respecto al método de la posición del primer

<sup>36</sup> En el sistema diseñado se eliminan los resultados hasta  $10f_0$  lo que supone más de 3 escalas de diferencia.

pico (501,99 Hz y 648,69 Hz). Nótese que en alguna de las hipótesis descartadas se facilitaban resultados aún más precisos, pero debido a la distorsión producida por el ruido y el solapamiento de las dos notas, la suma de las energías no ha proporcionado su valor máximo.

En la Figura 15 se pueden observar los dos tonos separados, así como los valores de amplitud de armónicos que se obtienen utilizando las frecuencias fundamentales detectadas mediante el algoritmo presentado. Se puede observar que las mediciones obtenidas son por lo general bastante precisas con valores muy cercanos a los picos reales, con la excepción de los armónicos que se superponen completamente en el entorno de 2000 Hz. El hecho de tener errores puntuales en algunos armónicos no debe suponer un problema ya que el clasificador del sistema debe tener la suficiente capacidad de generalización como para no clasificar erróneamente los vectores fruto de una FFT con unos pocos armónicos distorsionados.

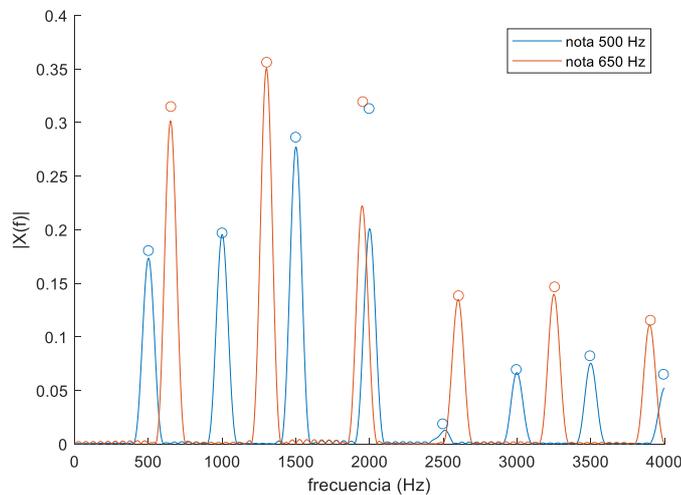


Figura 15. Amplitudes de los armónicos detectadas en el segmento

En las primeras pruebas de detección de frecuencias fundamentales se observó que se obtenían muchos resultados en segmentos que contenían únicamente ruido gaussiano sin señal útil. En ese tipo de segmentos, lo deseable es no obtener ninguna frecuencia fundamental, por lo que, para refinar el diseño, en la fase anterior se exige que la suma de energías mínima sea de -30 dB para aceptar una posible frecuencia fundamental.

Por otro lado, en las funciones anteriores se ha tenido en cuenta el doble proceso de inventariado del sistema. Así, en función del tipo de ventana utilizada solo se admiten aquellas posibles frecuencias fundamentales dentro de los rangos admitidos en cada caso. Además, en los cálculos con ventanas cortas, se pasa previamente como parámetro las frecuencias fundamentales de banda baja halladas en la ventana larga asociada, logrando así evitar que armónicos de las frecuencias fundamentales graves puedan ser confundidos como nuevas frecuencias fundamentales más agudas.

El último paso de este bloque es, para cada frecuencia fundamental detectada, extraer el espectro asociado a los lóbulos de los armónicos que conforman dicha nota. En este sistema se denomina "espectro

específico de la nota “ a este concepto, y simplemente consiste en tomar la FFT del segmento y anularla para todas las frecuencias que no se encuentren en el entorno de las posiciones de los armónicos<sup>37</sup>.

$$X_{esp,f_0}(f) = \begin{cases} X(f) & \text{si } \exists k \text{ tal que } |f - kf_0| < \beta \\ 0 & \text{para otro caso} \end{cases} \quad k \in \{1,2,3, \dots, 15\}$$

Esta tarea se ha programado en la función `espectro_especifico` y en la siguiente figura se observan los resultados obtenidos para las dos notas solapadas de los ejemplos.

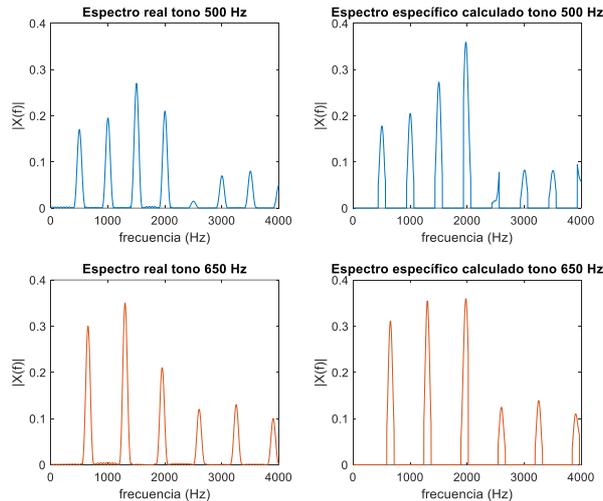


Figura 16. Espectros específicos obtenidos a partir de un segmento con dos notas simultáneas

Así, es posible extraer una versión aproximada del espectro de cada nota por separado reduciendo además la presencia de ruido. Esta FFT específica de cada nota será la base desde la que se obtendrá cada vector de características. Como se ha comentado, al solaparse armónicos de notas distintas se genera cierta distorsión que deberá ser solventada por el clasificador. En la Figura 16 se puede observar como la mayoría de armónicos conservan aproximadamente su forma, excepto los que se solapan en torno a 2000 Hz.

#### 4.6. Cálculo de los vectores de características

Una de las principales diferencias del sistema desarrollado respecto a las propuestas publicadas es que no se obtendrá un único vector de características por segmento de señal, sino que se obtendrán tantos vectores como notas (o frecuencias fundamentales) detectadas con tal de poder facilitar el reconocimiento de instrumentos simultáneos. Por lo tanto, el elemento del que se obtienen todas las mediciones incluidas en el vector es el espectro específico de cada nota.

Durante la primera fase de desarrollo del sistema se ha recopilado un conjunto de 128 características distintas que se extraen de cada nota detectada. Estas características se dividen en tres tipos:

<sup>37</sup> En el diseño desarrollado, el ancho de banda que se deja sin anular a cada lado de la ubicación de un armónico es de  $\beta = 64$  Hz en ventanas cortas y  $\beta = 24$  Hz en ventanas largas.

- Características **estáticas**: son las que se obtienen directamente a partir del espectro específico de la nota a representar en el vector.
- Características **dinámicas**: corresponden a primeras y segundas derivadas en el tiempo de características estáticas, de forma que dependen de unos pocos vectores anteriores y posteriores.
- Características **de grupo**: corresponden a mediciones estadísticas (medias y desviaciones típicas) realizadas sobre las características estáticas de un conjunto contiguo de vectores y cuyo resultado se asigna en bloque a todos los integrantes del grupo utilizado en el cálculo.

Para poder calcular las características dinámicas y de grupo es necesario antes agrupar todos aquellos vectores que representan a una misma nota en distintos instantes de su duración, ya que no tiene sentido obtener características en las que se mezcle información correspondiente a distintas notas y/o instrumentos. Como la longitud de las ventanas utilizadas es más corta que la duración de la gran mayoría de notas en una pieza, sería de esperar que la frecuencia fundamental de dicha nota se detectara en varios segmentos consecutivos. No obstante, eso no ocurre ya que la frecuencia fundamental de una nota no es constante en el tiempo, bien debido a pequeñas modulaciones en frecuencia producidas de forma natural por el instrumento o bien por fluctuaciones en la estimación del valor. Sin embargo, si se utiliza el concepto de número de semitono de la frecuencia fundamental, sí es posible obtener valores consecutivos en el tiempo, ya que, si el número de semitono variase, musicalmente correspondería a una nota distinta.

Así pues, en este sistema se entenderá como “grupo” a cada conjunto de vectores consecutivos en el tiempo cuya frecuencia fundamental tenga el mismo número de semitono. El conjunto de vectores de un grupo representa a una misma nota en sus distintas fases. Las derivadas en el tiempo de las características dinámicas, así como las mediciones estadísticas de las características de grupo se calcularán exclusivamente a partir de las características estáticas de los vectores pertenecientes a un mismo grupo. Además, se utilizará también el concepto de grupo para eliminar todos los vectores que sean fruto de ruidos u otras imperfecciones en la señal o bien posibles errores en la detección de una frecuencia. Se puede suponer que este tipo de situaciones no se repetirá en ventanas consecutivas, por lo que en el sistema se procederá a descartar todos aquellos vectores que no puedan agruparse en al menos 5 ventanas seguidas. De esta forma, para poder extraer vectores, es necesario que la nota tenga una duración mínima de  $5 \cdot \Delta T = 50 \mu s$ . Este mínimo de elementos también supone una garantía para que tenga sentido el cálculo de las características dinámicas y de grupo. Este proceso se programa en la función `agrupar_vectores`.

A partir de las premisas anteriores, se constituye inicialmente un vector conformado por 128 características: 40 estáticas, 68 dinámicas y 20 de grupo. Muchas de estas características aparecen en algunos de los artículos publicados<sup>38</sup> mientras que otras, como la amplitud relativa de los armónicos, sus derivadas en el tiempo o la pendiente de la amplitud en la región local, constituyen una nueva aportación. El listado completo de estas características se muestra en la Tabla 9. No obstante, antes de proseguir con el diseño es

<sup>38</sup> (Eronen & Klapuri, Musical instrument recognition using cepstral coefficients and temporal features, 2000), (Eronen, Comparison of features for musical instrument recognition, 2001), (Maliki & Sofiyannudin, 2018), (Azarloo & Farokhi, 2012), (Peeters, 2004)

necesario analizar si las características escogidas son realmente útiles para la detección de los instrumentos musicales, así como el grado de importancia que tienen para facilitar la detección. Aunque hay varios métodos formales para obtener esta medición, para este trabajo se ha preferido un sistema que ofrece unos resultados muy sencillos de interpretar.

El método se inicia entrenando una red neuronal con vectores extraídos de todos los archivos del corpus de entrenamiento (Anexo A) excepto los correspondientes al primer archivo de cada categoría que se reservan para calcular las tasas de aciertos. Una vez entrenada la red neuronal, se obtiene una selección de vectores de características de los archivos reservados para test y se calcula la tasa de vectores clasificados correctamente por la red neuronal. Concretamente, para este caso se ha obtenido una tasa de aciertos del 43,6% de vectores.

A continuación, se procede a permutar aleatoriamente los valores de cada característica entre los diferentes vectores (manteniendo el resto de las características intactas) y se obtiene la tasa de aciertos con los vectores distorsionados. Esta nueva tasa de aciertos oscilará entre la tasa aleatoria de aciertos de acuerdo al número de categorías ( $\frac{1}{26} = 3,85\%$ ) y la tasa que se ha obtenido sin adulterar vectores (43,6%). Si se obtiene el valor mínimo entonces esa característica es vital para clasificar correctamente y se le asigna una importancia del 100%. En cambio, si se obtiene la tasa máxima de aciertos es que la característica es totalmente superflua ya que su adulteración no provoca una reducción en el número de aciertos, lo que equivale a una importancia del 0%. Para obtener mayor precisión, cada categoría se mezcla aleatoriamente 5 veces y el valor asignado a la importancia corresponde a la media de los valores obtenidos en cada una de las permutaciones. En la Tabla 9 se indica la importancia calculada para cada categoría con este método, que se ha representado en la Figura 17.

A raíz de los resultados presentados en la Tabla 9, se obtienen las siguientes conclusiones:

- Las características dinámicas (excepto la pendiente de la amplitud en la región local y su error) son totalmente superfluas ya que al manipularlas la tasa de aciertos sigue intacta.
- Los coeficientes MFCC tienen mucha importancia, incluso para los índices más altos utilizados habitualmente<sup>39</sup>, por lo que podría ser positivo incluir más coeficientes.
- Las características de grupo basadas en medias estadística tienen, en general, mucha importancia, por lo que podría ser positivo añadir más características de esta índole.
- La amplitud relativa de los armónicos es útil para la detección hasta el 13º armónico. La media del primer armónico (que en la tabla corresponde al coeficiente “media primer triestímulo”) ofrece un buen resultado por lo que puede ser interesante añadir más valores medios de amplitudes de armónicos.

<sup>39</sup> En aplicaciones de procesamiento de sonido, se suelen utilizar tan solo los 13 primeros coeficientes MFCC.

#	Característica	Tipo	Aciertos al permutar	Importancia
1	frecuencia fundamental	estático	42,5%	2,9%
2	amplitud relativa harmónico 1	estático	39,2%	11,1%
3	amplitud relativa harmónico 2	estático	40,1%	9,0%
4	amplitud relativa harmónico 3	estático	41,3%	5,8%
5	amplitud relativa harmónico 4	estático	42,8%	2,2%
6	amplitud relativa harmónico 5	estático	42,3%	3,4%
7	amplitud relativa harmónico 6	estático	41,6%	5,2%
8	amplitud relativa harmónico 7	estático	42,9%	1,9%
9	amplitud relativa harmónico 8	estático	40,3%	8,4%
10	amplitud relativa harmónico 9	estático	42,6%	2,6%
11	amplitud relativa harmónico 10	estático	42,9%	1,9%
12	amplitud relativa harmónico 11	estático	42,9%	1,8%
13	amplitud relativa harmónico 12	estático	40,2%	8,5%
14	amplitud relativa harmónico 13	estático	43,3%	0,8%
15	amplitud relativa harmónico 14	estático	43,4%	0,7%
16	amplitud relativa harmónico 15	estático	43,5%	0,5%
17	suma total amplitudes harmónicos	estático	41,8%	4,6%
18	amplitud estimada nota	estático	43,3%	0,7%
19	crest factor	estático	38,8%	12,2%
20	centroide espectral (relativo)	estático	31,6%	30,3%
21	coeficiente MFCC 1	estático	21,2%	56,5%
22	coeficiente MFCC 2	estático	28,3%	38,5%
23	coeficiente MFCC 3	estático	30,1%	34,1%
24	coeficiente MFCC 4	estático	38,1%	14,0%
25	coeficiente MFCC 5	estático	35,8%	19,7%
26	coeficiente MFCC 6	estático	36,1%	18,9%
27	coeficiente MFCC 7	estático	37,8%	14,6%
28	coeficiente MFCC 8	estático	38,7%	12,5%
29	coeficiente MFCC 9	estático	38,2%	13,6%
30	coeficiente MFCC 10	estático	36,2%	18,6%
31	coeficiente MFCC 11	estático	37,7%	15,0%
32	coeficiente MFCC 12	estático	39,9%	9,3%
33	coeficiente MFCC 13	estático	39,6%	10,2%
34	1ª derivada frecuencia fundamental	dinámico	43,6%	0,0%
35	1ª derivada amplitud relativa harm. 1	dinámico	43,6%	0,0%
36	1ª derivada amplitud relativa harm. 2	dinámico	43,6%	0,0%
37	1ª derivada amplitud relativa harm. 3	dinámico	43,6%	0,0%
38	1ª derivada amplitud relativa harm. 4	dinámico	43,6%	0,0%
39	1ª derivada amplitud relativa harm. 5	dinámico	43,6%	0,0%
40	1ª derivada amplitud relativa harm. 6	dinámico	43,6%	0,0%
41	1ª derivada amplitud relativa harm. 7	dinámico	43,6%	0,0%
42	1ª derivada amplitud relativa harm. 8	dinámico	43,6%	0,0%
43	1ª derivada amplitud relativa harm. 9	dinámico	43,6%	0,0%
44	1ª derivada amplitud relativa harm. 10	dinámico	43,6%	0,0%
45	1ª derivada amplitud relativa harm. 11	dinámico	43,6%	0,0%
46	1ª derivada amplitud relativa harm. 12	dinámico	43,6%	0,0%
47	1ª derivada amplitud relativa harm. 13	dinámico	43,6%	0,0%
48	1ª derivada amplitud relativa harm. 14	dinámico	43,6%	0,0%
49	1ª derivada amplitud relativa harm. 15	dinámico	43,6%	0,0%
50	1ª deriv. suma total amplitudes harm.	dinámico	43,6%	0,2%
51	1ª derivada amplitud estimada nota	dinámico	43,6%	0,1%
52	1ª derivada crest factor	dinámico	43,6%	0,0%
53	1ª derivada centroide espectral	dinámico	43,6%	0,1%
54	1ª derivada coeficiente MFCC 1	dinámico	43,6%	0,1%
55	1ª derivada coeficiente MFCC 2	dinámico	43,6%	0,0%
56	1ª derivada coeficiente MFCC 3	dinámico	43,6%	0,0%
57	1ª derivada coeficiente MFCC 4	dinámico	43,6%	0,1%
58	1ª derivada coeficiente MFCC 5	dinámico	43,6%	0,0%
59	1ª derivada coeficiente MFCC 6	dinámico	43,6%	0,0%
60	1ª derivada coeficiente MFCC 7	dinámico	43,6%	0,0%
61	1ª derivada coeficiente MFCC 8	dinámico	43,6%	0,0%
62	1ª derivada coeficiente MFCC 9	dinámico	43,6%	0,0%
63	1ª derivada coeficiente MFCC 10	dinámico	43,6%	0,0%
64	1ª derivada coeficiente MFCC 11	dinámico	43,6%	0,0%
65	1ª derivada coeficiente MFCC 12	dinámico	43,6%	0,0%
66	1ª derivada coeficiente MFCC 13	dinámico	43,6%	0,0%
67	2ª derivada frecuencia fundamental	dinámico	43,6%	0,0%
68	2ª derivada amplitud relativa harm. 1	dinámico	43,6%	0,0%
69	2ª derivada amplitud relativa harm. 2	dinámico	43,6%	0,1%
70	2ª derivada amplitud relativa harm. 3	dinámico	43,6%	0,1%
71	2ª derivada amplitud relativa harm. 4	dinámico	43,6%	0,0%
72	2ª derivada amplitud relativa harm. 5	dinámico	43,6%	0,0%
73	2ª derivada amplitud relativa harm. 6	dinámico	43,6%	0,1%
74	2ª derivada amplitud relativa harm. 7	dinámico	43,7%	-0,1%
75	2ª derivada amplitud relativa harm. 8	dinámico	43,6%	0,1%
76	2ª derivada amplitud relativa harm. 9	dinámico	43,6%	0,1%
77	2ª derivada amplitud relativa harm. 10	dinámico	43,6%	0,0%
78	2ª derivada amplitud relativa harm. 11	dinámico	43,6%	0,0%
79	2ª derivada amplitud relativa harm. 12	dinámico	43,6%	0,0%
80	2ª derivada amplitud relativa harm. 13	dinámico	43,6%	0,0%
81	2ª derivada amplitud relativa harm. 14	dinámico	43,6%	0,0%
82	2ª derivada amplitud relativa harm. 15	dinámico	43,6%	0,0%
83	2ª deriv. suma total amplitud.harmónicos	dinámico	43,5%	0,3%
84	2ª derivada amplitud estimada nota	dinámico	43,6%	0,1%
85	2ª derivada crest factor	dinámico	43,6%	0,1%
86	2ª derivada centroide espectral	dinámico	43,6%	0,1%
87	2ª derivada coeficiente MFCC 1	dinámico	43,6%	0,1%
88	2ª derivada coeficiente MFCC 2	dinámico	43,6%	0,1%
89	2ª derivada coeficiente MFCC 3	dinámico	43,6%	0,1%
90	2ª derivada coeficiente MFCC 4	dinámico	43,6%	0,1%
91	2ª derivada coeficiente MFCC 5	dinámico	43,6%	0,1%
92	2ª derivada coeficiente MFCC 6	dinámico	43,6%	0,0%
93	2ª derivada coeficiente MFCC 7	dinámico	43,6%	0,1%
94	2ª derivada coeficiente MFCC 8	dinámico	43,6%	0,0%
95	2ª derivada coeficiente MFCC 9	dinámico	43,6%	0,1%
96	2ª derivada coeficiente MFCC 10	dinámico	43,6%	0,0%
97	2ª derivada coeficiente MFCC 11	dinámico	43,6%	0,0%
98	2ª derivada coeficiente MFCC 12	dinámico	43,6%	0,0%
99	2ª derivada coeficiente MFCC 13	dinámico	43,6%	0,0%
100	desv.estándar frecuencia fundamental	de grupo	42,8%	2,2%
101	segundo triestímulo	estático	42,2%	3,7%
102	tercer triestímulo	estático	40,2%	8,6%
103	media primer triestímulo	de grupo	35,5%	20,4%
104	media segundo triestímulo	de grupo	33,7%	24,9%
105	media tercer triestímulo	de grupo	35,4%	20,7%
106	desv.estándar primer triestímulo	de grupo	40,4%	8,1%
107	desv.estándar segundo triestímulo	de grupo	42,7%	2,4%
108	desv.estándar tercer triestímulo	de grupo	42,3%	3,5%
109	media espectroide central (relativo)	de grupo	23,7%	50,1%
110	desv.estánd. espectroide central (rel.)	de grupo	41,3%	5,9%
111	media espectroide central (absoluto)	de grupo	35,8%	19,6%
112	desv.estánd. espectroide central (absl.)	de grupo	43,1%	1,4%
113	extensión espectral	estático	41,9%	4,4%
114	media extensión espectral	de grupo	34,2%	23,7%
115	desv.estándar extensión espectral	de grupo	43,2%	1,0%
116	pendiente espectral	estático	41,5%	5,5%
117	media pendiente espectral	de grupo	31,1%	31,5%
118	desv.estándar pendiente espectral	de grupo	42,9%	1,8%
119	error ajuste pendiente espectral	estático	40,9%	6,8%
120	media error ajuste pendiente espectral	de grupo	32,2%	28,8%
121	flujo espectral	estático	43,5%	0,2%
122	media flujo espectral	de grupo	41,8%	4,5%
123	desv.estándar flujo espectral	de grupo	42,5%	2,9%
124	ratio energía impares/pares	estático	43,6%	0,0%
125	media ratio energía impares/pares	de grupo	43,6%	0,0%
126	desv.estánd. ratio energía impares/pares	de grupo	43,6%	0,0%
127	pendiente amplitud región local	dinámico	42,9%	1,9%
128	error pendiente amplitud región local	dinámico	42,4%	3,2%

Tabla 9. Componentes del vector de características preliminar y su importancia

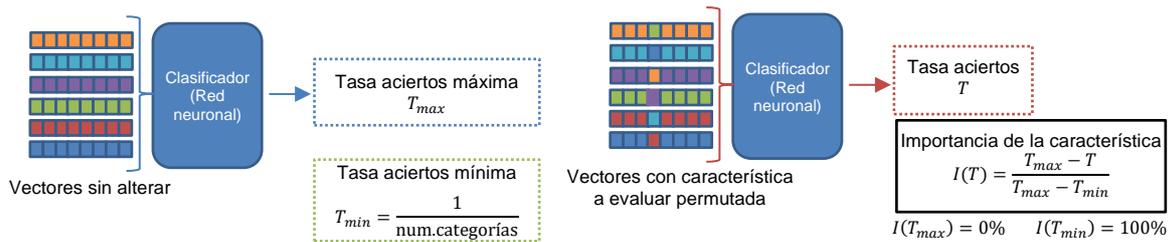


Figura 17. Método de cálculo de la importancia de las características

Atendiendo a las observaciones anteriores, se procede a definir el vector de características que se utilizará en la implementación definitiva del sistema. Por una parte, se eliminan las siguientes características.

- Todas las dinámicas excepto la pendiente de la amplitud en la región local y su error.
- La amplitud estimada de la nota y la amplitud relativa de los harmónicos 14<sup>o</sup> y 15<sup>o</sup>.
- Todas las otras características con una importancia inferior al 1%.

Estos vectores eliminados se han sombreado en gris en la Tabla 9. Por otro lado, se añaden las siguientes nuevas características las cuales se deduce que pueden facilitar información útil:

- Coeficientes MFCC del 14 al 20.
- Media estadística de la amplitud para los harmónicos 2<sup>o</sup> a 8<sup>o</sup>.

La lista definitiva de vectores se puede observar en la Tabla 10. Su cálculo se ha programado en las funciones `análisis_estatico` y `análisis_grupal`. Con esta nueva configuración, se vuelven a extraer los vectores del corpus de archivos de entrenamiento y se obtienen nuevamente las mediciones de la importancia de cada elemento, cuyo resultado se refleja en la misma tabla. A continuación, se comentan los métodos de cálculo de las características del vector definitivo:

- La **frecuencia fundamental**  $f_0$  y las amplitudes relativas de los harmónicos  $a_1, a_2, \dots, a_{15}$  se obtienen por los métodos explicados en el apartado 4.5. Las **amplitudes relativas** obtienen dividiendo por la **suma total del valor de los harmónicos** 1<sup>o</sup> al 15<sup>o</sup>. Esta suma conforma una característica adicional.
- El **crest factor** se define como la relación entre el pico de la señal en el tiempo y su valor eficaz (amplitud RMS)  $CF = \frac{x_{max}}{x_{rms}}$ . Estos dos valores dependen del desfase de los distintos componentes y dado que se trabaja con el módulo de la FFT, solo es posible obtener una aproximación de dicha magnitud. Como valor máximo de la señal se utiliza el que se obtendría si todos los harmónicos tuvieran el mismo desfase:  $x_{max} \approx |a_1| + |a_2| + \dots + |a_N|$ . Por otro lado, considerando que la fase de cada harmónico es una variable aleatoria uniforme, si se calcula la esperanza del cuadrado de la señal temporal se obtiene  $E_{t,\varphi}[y(t)^2] = \frac{a_1^2}{2} + \frac{a_2^2}{2} + \dots + \frac{a_N^2}{2}$ . Así, la amplitud RMS se puede aproximar como

$$x_{rms} = \sqrt{\frac{a_1^2}{2} + \frac{a_2^2}{2} + \dots + \frac{a_N^2}{2}}. \text{ De ahí, se calcula el } \textit{crest factor} \text{ aproximado como } CF \approx \sqrt{2} \cdot \frac{|a_1| + |a_2| + \dots + |a_N|}{\sqrt{a_1^2 + a_2^2 + \dots + a_N^2}}$$

- El **centroide espectral** se define como la frecuencia que constituye el centro de masas del espectro.

$$SC = \frac{\sum_{k=1}^N f[k] \cdot X[k]}{\sum_{k=1}^N X[k]}$$

En aquellas características que utilicen la versión relativa, se divide el resultado por  $f_0$ .

- Los coeficientes **triestímulo** son un concepto inspirado en la percepción visual RGB<sup>40</sup> y que tienen la función de caracterizar el timbre a partir de tres valores que corresponden a la suma de varias amplitudes relativas de los harmónicos.

$$T_1 = \frac{a_1}{\sum_{n=1}^{15} a_n} \quad T_2 = \frac{a_2 + a_3 + a_4}{\sum_{n=1}^{15} a_n} \quad T_3 = \frac{\sum_{n=5}^{15} a_n}{\sum_{n=1}^{15} a_n}$$

<sup>40</sup> (Peeters, 2004)

- o La **extensión espectral** se define como la media de la distancia de cada armónico al centroide espectral ponderada por la amplitud de cada armónico.

$$SS = \sqrt{\frac{\sum_{n=1}^{15} a_n \cdot (n - SC)^2}{\sum_{n=1}^{15} a_n}}$$

- o La **pendiente espectral** es la que corresponde a la recta de regresión que mejor se ajusta a las amplitudes de los armónicos y se obtiene por los métodos estadísticos de minimización de error cuadrático. Este error conforma una característica adicional.
- o El **flujo espectral** mide la variación del espectro entre dos segmentos consecutivos y se calcula como  $SF[t] = 1 - \frac{\sum_{k=1}^{15} a_k[t-1] \cdot a_k[t]}{\sqrt{\sum_{k=1}^{15} (a_k[t-1])^2 \cdot \sum_{k=1}^{15} (a_k[t])^2}}$ . De forma que si el espectro no cambia mucho entre ventanas consecutivas el valor de  $SF$  tiende a cero, y si tiene mucha variación tiende a uno<sup>41</sup>. Como la fórmula anterior no se puede calcular para el primer vector de un grupo, se igualará en ese caso al valor obtenido para el segundo vector.
- o La **pendiente de la amplitud en la región local** es la pendiente de la recta de regresión que modela la amplitud estimada de la nota en el entorno específico de 15 ventanas alrededor de la posición del vector. De esta forma se obtiene información de como es la tendencia de la amplitud de la señal en el tiempo y, por lo tanto, la posible etapa de ejecución de una nota. En el inicio de la misma, esta pendiente tenderá a ser positiva, y en el final, negativa. El cálculo se realizará a partir de las 7 muestras anteriores, la actual y las 7 posteriores. Cuando no se tengan muestras suficientes para este cálculo se reducirá al rango disponible.
- o Los **coeficientes MFCC** se han explicado en el Estado del arte. Para su cálculo se utilizan los métodos detallados en la bibliografía<sup>42</sup>.

<sup>41</sup> (Peeters, 2004)

<sup>42</sup> (Sreenivasa Rao & Vuppala, 2014) y (Standard ETSI ES 201 108 V1.1.3, 2013)

#	Característica	Tipo	Aciertos al permutar	Importancia	#	Característica	Tipo	Aciertos al permutar	Importancia
1	frecuencia fundamental	estático	37,9%	16,1%	36	coeficiente MFCC 14	estático	39,8%	11,5%
2	amplitud relativa harmónico 1	estático	39,3%	12,8%	37	coeficiente MFCC 15	estático	41,7%	6,9%
3	amplitud relativa harmónico 2	estático	42,9%	3,8%	38	coeficiente MFCC 16	estático	40,8%	9,1%
4	amplitud relativa harmónico 3	estático	44,1%	0,8%	39	coeficiente MFCC 17	estático	41,3%	7,7%
5	amplitud relativa harmónico 4	estático	44,1%	0,8%	40	coeficiente MFCC 18	estático	42,2%	5,4%
6	amplitud relativa harmónico 5	estático	43,9%	1,3%	41	coeficiente MFCC 19	estático	41,5%	7,3%
7	amplitud relativa harmónico 6	estático	43,7%	1,9%	42	coeficiente MFCC 20	estático	42,9%	3,7%
8	amplitud relativa harmónico 7	estático	44,4%	0,1%	43	desv.estándar frecuencia fundamental	de grupo	43,5%	2,3%
9	amplitud relativa harmónico 8	estático	43,3%	2,7%	44	media amplitud relativa harmónico 1	de grupo	32,2%	30,3%
10	amplitud relativa harmónico 9	estático	44,2%	0,6%	45	media amplitud relativa harmónico 2	de grupo	37,7%	16,7%
11	amplitud relativa harmónico 10	estático	43,9%	1,3%	46	media amplitud relativa harmónico 3	de grupo	40,6%	9,4%
12	amplitud relativa harmónico 11	estático	44,0%	1,2%	47	media amplitud relativa harmónico 4	de grupo	41,4%	7,6%
13	amplitud relativa harmónico 12	estático	42,1%	5,8%	48	media amplitud relativa harmónico 5	de grupo	42,2%	5,6%
14	amplitud relativa harmónico 13	estático	44,1%	0,9%	49	media amplitud relativa harmónico 6	de grupo	42,2%	5,7%
15	suma total amplitudes harmónicos	estático	42,7%	4,4%	50	media amplitud relativa harmónico 7	de grupo	43,3%	2,8%
16	crest factor	estático	38,8%	13,9%	51	media amplitud relativa harmónico 8	de grupo	41,6%	7,1%
17	centroide espectral (relativo)	estático	35,3%	22,5%	52	media amplitud relativa harmónico 9	de grupo	43,7%	1,8%
18	segundo triestímulo	estático	44,1%	0,8%	53	media segundo triestímulo	de grupo	35,4%	22,2%
19	tercer triestímulo	estático	42,3%	5,4%	54	media tercer triestímulo	de grupo	31,7%	31,4%
20	extensión espectral	estático	43,1%	3,3%	55	desv.estándar primer triestímulo	de grupo	41,9%	6,4%
21	pendiente espectral	estático	36,5%	19,7%	56	desv.estándar segundo triestímulo	de grupo	43,7%	1,8%
22	error ajuste pendiente espectral	estático	41,0%	8,6%	57	desv.estándar tercer triestímulo	de grupo	43,5%	2,4%
23	coeficiente MFCC 1	estático	19,7%	61,0%	58	media espectroide central (relativo)	de grupo	19,6%	61,1%
24	coeficiente MFCC 2	estático	29,4%	37,0%	59	desv.estánd. espectroide central (rel.)	de grupo	40,9%	8,8%
25	coeficiente MFCC 3	estático	29,5%	36,8%	60	media espectroide central (absoluto)	de grupo	31,4%	32,1%
26	coeficiente MFCC 4	estático	37,4%	17,4%	61	desv.estánd. espectroide central (absl.)	de grupo	43,5%	2,3%
27	coeficiente MFCC 5	estático	33,1%	28,1%	62	media extensión espectral	de grupo	34,6%	24,2%
28	coeficiente MFCC 6	estático	35,8%	21,3%	63	desv.estándar extensión espectral	de grupo	43,9%	1,3%
29	coeficiente MFCC 7	estático	36,5%	19,5%	64	media pendiente espectral	de grupo	29,6%	36,6%
30	coeficiente MFCC 8	estático	40,8%	9,1%	65	desv.estándar pendiente espectral	de grupo	42,6%	4,7%
31	coeficiente MFCC 9	estático	38,9%	13,7%	66	media error ajuste pendiente espectral	de grupo	33,8%	26,2%
32	coeficiente MFCC 10	estático	38,6%	14,5%	67	media flujo espectral	de grupo	43,3%	2,9%
33	coeficiente MFCC 11	estático	38,3%	15,0%	68	desv.estándar flujo espectral	de grupo	43,6%	2,0%
34	coeficiente MFCC 12	estático	38,2%	15,4%	69	pendiente amplitud región local	dinámico	43,9%	1,5%
35	coeficiente MFCC 13	estático	38,4%	14,9%	70	error pendiente amplitud región local	dinámico	43,3%	2,8%

Tabla 10. Componentes del vector de características definitivo y su importancia

## 4.7. Clasificación

El objetivo de las pruebas realizadas en esta etapa es encontrar el algoritmo de *machine learning* que mejor capacidad tenga para clasificar los vectores de archivos que no hayan formado parte del entrenamiento del sistema. Los archivos que se utilizan en el entrenamiento obtienen unas tasas de acierto muy elevadas ya que el clasificador los conoce, tal como se puede ver en las pruebas realizadas para la elección del filtro de énfasis (Tabla 3). Sin embargo, los aciertos en la clasificación de vectores se reducen significativamente cuando éstos proceden de otros archivos. La capacidad de un clasificador para categorizar correctamente nuevos datos se llama **generalización**, y para optimizar los algoritmos bajo esa premisa, en los entrenamientos se utilizará el método de validación cruzada<sup>43</sup>.

Este método consiste en dividir el corpus de entrenamiento en varios subconjuntos y excluir uno de ellos en el entrenamiento de los clasificadores. Una vez entrenado, se calcula la tasa de aciertos utilizando el grupo excluido, de forma que los datos de prueba son totalmente desconocidos por el sistema obteniendo así unos resultados más similares a los que obtendrían los usuarios finales. Para las pruebas que se van a realizar, se han dividido los 130 archivos del corpus de audios de entrenamiento (Anexo A) en 5 grupos de 26 archivos, tal como se muestra en la Tabla 11.

<sup>43</sup> *Cross-validation* en inglés.

		1	2	3	4	5	6
Violín	VN	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Viola	VA	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Violoncelo	VC	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Contrabajo	CB	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Arpa	AR	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Piano	PI	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Guitarra	GT	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Ukelele	UK	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Clavicémbalo	CV	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Flautín	FN	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Fl.Travesera	FT	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Flauta Dulce	FD	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Oboé	OB	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Corno Inglés	CI	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Clarinete	CL	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Clarinete Bajo	CJ	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Fagot	FA	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Contrafagot	CF	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Trompeta	TP	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Trompa	TA	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Trombón	TN	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Tuba	TB	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 5
Xilófono	XI	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	-
Acordeón	AC	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Órgano	OR	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-
Saxofón	SX	Grupo 1	Grupo 2	Grupo 3	Grupo 4	-	-

Tabla 11. División del corpus de entrenamiento en 5 grupos de *cross-validation*

Dado que los entrenamientos ocupan mucho tiempo de CPU es necesario hacer una selección inicial entre la gran variedad de algoritmos de *machine learning* que proporciona MATLAB. En la primera prueba se procede a entrenar 17 algoritmos distintos con sus parámetros de configuración por defecto. Para el entrenamiento se utilizan los grupos 2, 3, 4 y 5 con una estratificación de 400 vectores por categoría; y para obtener las tasas de aciertos en vectores y archivos<sup>44</sup> se evalúa la totalidad de vectores del grupo 1. Los resultados obtenidos son:

Sistema <i>machine learning</i>	Tasa de aciertos vectores (%)	Tasa de aciertos archivos (total: 26)
<i>Tree (Fine)</i>	28,5%	13
<i>Tree (Medium)</i>	20,2%	10
<i>Linear Discriminant</i>	33,2%	17
<i>Gaussian Naive Bayes</i>	26,2%	12
<i>SVM Linear</i>	<b>37,4%</b>	20
<i>SVM Quadratic</i>	<b>42,0%</b>	21
<i>SVM Cubic</i>	<b>41,6%</b>	19
<i>KNN Fine</i>	30,8%	17
<i>KNN Medium</i>	32,5%	18
<i>KNN Coarse</i>	30,6%	17
<i>KNN Cosine</i>	33,5%	19
<i>KNN Weighted</i>	33,9%	19
<i>Boosted Trees</i>	22,1%	10
<i>Random Forest</i>	<b>35,6%</b>	20
<i>Subspace Discriminant</i>	33,4%	18
<i>RUSBoosted Trees</i>	20,2%	10
<i>Red neuronal</i> <sup>45</sup>	<b>37,0%</b>	19

Tabla 12. Tasas de acierto de los algoritmos *machine learning* (primera selección)

A partir de esos resultados, se descartan todos aquellos algoritmos que no hayan sido capaces de clasificar un mínimo del 35% de los vectores de los audios no entrenados. Así pues, se descartan todos los algoritmos

<sup>44</sup> Tal como se ha comentado, para categorizar un archivo mono-instrumental se selecciona la categoría más abundante que se obtiene al clasificar sus vectores de características.

<sup>45</sup> Para la red neuronal se utiliza una configuración de capa única con 60 neuronas.

de clasificación excepto los siguientes: SVM de Kernel lineal, cuadrático y cúbico, Random Forest y Red neuronal.

La siguiente prueba se realiza con el objetivo de observar como evolucionan las tasas de aciertos con los métodos seleccionados al variar el número de vectores por categoría utilizados en el entrenamiento. Se procede a comprobar la capacidad de generalización de los algoritmos utilizando 100, 200, 500 y 1000 vectores por categoría y las configuraciones de hiperparámetros por defecto. Los resultados son:

Vectores/categoría	100		200		500		1000	
	vectores (%)	archivos (sobre 26)						
<b>Sistema <i>machine learning</i></b>								
<i>SVM Linear</i>	33,1%	16	35,4%	17	36,5%	17	38,2%	18
<i>SVM Quadratic</i>	36,4%	18	37,7%	19	42,3%	19	43,3%	20
<i>SVM Cubic</i>	35,5%	19	37,0%	20	41,6%	18	43,0%	20
<i>Random Forest</i>	31,8%	18	34,5%	18	36,6%	20	37,7%	19
<i>Red neuronal</i>	33,0%	18	35,9%	19	39,0%	18	40,6%	21

Tabla 13. Tasas de acierto de los algoritmos *machine learning* (segunda selección)

En general se observa que, a mayor número de vectores de entrenamiento utilizados, mejores tasas de acierto se consiguen. No obstante, una mejor tasa de aciertos en vectores no implica necesariamente siempre un mayor número de archivos acertados, a causa de la aleatoriedad intrínseca de algunos sistemas de *machine learning* como las redes neuronales. Por limitaciones en el tiempo de entrenamiento, no se hacen pruebas por encima de los 1000 vectores/categoría, aunque para el entrenamiento del clasificador final implantado en el sistema se utilizarán 3000 vectores/categoría.

Con esta segunda ronda de pruebas se procede a descartar aquellos algoritmos en los que no se ha logrado obtener un 40% de vectores acertados en alguna de las configuraciones probadas. Por lo tanto, la decisión final se realiza entre los sistemas SVM Cuadrático, SVM Cúbico y Red neuronal. Se procede a continuación a optimizar los hiperparámetros de cada uno de ellos con el objetivo de encontrar la configuración con la que mejor resultado se obtiene en cada método .

### Optimización de hiperparámetros en SVM Cuadrático

La aplicación *Classification Learner* de MATLAB dispone de los siguientes hiperparámetros para el entrenamiento del sistema *SVM Quadratic*:

- *Box Constraint Level*: con valores entre  $10^{-3}$  y  $10^3$ , indica el grado de penalización de cada error en la clasificación al trazar las fronteras entre categorías.
- *Kernel scale*: indica el factor de escalado en la función de transformación del clasificador.
- *Multiclass method*: indica si se diseña el clasificador de forma que compare las categorías entre pares (One-vs-One) o cada una de ellas contra el resto (One-vs-All).
- Estandarización de datos: redimensiona todas las características de entrenamiento para que queden situadas en el rango [-1, 1]

Se opta por dejar el *Kernel scale* en modo automático para que la propia aplicación lo optimice y la estandarización de datos se activa ya que al hacer pruebas con este parámetro desactivado el tiempo de entrenamiento era tan elevado que no se logró finalizar. Se procede a realizar varios entrenamientos del

clasificador, excluyendo de nuevo el grupo 1 de archivos para poder obtener las ratios de aciertos a partir de ellos. En cada experimento y de forma sistemática, se prueba una combinación distinta de los hiperparámetros obteniendo que la configuración óptima se consigue con *Box Constraint Level* = 1 y método multiclase *One-vs-All*. Con esta configuración optimizada se calculan las tasas de aciertos que se obtienen excluyendo cada uno de los grupos de *cross-validation* en el entrenamiento, así como la media global.

Grupo test (excluido de entrenamiento)	1	2	3	4	5	Total
Tasa de aciertos vectores	44,1%	40,1%	47,0%	43,1%	47,2%	<b>44,1%</b>
Tasa de aciertos archivos	22 (85%)	22 (85%)	24 (92%)	21 (81%)	21 (81%)	<b>110 (85%)</b>

Tabla 14. Tasas de aciertos con SVM Cuadrático optimizado

### Optimización de hiperparámetros en SVM Cúbico

Los hiperparámetros que dispone aplicación *Classification Learner* de MATLAB para SVM Cúbico son los mismos que en SVM Cuadrático. Al igual que en el caso anterior, se vuelve a fijar el *Kernel scale* en modo automático y la estandarización de datos activada. Repitiendo el mismo sistema de pruebas se obtiene una configuración óptima con *Box Constraint Level* = 0,1 y método multiclase *One-vs-All*. Con esta configuración óptima, las tasas de acierto obtenidas mediante *cross-validation* son:

Grupo test (excluido de entrenamiento)	1	2	3	4	5	Total
Tasa de aciertos vectores	44,5%	39,7%	46,2%	42,6%	46,8%	<b>43,8%</b>
Tasa de aciertos archivos	22 (85%)	22 (85%)	24 (92%)	23 (88%)	22 (85%)	<b>113 (87%)</b>

Tabla 15. Tasas de aciertos con SVM Cúbico optimizado

### Optimización de hiperparámetros en Red neuronal.

Finalmente se procede a buscar la configuración de Red neuronal que mejor generaliza los datos. Para ello utilizando los métodos de entrenamiento y validación por defecto (*Scaled Conjugate Gradient* y *Cross-Entropy*) y, repitiendo el sistema de pruebas anteriores, se procede a probar diversas combinaciones de hiperparámetros. Los elementos configurables con MATLAB son:

- Número de capas ocultas de la red neuronal
- Número de neuronas en las capas ocultas
- Sistema de cálculo del error cometido por la red en la validación de los datos (*performance*) y que determina el momento en que debe dejarse de entrenar para optimizar la generalización.
- Factor de regularización (entre 0 y 1). Si es mayor que 0, el cálculo del error cometido se pondera entre los errores en la salida y la media cuadrática de los pesos de las conexiones neuronales según la proporción indicada. Esto hace que los pesos tiendan a reducirse y por lo tanto que la red generalice mejor.
- Método de normalización de los datos, para indicar si los valores de salida deben escalarse en un determinado rango. Hay 3 opciones: 'none' para no regularizar; 'standard' para normalizar las salidas al rango [-1, +1] y 'percent' para normalizar las salidas al rango [-0.5, 0.5]. Nótese que, en el sistema diseñado, las respuestas que se presentan a la Red neuronal están formadas por valores 0 para las categorías incorrectas y 1 para la correcta.

Repitiendo el mismo sistema de pruebas de los casos anteriores, los mejores resultados se obtienen con una única capa oculta de 80 neuronas, cálculo de rendimiento *sum squared error* (suma del error cuadrático), tasa de regularización 0,6 y sin normalización en las respuestas. Con esta configuración óptima, las tasas de acierto obtenidas mediante *cross-validation* son:

Grupo test (excluido de entrenamiento)	1	2	3	4	5	Total
Tasa de aciertos <b>vectores</b>	43,6%	36,7%	43,9%	39,3%	46,0%	<b>41,3%</b>
Tasa de aciertos <b>archivos</b>	21 (81%)	21 (81%)	23 (88%)	19 (73%)	22 (85%)	<b>106 (82%)</b>

Tabla 16. Tasas de aciertos con Red neuronal optimizada

Para elegir el clasificador que se utilizará en la implementación del sistema se procede a comparar las tasas de aciertos globales obtenidas con cada uno de los métodos optimizados. Además, para tener una visión más completa de la capacidad de generalización que se consigue con cada algoritmo, se obtiene las matrices de confusión mediante las cuales se evalúa la tasa de aciertos de cada categoría y a las que se asignan tres posibles valoraciones: “buena”, “regular” o “deficiente”. Se considera que una categoría tiene una tasa de aciertos “buena” si el instrumento con más predicciones corresponde al correcto y si la tasa de error más alta entre el resto de los instrumentos es inferior a un tercio de la tasa de aciertos. Como tasa de acierto “regular” se considerará si la categoría con más predicciones es la verdadera, pero tiene instrumentos con tasas de error superiores a 1/3 de la correcta, lo que implica cierta probabilidad de clasificar habitualmente los archivos de forma errónea. Por último, por tasa de acierto “deficiente” se entenderá cuando el instrumento con más predicciones no es el correcto por lo que la probabilidad de fallar en la clasificación de un archivo es mayor que la de acertar.

Comparando todas las medidas comentadas se obtiene:

Sistema <i>machine learning</i>	Aciertos vectores	Aciertos archivos	Valoración tasas aciertos por categorías		
			Buena	Regular	Deficiente
SVM Cuadrático	44,1%	110 (85%)	19	4	3
SVM Cúbico	43,8%	113 (87%)	20	3	3
Red neuronal	41,3%	106 (82%)	17	5	4

Tabla 17. Comparación de los tres sistemas *machine learning* optimizados

Así pues, se observa que el sistema que generaliza mejor es SVM Cúbico ya que consigue el mayor número de aciertos en archivos desconocidos (objetivo principal del sistema) así como un mejor número de categorías con buena tasa de acierto (20 de 26). Por lo tanto, el algoritmo de *machine learning* que se implanta en el sistema tiene las siguientes características:

- **Algoritmo:** *SVM Cubic*
- **Box Constraint Level:** 0,1
- **Kernel scale:** automática
- **Método multiclase:** One-vs-All
- **Estandarización de datos:** activada

Finalmente, se procede a entrenar este clasificador con una estratificación de 3000 vectores por categoría, elegidos aleatoriamente entre todos los que produce el corpus de archivos de entrenamiento (Anexo A).

## 5. Implementación

### 5.1. Programación de una MATLAB App con interfaz gráfica para pruebas

Para poder testear el diseño propuesto, así como el clasificador entrenado se ha diseñado una interfaz gráfica ejecutable como una aplicación del entorno MATLAB. El código de esta APP, desarrollada con el soporte de la funcionalidad *APP Designer* de esta plataforma, se puede encontrar en el Anexo D.

Mediante esta aplicación no solo se podrá obtener la respuesta del clasificador a cualquier archivo en formato WAVE (.wav) para el número de instrumentos indicado, sino que se mostrará gráficamente como se han distribuido los vectores de características extraídos entre las 26 categorías disponibles.

### 5.2. Requisitos de instalación

Para poder instalar la aplicación se necesita un equipo con el sistema MATLAB instalado, incluyendo los complementos *Statistics and Machine Learning Toolbox* y *Signal Processing Toolbox*.

Esta aplicación se ha desarrollado y probado satisfactoriamente utilizando la versión R2020a de MATLAB. También se ha probado con la versión R2018b observando pequeñas deficiencias al mostrar caracteres con tildes y alguna diferencia en la presentación de la interfaz gráfica, pero manteniendo completamente la funcionalidad de la aplicación.

### 5.3. Instrucciones de instalación

La instalación se inicia al ejecutar el archivo “Clasificador de instrumentos.mlappinstal”, el cual se incluye en los entregables del trabajo, y siguiendo a continuación las instrucciones del instalador desarrollado por MATLAB.



Figura 18. Archivo ejecutable del instalador de la MATLAB App con que se ha implementado el sistema

Se abrirá el entorno MATLAB (si no estuviera ya en ejecución) y se mostrará una ventana preguntando si se desea instalar la aplicación en *My Apps*, debiendo seleccionar la opción “*Install*”. Una vez realizado este paso, se ejecuta la instalación y, al finalizar, ya aparece la aplicación en la pestaña APPS de MATLAB, y cuyo nombre es “Clasificador de instrumentos”.



Figura 19. Icono de inicio de la MATLAB APP desarrollada una vez instalada

## 6. Demostración y resultados

### 6.1. Instrucciones y ejemplo de uso del sistema

El aspecto de la interfaz gráfica es el siguiente:

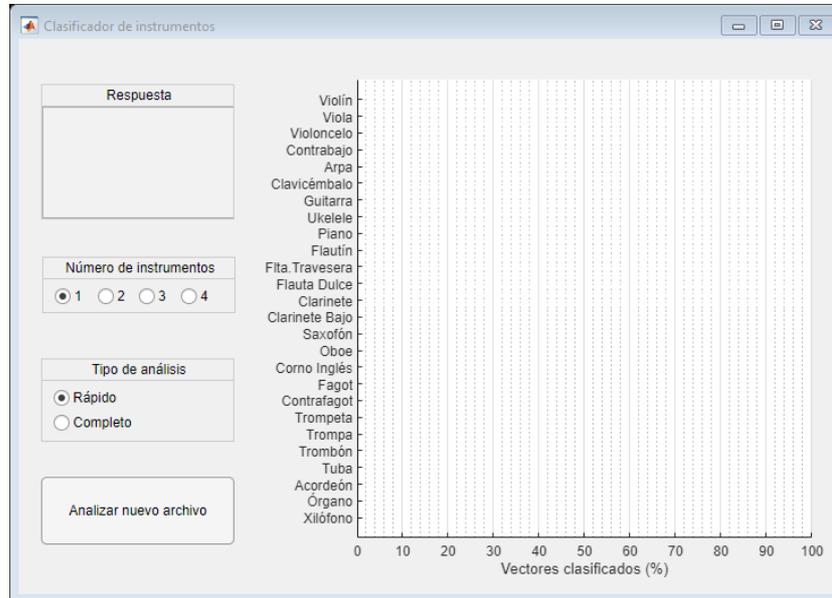


Figura 20. Interfaz gráfica de la APP desarrollada

Consta de los siguientes elementos:

- Ventana de presentación de respuesta
- Selector de número de archivos a identificar
- Selector de tipo de análisis
- Botón de selección y análisis de un nuevo archivo
- Gráfico con el porcentaje de vectores clasificados en cada categoría.

Para analizar una señal de sonido, el primer paso es pulsar el botón “Analizar nuevo archivo” y seleccionar el archivo .wav que se desea clasificar:

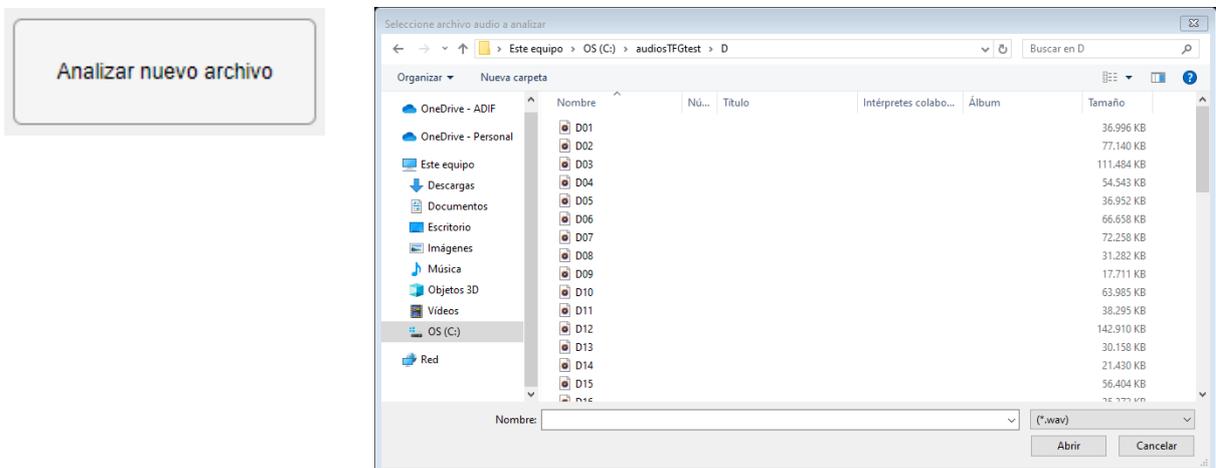


Figura 21. Botón e interfaz de selección de archivos .wav

Una vez seleccionado el archivo, el programa inicia el proceso de extracción de los vectores de características, y al terminar, comienza el proceso de clasificación de los mismos. La evolución de ambos procesos se muestra mediante sendas barras de progreso.



Figura 22. Barras de progreso durante el análisis de un archivo

Una vez terminada la clasificación de los vectores se muestra en el gráfico la distribución de los mismos en las distintas categorías asignadas. En la ventana de respuesta, se muestra la clasificación global del archivo, consistente en las categorías con más vectores clasificados.

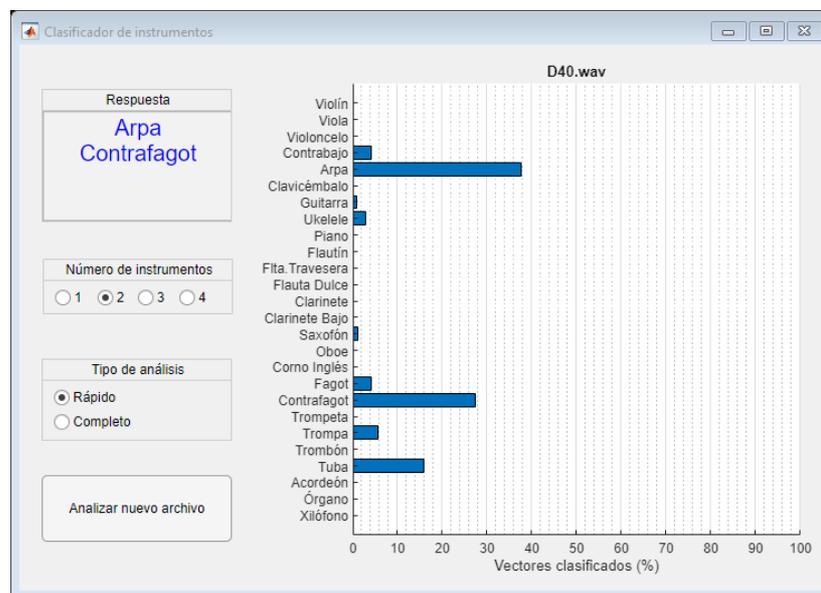


Figura 23. Interfaz gráfica mostrando los resultados de la clasificación de un archivo

Se ha incluido el selector “Número de instrumentos” para que el usuario pueda indicar cuantos instrumentos es necesario identificar en una señal que contiene varios de ellos. El valor indicado se puede modificar también con el audio ya clasificado, de forma que se actualiza la respuesta ofrecida.

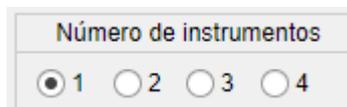


Figura 24. Selector del número de instrumentos a detectar

## 6.2. Modo de análisis rápido

Dado que para audios largos el proceso de extracción de vectores de características puede prolongarse durante varios minutos, se ha dispuesto el cuadro de selección “Tipo de análisis” para posibilitar la opción de hacer un análisis más rápido.

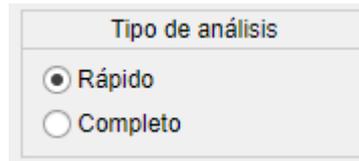


Figura 25. Selector de tipo de análisis

Si está seleccionada la opción "Rápido" el sistema procede a reducir el análisis de la señal a solo 60 segundos de sonido mediante la extracción de 10 extractos de 6 segundos distribuidos a lo largo del mismo. De esta forma el análisis y clasificación dura unos pocos segundos. En cambio, si se selecciona la opción "Completo" el sistema procede a extraer y clasificar los vectores de características del archivo entero. Se puede cambiar de un análisis a otro sin necesidad de volver a seleccionar el mismo archivo ya que al cambiar de opción el sistema pregunta si se quiere aplicar el nuevo tipo de análisis al archivo actual.



Figura 26. Cuadro de diálogo de cambio de tipo de análisis

### 6.3. Medidas de desempeño del sistema

A continuación, se obtienen diversas mediciones del desempeño del sistema diseñado. Para ello, se han utilizado como prueba 370 nuevos archivos distintos a los de entrenamiento y que se encuentran indicados en el Anexo B junto con la respuesta que ofrece el sistema para cada uno de ellos. La matriz de confusión y las tasas de sensibilidad y especificidad se refiere a vectores extraídos de audios de un solo instrumento, condición necesaria para conocer con exactitud la categoría correcta de cada vector.

#### 6.3.1. Tasas de aciertos de categorías en señales con un instrumento

Se prueba el sistema con 260 archivos (10 por categoría) conteniendo un único instrumento, de los cuales la detección se realiza correctamente en 232 casos, lo que supone una tasa de aciertos del 89%. Es interesante destacar que 20 de los 28 errores cometidos se concentran en las categorías de viola, guitarra y contrabajo, por lo que la tasa de aciertos excluyendo estas categorías es del 97%.

Señales con 1 instrumento	89%
Señales con 1 instrumento (excluyendo viola, guitarra y saxofón)	97%

Tabla 18. Tasas de aciertos en instrumentos detectados para señales con un único instrumento

### 6.3.2. Tasas de aciertos de categorías en señales con varios instrumentos

Dado que el sistema no tiene buen rendimiento con las categorías de viola, guitarra y saxofón, estas se excluyen para las pruebas con archivos de varios instrumentos ya que pueden distorsionar la observación de la reducción en la tasa de aciertos debido a la superposición de los mismos. Para estas pruebas se analizan 50 audios de 2 instrumentos, 35 de 3 instrumentos y 25 de 4 instrumentos. La tasa de aciertos en las predicciones de instrumentos incluidos en ellos es de:

Señales con 2 instrumentos (excluyendo viola, guitarra y saxofón)	63%
Señales con 3 instrumentos (excluyendo viola, guitarra y saxofón)	59%
Señales con 4 instrumentos (excluyendo viola, guitarra y saxofón)	56%

Tabla 19. Tasas de aciertos en instrumentos detectados para señales con varios instrumentos simultáneos

### 6.3.3. Matriz de confusión del sistema por vectores extraídos

La matriz de confusión muestra la distribución de los diferentes vectores en función de su categoría real y la categoría asignada por el clasificador. Dado que en el cálculo se han utilizado 260.000 vectores, los datos se muestran en forma de porcentaje respecto al total de vectores reales por categoría.

		Categoría real																									
		VN	VA	VC	CB	AR	CV	GT	UK	PI	FN	FT	FD	CL	CJ	SX	OB	CI	FA	CF	TP	TA	TN	TB	AC	OR	XI
Categoría asignada (Predicción)	VN	57	20	7	0	3	4	4	2	0	1	0	1	4	1	12	1	1	0	0	2	0	1	0	8	3	0
	VA	7	21	9	1	2	1	3	2	1	0	0	0	2	1	9	1	1	0	0	1	1	1	0	9	2	0
	VC	1	10	28	9	2	2	11	1	4	0	0	0	0	2	6	0	0	0	3	0	0	1	0	2	3	0
	CB	1	3	6	46	4	1	10	1	5	0	0	1	0	2	2	0	0	0	2	0	0	1	1	1	2	0
	AR	1	1	3	6	34	0	7	11	6	0	2	0	2	2	0	0	0	1	6	0	1	0	4	1	4	1
	CV	1	1	0	0	0	69	4	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	2	0	0
	GT	1	1	4	3	6	1	17	4	5	0	0	0	0	1	2	0	0	0	1	0	0	1	1	3	4	0
	UK	2	4	1	2	11	2	15	51	2	0	0	2	2	1	4	0	0	1	1	1	0	1	0	3	3	1
	PI	1	4	4	5	7	2	5	2	37	0	2	0	2	2	2	0	1	1	2	0	2	2	0	5	12	0
	FN	4	1	1	1	0	0	0	0	0	77	5	2	1	1	1	8	1	0	0	2	0	0	0	1	0	4
	FT	2	2	2	2	2	0	0	1	2	10	43	14	7	3	1	3	1	0	0	4	1	1	0	1	0	5
	FD	1	0	1	0	2	1	0	1	0	3	6	70	1	1	1	0	1	1	0	1	1	1	1	1	0	5
	CL	3	3	2	1	3	0	1	3	2	1	12	1	55	6	2	5	3	0	1	4	1	1	0	2	1	2
	CJ	0	2	6	2	1	0	1	2	2	0	2	0	6	67	3	1	0	0	1	1	1	2	0	1	1	0
	SX	4	4	3	1	2	0	3	3	1	0	2	2	5	2	14	0	4	2	1	2	5	3	0	2	0	0
	OB	2	1	1	1	1	0	0	1	0	2	4	0	5	1	1	67	8	0	0	11	1	1	0	2	1	1
	CI	1	2	2	1	2	0	1	2	2	0	4	0	1	1	3	2	69	2	1	4	3	3	0	1	0	0
	FA	0	0	0	1	2	0	1	1	4	0	1	0	0	1	1	0	2	73	3	0	10	17	3	0	0	1
	CF	0	1	6	5	3	0	7	1	10	0	0	1	0	1	2	0	0	2	65	0	2	2	7	0	2	0
	TP	4	6	2	0	1	0	0	1	1	0	3	0	2	1	8	7	3	0	0	51	1	2	0	4	2	0
	TA	1	2	2	1	2	0	3	5	3	0	1	0	1	1	11	0	1	5	4	1	46	13	3	1	0	0
	TN	1	3	4	3	1	1	1	1	5	0	2	0	1	2	6	0	4	8	1	14	15	43	1	2	2	0
	TB	0	0	1	3	2	0	1	1	1	0	0	0	0	0	0	0	0	1	6	0	8	3	78	0	0	0
	AC	4	4	2	0	1	7	2	1	1	1	0	0	0	0	5	0	0	0	1	1	0	1	0	35	4	0
	OR	2	2	4	4	4	7	4	1	6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	10	56	0
	XI	0	0	0	1	1	0	0	1	0	4	6	3	1	0	0	1	0	0	0	0	0	0	0	0	0	77

Tabla 20. Matriz de confusión en vectores (valores en %)

Para todas las categorías, la tasa de acierto es superior a la de confusión con cualquier otro instrumento. No obstante, las tasas de acierto indicadas en amarillo corresponden a aquellos casos en que existen clasificaciones erróneas (marcadas en color gris) cuya tasa de confusión es superior a un tercio de la tasa de acierto y, por lo tanto, son proclives a proporcionar errores con frecuencia.

### 6.3.4. Tasas de sensibilidad y especificidad por vectores extraídos

A continuación, se muestran las tasas de sensibilidad y especificidad de las distintas categorías obtenidas a partir de la matriz de confusión anterior. Los datos se han ordenado de mayor a menor sensibilidad.

		Tasa de sensibilidad	Tasa de especificidad
TB	Tuba	78,1%	98,9%
FN	Flautín	77,2%	98,7%
XI	Xilófono	76,8%	99,2%
FA	Fagot	73,1%	98,1%
FD	Flauta Dulce	69,9%	98,8%
CV	Clavicémbalo	68,8%	99,5%
CI	Corno Inglés	68,8%	98,4%
CJ	Clarinete Bajo	67,0%	98,6%
OB	Oboe	66,9%	98,2%
CF	Contrafagot	65,1%	97,9%
VN	Violín	56,9%	96,9%
OR	Órgano	55,6%	98,1%
CL	Clarinete	55,3%	97,6%
UK	Ukelele	51,0%	97,7%
TP	Trompeta	50,6%	98,0%
TA	Trompa	46,3%	97,4%
CB	Contrabajo	46,2%	98,2%
FT	Fita.Travesera	43,1%	97,5%
TN	Trombón	42,5%	96,8%
PI	Piano	36,9%	97,4%
AC	Acordeón	35,4%	98,6%
AR	Arpa	34,3%	97,7%
VC	Violoncelo	27,5%	97,7%
VA	Viola	21,4%	97,8%
GT	Guitarra	16,9%	98,5%
SX	Saxofón	14,1%	97,9%

Tabla 21. Tasas de sensibilidad y especificidad en vectores extraídos

### 6.3.5. Respuesta del sistema a categorías desconocidas

Finalmente, a modo experimental, puede resultar interesante conocer como se comporta el sistema ante señales de sonido que contienen fuentes que no forman parte del conjunto de categorías entrenadas. Se analizan dos audios diferentes de cinco fuentes sonoras que el clasificador desconoce, obteniendo:

Fuente sonora	Respuesta del sistema
<b>Banjo</b>	Piano / Violín
<b>Armónica</b>	Violín / Clavicémbalo
<b>Flauta de pan</b>	Clarinete / Flauta travesera
<b>Canto de pájaros</b>	Flautín / Xilófono
<b>Soplido en botella</b>	Ukelele / Flauta dulce

Tabla 22. Respuesta del sistema a algunas fuentes desconocidas

Aunque para algunos casos se obtiene una respuesta con un timbre parecido, para muchos otros hay bastante diferencia perceptiva entre el timbre de la fuente sonora introducida y el de la respuesta del sistema. Por lo tanto, y a falta de un estudio más profundo al respecto, todo parece indicar que el mecanismo que utiliza el sistema para diferenciar los instrumentos es bastante diferente al que utiliza el sistema cognitivo humano a partir de la percepción auditiva del timbre. Una de las causas de esta diferencia podría originarse en la ausencia, en el diseño propuesto, de características obtenidas en el dominio temporal.

## 7. Conclusiones y líneas de futuro

### 7.1. Conclusiones

Aunque a priori la distinción de instrumentos musicales parece una tarea sencilla, la necesidad de recurrir al *machine learning*, así como el gran número de características incluidas en los vectores son una muestra de la complejidad del problema, que dista todavía de estar totalmente resuelto.

Los resultados conseguidos en la clasificación de archivos con un solo instrumento son satisfactorios ya que mejoran los obtenidos en algunas de las propuestas publicadas que han servido de punto de partida para idear el sistema. No obstante, en algunas categorías, como el saxofón y la guitarra, las tasas de sensibilidad son bastante deficientes. Sería interesante analizar si el rendimiento de estas categorías se mantiene entrenando el sistema con otro conjunto de señales distinto y así poder determinar si los puntos débiles son fruto de un diseño incompleto o de una selección del corpus poco exhaustiva.

Por otro lado, la tasa de aciertos en archivos de más de un instrumento no es suficiente como para desarrollar una aplicación comercial. En estos casos, las tasas de instrumentos detectados son aproximadamente del 60%. Aún así, el acierto en algunos casos de la totalidad de instrumentos presentes en archivos con 3 y 4 categorías simultáneas indica que el sistema tiene potencial para realizar esta funcionalidad y que sería interesante explorar las acciones de mejora que se proponen en el apartado siguiente.

En el seguimiento de la planificación se ha seguido por lo general los plazos especificados, no obstante, se han encontrado varios problemas que han prolongado algunos de los tiempos estimados:

- La obtención del corpus de audios, tanto de entrenamiento como de test, ha sido complicada, especialmente para encontrar grabaciones de calidad para los instrumentos menos populares.
- Las tareas de entrenamiento y optimización de los algoritmos de *machine learning* se ha complicado por las grandes cantidades de tiempo que requerían algunos métodos, de forma que el sistema implementado se ha entrenado tan solo 78.000 vectores cuando el corpus proporcionaba cerca de 1,7 millones.

Finalmente, un aspecto que indirectamente se ha aprendido durante la realización de este trabajo ha consistido en la optimización del código. En las primeras fases de desarrollo, el tiempo necesario para extraer los vectores de un archivo quintuplicaba la duración de la señal contenida. Actualmente es inferior a un tercio lo que permitiría la utilización del sistema en tiempo real.

## 7.2. Líneas de futuro

A raíz de los resultados obtenidos y reflexionando sobre el diseño del sistema, se pueden plantear diversas líneas de actuación para mejorar los resultados o analizar en mayor profundidad sus prestaciones.

- La falta de mediciones de tipo temporal en el vector de características, dado que este se origina a partir de la FFT específica de las notas, ha privado la inclusión de muchos elementos con los que los humanos reconocemos los instrumentos. Se podría investigar la extracción de nuevas características mediante la reconstrucción temporal de la señal a partir de los armónicos de cada nota teniendo en cuenta su desfase.
- Se podrían desarrollar algunas técnicas que permitieran reducir el error que se produce al superponer armónicos de distintas notas. Dado que se conocen las frecuencias fundamentales, se puede deducir las posiciones de los armónicos que colisionan y a partir de ahí establecer un mecanismo de corrección.
- El entrenamiento del sistema se ha visto limitado por los tiempos de computación y la falta de grabaciones disponibles de alta calidad. Así, el corpus de entrenamiento solo contiene 20 minutos por instrumento, de los cuales solo un 4,5% de los vectores extraídos se han utilizado finalmente. A lo largo de las pruebas realizadas, siempre se ha observado una mejora en los resultados al aumentar los datos de entrenamiento del clasificador, por lo que sería interesante explorar el potencial del sistema al utilizar un conjunto de datos de mayor tamaño y calidad.
- Para categorías con confusiones habituales (por ejemplo, violín y viola) se podría estudiar la inclusión de clasificadores secundarios entrenados específicamente para el tratamiento de los casos más problemáticos.
- Dado que la optimización realizada en el código de extracción de características permite obtener los vectores y clasificarlos en un tiempo inferior al que se necesita para reproducir el sonido; una interesante acción de mejora de este sistema consistiría en mostrar la respuesta del clasificador en tiempo real, al mismo tiempo que se reproduce o se graba el sonido analizado.

## Bibliografía

- Azarloo, A., & Farokhi, F. (2012). Automatic Musical Instrument Recognition Using K-NN and MLP Neural Networks. *2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks*, 289-294.
- Ball, P. (2011). *El instinto musical*. Madrid: Turner.
- Eronen, A. (2001). Comparison of features for musical instrument recognition. *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, 19-22.
- Eronen, A., & Klapuri, A. (2000). Musical instrument recognition using cepstral coefficients and temporal features. *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing . Proceedings (Cat. No.00CH37100)*, 2, 11753-11756.
- Ghosh, A., Pal, A., Sil, D., & Palit, S. (2018). Music Instrument Identification Based on a 2-D Representation. *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 509-513.
- González, R. (16 de Enero de 2017). *Cuál es la diferencia entre un violín y una viola*. (unCOMO) Recuperado el 1 de Marzo de 2020, de <https://educacion.uncomo.com/articulo/cual-es-la-diferencia-entre-un-violin-y-una-viola-22514.html>
- Kitahara, T., Goto, M., Komatani, K., Ogata, T., & Okuno, H. (2006). Instrogram: A New Musical Instrument Recognition Technique Without Using Onset Detection NOR F0 Estimation. *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, V-V.
- Klapuri, A. (1999). Pitch estimation using multiple independent time-frequency windows. *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. WASPAA'99 (Cat. No.99TH8452)*, 115-118.
- Lago Castro, P., Ponce de León Barranco, L., & Balo, M. (2014). Reconocimiento del timbre en la educación musical. *Eufonía: Didáctica de la música*(62), 73-81.
- Maersch, K., Rohde, U., Seiffert, O., & Singer, U. (1994). *Atlas de los instrumentos musicales*. Madrid: Alianza Atlas.
- Maliki, I., & Sofiyanudin. (2018). Musical Instrument Recognition using Mel-Frequency Cepstral Coefficients and Learning Vector Quantization. *IOP Conference Series: Materials Science and Engineering*, 407.
- Moore, H. (2018). *MATLAB for Engineers*. Pearson.
- Oppenheim, A., & Schafer, R. (2004). From frequency to quefrequency: a history of the cepstrum. *IEEE Signal Processing Magazine*, 21(5), 95-106.
- Oppenheim, A., & Willsky, A. (1998). *Señales y sistemas*. Pearson Prentice Hall.
- Peeters, G. (2004). A large set of audio features for sound description (similarity and classification) in the CUIDADO project.
- Sreenivasa Rao, K., & Vuppala, A. (2014). *Speech Processing in Mobile Environments*. Cham, Switzerland: Springer.

Standard ETSI ES 201 108 V1.1.3. (2013). *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*. Francia.

Universidad ESAN. (16 de Octubre de 2015). *Marketing: ¿Cómo definir nuestro público objetivo?* Recuperado el 31 de Mayo de 2020, de <https://www.esan.edu.pe/apuntes-empresariales/2015/10/marketing-como-definir-publico-objetivo/>

Zamacois, J. (1973). *Teoría de la Música*. Barcelona: Labor.

Zlatintsi, A., & Maragos, P. (Abril de 2013). Multiscale Fractal Analysis of Musical Instrument Signals With Application to Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(4), 737-748.

# Anexos

## Anexo A: Corpus de audios de entrenamiento del sistema

En este apartado se muestra el listado de los videos online disponibles en la plataforma YouTube que se han consultado para la extracción de las señales de audio utilizadas en el entrenamiento del sistema. Se han seleccionado considerando los siguientes requisitos:

- El sonido incluido en el archivo debe proceder de un único instrumento. Este requisito es necesario para poder etiquetar correctamente y sin ambigüedades la categoría a la que pertenecen los vectores extraídos del mismo, ya que éstas corresponden a instrumentos individuales.
- Se requiere la certeza de que se trata de sonidos realizados con instrumentos reales y no sintetizadores, ya que se supone que estos últimos tienen un comportamiento más determinista y, por lo tanto, más fácil de clasificar. Los instrumentos reales aportarán una mayor variabilidad en los vectores de características que será útil para enriquecer el entrenamiento del clasificador.
- Para cada instrumento, se seleccionan como mínimo 4 grabaciones de distintos orígenes, para abarcar más variabilidad en las muestras en cuanto a técnicas de interpretación, repertorio y diferentes especificaciones de un mismo instrumento.
- La duración total de las grabaciones para cada categoría es entre 20 y 25 minutos para equilibrar el número de vectores obtenidos por instrumento.

Tabla 23. Corpus de audios de entrenamiento del sistema

Nombre archivo	Título del video	Dirección URL youtube.com /watch?v=	Autor / Usuario	Durac.
<b>VIOLÍN</b>				
VN1.wav	Hilary Hahn - J.S. Bach: Partita for Violin Solo No. 1 in B ...	iEBX_ouEw1l	Hilary Hahn	3:33
VN2.wav	N. Paganini Caprice no. 5   Sumina Studer	0jXXWBt5URw	Sumina Studer Violinist	2:45
VN3.wav	Incredible Wedding Violin Solo Music / Bride Entrance / Bridal...	9OZHbvWdBbQ	Amy Spaulding	5:29
VN4.wav	Victoria Daskalova - J.S.Bach - Grave from 2 <sup>nd</sup> sonata for solo...	T-Ab1a8qFIQ	Nikolay Daskalov	4:28
VN5.wav	Ray Chen plays Ysaye Sonata No. 2 Obsession (Prelude)	QmzavUR0Lfs	Ray Chen	3:05
VN6.wav	J. S. BACH: Violin Solo Sonata No.1 BWV 1001 'Adagio'	diAA2urTqkY	Sofiko Tchumburidze	4:02
<b>VIOLA</b>				
VA1.wav	B. Campagnoli Caprice No.3 for viola solo	c6lvfsuOkQY	Kin Fung Leung	1:24
VA2.wav	Campagnoli Caprice #1 for Viola	kuxkGGU8vXM	Kristina Giles	2:43
VA3.wav	Hoffmeister - Viola Etude no.4	Sq7rZogdQaQ	Obey thoven	4:55
VA4.wav	Paganini Caprice 24 for viola	sR4cND_-tMU	3foisC	5:40
VA5.wav	Henri Vieuxtemps, Capriccio for Viola solo (Roland Glassl, Viola)	5B3Yj2vJS3Y	Roland Glassl	3:02
VA6.wav	Suite No. 5 for Viola Solo in C Minor, BWV 1011: IV. Sarabande	FvrvIE5Z7cM	Nils Mönkemeyer - Tema	5:04
<b>VIOLONCELO</b>				
VC1.wav	Bach: Prélude, Cello suite Nr.1   Ophélie Gaillard	poCw2CCrfzA	Aparté Music	2:56
VC2.wav	Cello Suite No. 5 in C Minor BWV 1011: I. Prélude	AJTM5DA35jA	Paul Tortelier - Tema	7:42
VC3.wav	Cygne Saint Saens cello solo	kCrqpCEr0uk	Robin Dupuy	4:31
VC4.wav	Alfredo Ferre plays J.S.Bach - Suite for cello solo n°6 BWV....	3Pr5uw7BJ-M	Onomeya - Alfredo Ferre	5:33
VC5.wav	Thiago Wolf - Careless Whisper (George Michael) for cello solo	CXr08frD9zA	Thiago Wolf	3:47
<b>CONTRABAJO</b>				
CB1.wav	James Cole Double Bass: Played by Lorraine Campet	Ppnp2qd23X4	Thomas & George Martin Violin Makers	2:54
CB2.wav	Alexander Verster: Six Vignettes for Double Bass (2018)	q_tGHtKkHRM	Alexander Verster	13:42
CB3.wav	Alexander Muravyev "Cheshire Cat"   CLIP (double bass solo)	vhaezjuK428	Alexander Muravyev	2:47
CB4.wav	DOUBLE BASS SOLO: Suspíria	OPUaCJPGejQ	Domenico Botelho	3:11
<b>ARPA</b>				
AR1.wav	J.S. Bach - Toccata and Fugue in D Minor BWV 565 // Amy...	oPmKRtWta4E	Amy Turk	9:32
AR2.wav	Solo Harpist for Hire - "Libertango" by Piazzolla	uLUzVcCfI3A	nhmusicuk	2:13
AR3.wav	Franz Liszt - Un Sospiro - arranged for harp	B4qD30h6tdc	Valeria Kurbatova	6:11
AR4.wav	Noa Gabay, Harp solo from the opera "Lucia di Lammermoor"	wuSH0Bu_WKq	Noa Gabay	2:55
<b>CLAVICÉMBALO</b>				
CV1.wav	P. Royer: La Marche des Scythes - Angelica Selmo, ...	PsoixV362mM	Angelica Selmo	7:49
CV2.wav	Jean Rondeau records Rameau's Les Sauvages	KQibIb_kIT8	Warner Classics	1:50
CV3.wav	Haydn Harpsichord : Elaine Comparone plays Adagio, Sonata...	ahXYhPSebm4	Elaine Comparone	9:01
CV4.wav	J.S.Bach Sinfonia 2 in C Minor BWV 788 on Harpsichord	a-njwumjG2l	Kanji Daito 大藤莞爾	2:21
<b>GITARRA</b>				
GT1.wav	The Entertainer - Scott Joplin (Michael Lucarelli, classical guitar)	psyOPSrs0BI	Michael Lucarelli	4:41
GT2.wav	BOHEMIAN RHAPSODY MEETS CLASSICAL GUITAR	BZXO-rfkJ0g	Beyond The Guitar	5:56
GT3.wav	Peter Richardson - Solo Guitar	1MnYc0wgbdY	Sam Hill Entertainment	1:24
GT4.wav	Turina: Sonata for Guitar Solo, Op.61 - 2. Andante	_0t-wWe0zCs	Pepe Romero - Tema	3:42
GT5.wav	Liebstraum No. 3   Franz Liszt   NBN Guitar	MAA6uSDfZ30	joshua rogers	2:04
GT6.wav	Greensleeves (Classical Guitar Arrangement by Giuseppe....	5Prc4IGsGZQ	Giuseppe Torrisi	3:26
<b>UKELELE</b>				
UK1.wav	Yesterday - The Beatles (Tenor Ukulele Instrumental Cover)	jggDZqRtfHl	Jared DeMeester	2:24
UK2.wav	一期一会 -Ichigolchie- / Jake Shimabukuro をウクレレで...	7eG0Qd0YiNg	ukelele channel	4:44
UK3.wav	Gracias a la vida ukelele cover ♪ Violeta Parra instrumental	V0wm-wcy86A	Ukulele At Home	1:55
UK4.wav	Juice WRLD - Lucid Dreams Ukulele Instrumental Cover	RXX_PffPU8U	Randy Gapasin	2:56
UK5.wav	(Jason Mraz) I'm Yours - Sungha Jung (Ukulele)	E2OEuyF_2u8	Sungha Jung	4:24

Nombre archivo	Título del video	Dirección URL youtube.com /watch?v=	Autor / Usuario	Durac.
UK6 .wav	Waiting in Vain - Bob Marley (Ukulele Instrumental cover)	019QUKQUsVI	Paul Caca - Ukulele	4:49
PIANO				
PI1 .wav	Salut d'Amour - PIANO SOLO - Edward Elgar Op.12	snqAA5POwHY	Hazel Nguyen Pianist	3:04
PI2 .wav	Etincelles, Op.229 (arr. for piano solo), Émile Waldteufel	d_0ipm7utHU	Musopen	7:25
PI3 .wav	Waiting in the Water (The Isaacs) - Jared Roberts - Piano Solo	8x2dyCuScQg	Jared Roberts	4:10
PI4 .wav	Massenet: Meditation for Thaïs for piano (Andrew von ...	duDlzSgwEnl	Warner Classics	5:18
PI5 .wav	Playing love - Ennio Morricone - The Legend of 1900 - Piano ....	_qXTkGpcyJs	matteo trimigno	2:29
FLAUTÍN				
FN1 .wav	Daniel Dorff: TWEET for solo Piccolo flute	kc8WZbAWE1w	beoclip	4:42
FN2 .wav	Farndale Dances for Piccolo Solo: The stream	7Yz-9d1KM-E	Debora Rosti - Tema	2:18
FN3 .wav	Premier recueil de pièces: Gigue en rondeau	MJ8mzC9tpL8	Debora Rosti - Tema	3:58
FN4 .wav	Persichetti Parable XII for solo piccolo	Mzm_wnWki2k	Jessica L. Banks	3:04
FN5 .wav	Mimosa for piccolo solo	7FzqATPX18o	Hong-Da Chin	4:55
FN6 .wav	Piccolo Audition - Victory for Wahington + Bow Down to....	pi_8dAmr-mU	Bridgette Chen	2:27
FLAUTA TRAVESERA				
FT1 .wav	Elisabeth Wentland - Kuhlau fantasy for flute solo op. 38 D major	be1jJCH32OU	Elisabeth Wentland	3:38
FT2 .wav	Debussy: Syrinx for solo flute (Emmanuel Pahud)	RNjroFNI7mA	Warner Classics	3:21
FT3 .wav	Flute Solo] Paganini Caprice No.24 파가니니 카프리시 24 번...	kafePIAD—k	Flute Jasmine	5:17
FT4 .wav	Johann Sebastian Bach, Toccata and Fugue d minor BWV 565	8p0qo30LUfo	Helena Schulthess	9:36
FLAUTA DULCE				
FD1 .wav	Coral, for Recorder solo	vFFvpa4owhw	The Rosenberg Trio - Tema	5:54
FD2 .wav	Genevieve Lacey - Telemann Fantasia No. 1 (live from ....	BRkkRDQzjl	ABC Classic	3:38
FD3 .wav	Lucie Horsch - Le Printemps (Spring)	H3FucovSdsI	LucieHorschVEVO	3:50
FD4 .wav	Meditation for alto Recorder solo Hirose	brteU2YeuWc	Caoimhe de Paor	7:10
CLARINETE				
CL1 .wav	Sonata for Solo Clarinet: I. First Movement	9fjvCMYRxno	Phillip Rehfeldt - Tema	3:31
CL2 .wav	Kvandal: Fantasy For Solo Clarinet, Op 68 Nr 2	xmITsXYomG8	Fredrik Fors - Tema	6:01
CL3 .wav	Stravinsky: Three Pieces for Clarinet Solo – Volodymyr....	OxQXmZWq1Jg	hmtRostock	4:04
CL4 .wav	Sérgio Pires - G. Donizetti - Studio Primo	HqPJWvqyR00	Sérgio Pires	5:01
CL5 .wav	5 Pieces for Clarinet Alone: No. 4, Singing	a8M8JX8_fwK	Jose Daniel Cirigliano - Tema	2:39
CLARINETE BAJO				
CJ1 .wav	Paws - David Bennett Thomas, for Bass Clarinet solo	FGMuF_Xsaas	SeBass	6:30
CJ2 .wav	Solo for bass clarinet	8ojW4bd9v_g	John Rodgers - Tema	6:04
CJ3 .wav	Pascaliana I for bass clarinet solo	GsEGRHBkr2I	claribaixos	2:58
CJ4 .wav	Daniel Dorff - In a Deep Funk: III. Bear Hug	Z1eLXcMeUaA	Dalton Ringey	3:14
CJ5 .wav	Bach Cello Suite No. 1 Courante on Bass Clarinet	2k39KAph_6Q	Angelique	3:02
CJ6 .wav	F. Demnitz - Chord Study No. 1 in C Major (on Bass Clarinet)	kvFuQVPdEYc	Soundole Music	1:12
SAXOFÓN				
SX1 .wav	Pepito Ros-Tre Canti for solo Saxophone	PdFKSoJz9us	Pepito Ros	10:28
SX2 .wav	Japanese Improvisation for Solo Saxophone	Ddb108x9IPI	Sohre Rahbari - Tema	3:02
SX3 .wav	"Scarlet" for Solo Saxophone (Collevoxus, Florida State ....	czjMB_-b-UU	Cody Myre, Composition and Engraving	6:24
SX4 .wav	Sonata for Alto Saxophone Solo: III. Prestissimo	dutSPHfRDw	Various Artists - Topic	3:34
OBOE				
OB1 .wav	Persichetti - Parable for Solo Oboe	Rh8MmapKkq8	Stefanie Cohn	5:48
OB2 .wav	Six Metamorphoses after Ovid for Oboe Solo, Op. 49 - ....	J7YndQPDSWs	Vilém Veverka - Tema	2:58
OB3 .wav	G.F. Telemann Fantasie No.1 in A major : Alexander....	2_Lnw953Jng	Alexander Vvedenskiy	3:09
OB4 .wav	Six Methamorphoses after Ovid for Oboe Solo, Op. 49: VI....	T52_5iqXw4s	Thomas Indermühle - Tema	2:11
OB5 .wav	Silvestrini etude n.1 for oboe solo "Hotel des Roches noires a ...	MTjeQrd5jMw	Екатерина Легостаева	4:33
OB6 .wav	Sieben Studien (Capricci) für Oboe Solo - III. Lento	Nee340GJ13A	Various Artists - Topic	2:24
CORNO INGLÉS				
CI1 .wav	The Swan by Saint-Saens	i5AiccbVpZA	Shea Tully	2:00
CI2 .wav	Soliloquy for Solo English Horn	GTppVpyPyyE	Thomas Stacy - Tema	4:10
CI3 .wav	English Horn solo by Jiwon Park, Dvorak's Sym. No.9	gC5IBEG3Goo	JW Park	1:23
CI4 .wav	Luís Carvalho - "... pour le temps perdu" for solo English Horn	v7nyFz5Z1MY	Luís Matos	9:05
CI5 .wav	Alexis Mitchell: Shostakovich 8 EH Solo	wH5dvrTAqRs	Alexis Mitchell	3:24
FAGOT				
FA1 .wav	Allemande - Partita para fagot solo. BWV 1013 J.S.Bach - ...	Hb_UTQsnFqA	Céline Camarassa Castelló	3:08
FA2 .wav	Rhapsody for Solo Bassoon	guQI_OEt_FU	Christopher Lamb - Tema	5:51
FA3 .wav	Partita for solo Bassoon - Gordon Jacob	5wysrfqCa_g	Francesco Pio Russo	5:30
FA4 .wav	J. S. Bach, Partita BWV1013 (arranged for Bassoon Solo), I. ....	P6Pn_Pfg8w	Daria Fedorova	3:36
FA5 .wav	Sonatina for Solo Bassoon: II.	s308s_6rHoE	Rosie Burto - Tema	3:01
CONTRAFAGOT				
CF1 .wav	Baßnachtigall for Contrabassoon Solo, Op. 38: Melancolia	SZOlyzZyWdw	Simon Van Holen - Tema	2:25
CF2 .wav	Bach Violin Partita 2	602vP3SnFr0	Kris King	3:08
CF3 .wav	Invierno en el Trópico, Estudio para Contrafagot	xU_H9yaKb5M	Mónica Fucci - Tema	6:42
CF4 .wav	Sublime Gracia en Contrafagot - Amazing Grace ....	LXediYzi3HQ	Enzo Zuleta	1:36
CF5 .wav	YouTube Symphony Contrabassoon Audition 2011	zqJKnJV6avQ	Shawn Jones	3:28
CF6 .wav	Bagatella V Carlo Galante	O-d24KmA8IM	Alessio Pisani	3:15
TROMPETA				
TP1 .wav	Chalkinoos for Trumpet solo	VgNqUhaK36c	Nikos Xanthoulis - Tema	5:10
TP2 .wav	Divertimento for solo trumpet, Op.21 (1970) : Allegro	3oj4Folv-40	tinethinghelseth	3:35
TP3 .wav	Stefan Wolpe: Solo Piece for Trumpet (1966)	6tLBO4E862o	Wellesz Rhapsody	2:44
TP4 .wav	Anthony Plog - Postcards for solo trumpet performed by ....	88F7Elhsaqc	Andon Markov	6:15
TP5 .wav	Rund-Funk for Solo Trumpet (Tabea Squire)- Josh Rogan	2y9PZsmWfy0	Josh Rogan	2:26
TROMPA				
TA1 .wav	Jacques-François Gallay - Caprice op. 32 No. 12 pour Cor ...	U0NF1G7L4dU	Alessandro Denabian	3:54
TA2 .wav	Caprice pour cor Op.32 No.3: Jacques-François Gallay, ....	UnWPJyulMu4	Ricardo Oliver Rodríguez García	4:24
TA3 .wav	Oswaldo Lacerda – Melody for horn solo (1974) – Zuva – Yuri...	iqd88-FxoJM	Zuva - Yuri Zuvanov	4:33

Nombre archivo	Título del video	Dirección URL youtube.com /watch?v=	Autor / Usuario	Durac.
TA4 .wav	Camargo Guarnieri – Etude for horn solo – Zuva - Yuri Zuvanov	moFOG8V0lqo	Zuva - Yuri Zuvanov	2:43
TA5 .wav	Bernhard Krol, "Laudatio" for Horn Solo , Grigory Yakubovich,...	pwSD7CJppfl	Grigory Yakubovich	4:47
TROMBÓN				
TN1 .wav	Daniel Watt, trombone 2012 New World Symphony ...	lZmGCTRaJWI	Daniel Watt	6:01
TN2 .wav	NYO-USA 2017 Audition: Trombone excerpts	KrQEYrEoeZo	Cole Davis	3:51
TN3 .wav	Scelsi - Three Pieces for Trombone, Piece No. 3	lxV8BhJAd3M	Michael Clayville	3:17
TN4 .wav	Keren for Solo Trombone	H7Us_KYVUvg	Michael Buchanan - Tema	7:30
TUBA				
TB1 .wav	Solo for Tuba	XDOtRZsGQHo	Various Artists - Topic	6:47
TB2 .wav	Three Essays for Solo Tuba: Interlude	pK0rDCIQJk	Gene Pokorny - Tema	2:59
TB3 .wav	John Fletcher - Fantasy for Tuba	z5fJI-7Utwc	EGIL94	4:42
TB4 .wav	Monologue No. 9 for Tuba: II. Allegro vivace	gluMo9SCb8M	Various Artists - Topic	3:00
TB5 .wav	Suite for Unaccompanied Tuba: III. Air: Andante	da-WVVFHS9KY	David Zerkel - Tema	1:49
TB6 .wav	Fantasy for Tuba, Op. 102	rKaMIXDW_ZA	Eugene Dowling - Tema	4:11
ACORDEÓN				
AC1 .wav	Carlos Gardel Argentine Tango Argentino VOLVER -...	1UewFrzAMXU	Accordeon Accordion Jo Brunenberg	3:19
AC2 .wav	Adios nonino acordeon solo, Pedro Saad	FARM0KBZwLo	pedrosaad4	4:47
AC3 .wav	Vilma plays Chardas by Monti on accordion. Dragspel ....	qVitiRTgIAM	Slattegard	4:08
AC4 .wav	J.S.Bach: Prelude et Fugue BWV 543 (Michele Bianco:....	QZYIMetYTcM	Michele Bianco	9:48
ÓRGANO				
OR1 .wav	Tocatta and Fugue in D Minor (Best Version Ever)	ho9rZjlsyYY	MovieMongerHZ	9:20
OR2 .wav	BEETHOVEN 5 <sup>TH</sup> SYMPHONY - ORGAN - ST NICOLAS ....	NSSbF0HTgHs	scottbrothersduo	9:04
OR3 .wav	Wedding Processional from the Sound of Music on Organ   ....	zw8bEkxluEE	Reagan Reese	1:54
OR4 .wav	Organ Concerto in A Minor, BWV 593: I. (Allegro)	Y1RjQChnTz4	Marie-Claire Alain - Tema	4:14
XILÓFONO				
XI1 .wav	Firebird, Hary Janos, Porgy and Bess, Lady Macbeth - ...	OpDmGM7d1GI	Alex Frederick-Percussion	6:58
XI2 .wav	Spanish Dance (Manuel de Falla). Xylophone solo.	aBJ7iXFjg7U	LEGE ARTIS	3:05
XI3 .wav	"Geometria" for Xylophone - Composed by Lok Yin Tang	JbGvqve_JH0	Lokyin Tang	2:12
XI4 .wav	Sonata for Xylophone	aFoEDxoBijQ	Zach Koors	8:08
XI5 .wav	Xilófono: "El Vuelo del Moscardón"	Zcu8TH9UyPg	Eva Zapico García	1:26

## Anexo B: Corpus de audios de test y respuestas proporcionadas por el sistema

En este apartado se muestra el listado de los videos online disponibles en la plataforma YouTube que se han consultado para extraer señales de audio con las que probar el sistema implementado. También se muestra la categoría real de los instrumentos contenidos y la respuesta que facilita el sistema.

Tabla 24. Corpus de audios de test y respuestas proporcionadas por el sistema

Título del video	Dirección URL youtube.com/ watch?v=	Autor / Usuario	Categoría real	Respuesta del sistema
YTSO 2011 Violin Audition	zRi1NwshMbs	Leyan Lo	VN	VN
Violin Audition for the YTSO - Oksana Kurtash	t0jdigM3v1E	Oksana Kurtash	VN	VN
Alexandra Early, Schumann/Mozart excerpts	RoAoZEwoJtc	shmalibu919	VN	VN
Woo's Brevard 2018 violin audition (concertmaster)	xbkPX2vTqaE	Woo, Eunmin	VN	VN
Gisela Santos - Violin Audition for the YTSO 2011	AmhhQI1lmi0	GiselaSantos16	VN	VN
Violin solo - 21 days before the audition!	Cmahjh65yTU	LJF1975	VN	VN
Olivia Knudsen- Reverie 2nd violin solo audition	Ltub12Qe6jl	Olive Knudsen	VN	VN
2016 Aspen Violin Fellowship Audition	xfskjm9_Gzs	Greg Gennaro	VN	VN
Youtube Symphony AUDITION VIOLIN Practice FASTER (for fun)...	WRA6ltb9Z7I	risingpianofire	VN	VN
David Hartnett -- Violin -- MYS CO Audition 2017 (solo piece)	eX58BEiPorM	Temporary Channel	VN	VN
YTSO 2011 Viola Audition	_WDqh2RDBiE	lachrymae1	VA	VN
YTSO 2011-Viola Audition	7tm_jQ592So	jemus20	VA	VA
MAX REGER SUITE NR. 1 VIOLA SOLO   2nd Movement   Sara...	_ateoTWJxoM	Sara Ferrandez	VA	VN
"Dream" for viola solo	1UW0DCnenJ8	Povilas Syrrist-Gelgota	VA	CL
Necil Kazim Akse - Capriccio for Viola solo	Dz__vAry-RI	fahrettin arda	VA	VA
Zelda Theme - Viola Solo (Not a Violin!)	NpGnl_4SK_4	Chris   Amaterasu	VA	VA
Marco Misciagna plays Anzoletti Caprice No 2 op.125 for viola solo	IECVUENo4uY	marcomisciagna	VA	VN
HENRI VIEUXTEMPS Capriccio for Solo Viola "Hommage à ....	rHFcX82rEcY	Julia Kim	VA	VA
Penderecki Cadenza for Solo Viola	m22CIV44DKk	Violist Haesue Lee 이해수	VA	VN
Max Reger - Suite No. 1 in G minor for Solo Viola (Tabea Z....	RXrmsKo3KaY	ClassicalMusicMatters Eric Moon	VA	VN
Aurore Dasse - Suite for cello solo - Gaspar Cassadó	9NMEqj7LG6w	Young Belgian Talent	VC	VC
F. Schubert Ave Maria Cello Solo Wedding Ceremony	Isf_UHKqcF8	Cellopedia	VC	VA
GYÖRGY LIGETI - Sonata for Cello solo   Mathias Johansen	s92f3CW9ldQ	mathias johansen	VC	VC
Cassadó - Suite para violonchelo solo - I Preludio Fantasía	MNB8H2p9Kk0	LUIS CARLOS HERNÁNDEZ-LUQUE	VC	VC
Bach Cello Suite No.1 Preludio	_cFr6thUi0	Yan Rodriguez Hachimaru	VC	VC
Sarabande de la Suite N°1 para Violonchelo Solo de J.S. Bach	M2iCilPST7k	Toca Chelo Music By Armando Alterio	VC	VC
Jens Peter Maintz plays Ligeti Sonata 1. Mov.	Y4MGkjXSZRQ	Jens Peter Maintz	VC	VC
'J.S. Bach - Suite for Solo Cello no. 1 in G major - Sarabande' by...	pEzLdb0fFeM	CBC Music	VC	VC
Yo-Yo Ma - Bach: Cello Suite No. 5 in C Minor, Allemande	JkskMu6E2Mk	Yo-Yo Ma	VC	VC
Preludio, Suite para violonchelo solo (G. Cassadó) – Alberto....	vkb_VRFualo	Daniel Berlanga	VC	VC
Ode de España   François Rabbath   Double Bass Solo	4rgunENU-HU	Peaking NewLowZ	CB	CB
Dominik Wagner: Paganini Variations for double bass solo	hcfngx-FMxs	Dominik Wagner	CB	CB
Europeanut Memories for solo Double bass	EqQO9lwAVtl	youngseog sung	CB	CB
Double Bass Excerpts - Mozart Symphony No. 40	jTFkwIQMwQc	DrMarkMorton	CB	VC
Daft Punk Bass Solo - Sam Suggs	FmOTUULSsXw	Sam Suggs	CB	CB
double Bass Solo-Vranjanka /// By Nenad Vasilic	x2SuEU_umY	Nenad Vasilic Music	CB	CB
Paul Chambers's double bass solo on Straight, no chaser	eTae-biUqKU	Simone Vason	CB	CB
Jazz Upright Bass solo: "What Is This Thing Called Love"	Vf1QcpgTvnQ	marcopanascia	CB	CB
Andrés Martín: Sonata for Solo Double Bass	u17C1gPB7z0	DaddarioOrchestral	CB	CB
VERDI: Rigoletto double bass solo excerpt. Acto 1 nº3, Duetto. ....	QSVyrJe61JE	Damian Rubido Gonzalez	CB	CB
Cascada - Daniel Jordan (arpa)	wv8c5FP5gTs	Daniel Jordan	AR	AR
Hotel California - Solo de arpa	IU3UMBPPQQA	Linda Iriarte	AR	AR
The Fountain — Marcel Lucien Grandjany	S1-LCnGoinw	ClassicMmediaGroup	AR	AR
Beauty and the Beast [Alan Menken] // Amy Turk, Harp	cS5d2wD8Ool	Amy Turk	AR	AR
Despacito - Luis Fonsi - Harp cover by Evéline Simon - arpa - harpe	zuTcGvSNG4E	Evéline Simon	AR	AR
Fantaisie for Solo Harp Op.95 - Camille Saint-Saëns (Maia Darne)	U7KyfpP-eoQ	Maia Darne	AR	AR
Youtube Symphony Orchestra - Maria Chiossi Harp Audition	WAbFNMFdiq8	skiobala	AR	AR
Glain Dafydd orchestral harp audition 1	2nBB08vWNw0	Glain Dafydd	AR	AR
Angelica Hairston 2011 Youtube Orchestra Audition	4szYryGuUII	Angelica Hairston	AR	AR
Kassandra Harp Audition	ubu8OBMPzLk	Malcolm Lillie Jr.	AR	AR
Alessandro Scarlatti, Folia - Giannalisa Arena, clavicembalo ....	QHF8HQ-LsFA	elisabetta09	CV	CV
18 Clavicémbalo	2-Oda-nK_rw	Carlos Remón	CV	CV
Domenico SCARLATTI - Sonata K 29, L. 461 - Gianandrea ....	Y5qrU2dcEpA	ALTELAUTE	CV	CV
CLAVICEMBALO	9ooJcT5NQMg	José María Berenguer	CV	CV
Aria - W. A. Mozart - versione clavicembalo	OgATeZMOPuM	NICLA MARONE	CV	CV
Jean-Philippe Rameau: "Les Sauvages" (Gianandrea Pauletta / ....	3c5OD70M8Zc	Gianandrea Pauletta	CV	CV
Rameau Harpsichord Solo Les Tourbillons	466KkuormNQ	Elaine Comparone	CV	CV
Les Cyclopes	0Yttn57fm_o	Jory Vinikour	CV	CV
David Louie   Rameau   La Lapoplinaire   Solo Harpsichord Version	Q3QJmOzKQnM	David Louie	CV	CV
Tit manman : Suite Cabman #2 - for solo harpsichord (Paulo Bottas)	v3mpYcPWrhU	Paulo Bottas - Music	CV	CV
Spanish Guitar Solo ... On Acoustic Guitar.	z8X1LXDFFiQ	Marco Cirillo	GT	CB
Best Solo .... Spanish Guitar	J7lY6jKVC8	Dj.Sasha Valadi	GT	GT
Malagueña - Flamenco Guitar - Ben Woods	I3jWsoQ8W8g	Ben Woods	GT	CV
The Mask of Zorro theme - Fingerstyle Guitar (Marcos Kaiser) #102	RmLs8BfEbsU	Marcos Kaiser	GT	CV
Rumba Flamenca, Solo Guitar	aleh4SL6mfY	Guitarschool at	GT	GT

Título del video	Dirección URL youtube.com/watch?v=	Autor / Usuario	Categoría real	Respuesta del sistema
Ana Vidovic plays Asturias by Isaac Albéniz on a Jim Redgate ....	inBKFMB-yPg	SiccasGuitars	GT	CF
Classical Guitar - Joaquín Rodrigo - Adagio from Concierto de....	knZyzhMNeP0	Michael Christian Durrant	GT	GT
Spanish Guitar Flamenco Malaguena !!! Great Guitar by Yannick....	oTXa6FFnPI0	Je Pousse Un Cri	GT	VC
The Most Evolved - classical / spanish guitar - John H. Clarke	PHU-AJTn0I8	johnclarkemusic	GT	GT
Armik - Toledo Solo (Live Variation) - Official – Nouveau....	OctS6om3rIE	Armik	GT	VC
Hallelujah - Ukelele Solo - Jake Shimabukuro Version	7nMtyfLN3Lo	hanskait	UK	UK
Georgia On My Mind (Ukulele Solo)	UxxqJeF6kQI	James Hill	UK	UK
Daft Punk's "Get Lucky" Cover by Jake Shimabukuro   People	VbTrdj7vPGU	PeopleTV	UK	UK
Yellow (Coldplay) ukelele solo cover	GELMSOHHNmU	Cesc martorell pujol	UK	UK
Pétalo se Sal (Fito Páez) - versión ukelele solo	SNnUlr4i-M	Martin Diego Cabello	UK	UK
Donde se cierra una puerta, otra se abre - Ukelele solo	9C-0YCF5L3s	Choan Gálvez	UK	UK
Stand By Me (ukulele instrumental) HD	VCflhu57N8g	moonkoe	UK	UK
Creep { ukulele cover - instrumental / fingerstyle }	61gW-9F9bCk	Le Grave Musicien	UK	UK
All Of Me - John Legend (Ukulele Instrumental)	g0pjCrua6pU	ukulenny	UK	UK
Rick and Morty - Evil Morty's Theme [For the Damaged Coda] ....	pTbmKzdcJQc	SantiagoT	UK	UK
Jarrold Radnich - Virtuoso Piano Solo - Pirates of the Caribbean....	n4JD-3-UaZM	Jarrold Radnich	PI	PI
Queen - Don't Stop Me Now   Piano Cover - Peter Bence	LrQnnhAXLT0	Peter Bence	PI	PI
Renaissance - David Hicken (el arte de Piano) Piano Solo -....	I3lzWAaj9f8	David Hicken	PI	PI
"IMAGINE" (JOHN LENNON) PIANO SOLO.	LE0YdUhn8cY	Lucio Villegas	PI	PI
Cuna de Lobos - Pedro Plascencia - Piano Solo	L8Pc_Pj_EVI	DarkPianoLord	PI	PI
Classical Piano Audition	CyHLWL_G6kY	Ronnie Andrade	PI	PI
"Misty" - (piano solo)	s-UjwvJ0eeA	Jeff Williams	PI	PI
PERSUASION - PIANO SOLO BY DAVID HICKEN (sheet music ...	JAneZddWMx8	David Hicken	PI	PI
The Pixies - Where Is My Mind (piano solo)	36mktcbEwS4	TheWh0K0wZ	PI	AR
Gabriel Fauré - Pavane Op.50 - Piano Solo - P. Barton, FEURICH...	_cfkbUzX2a8	Paul Barton	PI	PI
Zach Sheets YTSO 2011 Piccolo Audition.dv	OuQ_ga788dU	Zach Sheets	FN	FN
Youtube Symphony Piccolo audition	2GBJtWCdBt4	Daniel Sharp	FN	FN
Piccolo Audition	41SHvMkVVKc	Galina Popkov	FN	FN
Gioachino Rossini Semiramide Overture Piccolo Excerpt Inteiro	2hyphnGLvoo	Júlio Zabaleta	FN	FN
YouTubeSymphonyOrchestra:BethLarsen:Piccolo	I9FAWsuKwuM	Beth Larsen	FN	FN
YouTube Symphony Audition 2011 - Zachariah Galatis, piccolo	cNT8PqxcgDQ	musicactually	FN	FN
Bolero USMC Audition Piccolo	uQSS5g8BbXY	Ryan Pederson	FN	FN
youtube symphony audition piccolo	XjqTulHgiUI	taiwanfluteband	FN	FN
Pitt Band Audition: Piccolo-Kaytie Ng	UYIaXeR7djA	Kaytie Ng	FN	FN
piccolo orchestral excerpts	OV_qGwRmNrg	Ju Hee Kim	FN	FN
2011 YouTube Symphony Flute Audition - Paul Hung	JclNk5jqICE	Paul Hung	FT	FT
Flute audition for the Orchestre de la Francophonie	aJ71eVf28Y8	Luke Fitzpatrick	FT	FT
Malcolm Arnold: Fantasy Op. 89 for solo flute	Whuwn3h_tv0	Carlos Feller	FT	FT
Flute Audition	wrEIS7BPRxU	kylie furman	FT	FT
UCLA Marching Band Audition - Flute	SYVV4YbUZ_w	Joy Guo	FT	FT
Syrinx flute solo by Emma He	QAiwlq3aP2U	emma he	FT	FT
Emmanuel Pahud records 'Jade' for Solo Flute (Pierre-Octave....	IF8D2HraG7I	Warner Classics	FT	FT
Paganini Caprice No.24 Flute 파가니니 카프리스 24 번 플루트....	Sz40f_geQRo	Jasmine Choi	FT	FT
Fantasy for Flute, Op. 89	vLnbvm3ohjQ	Sir James Galway	FT	FT
2017 CO All-State Flute Audition- Performance of Ernesto....	sYPeRGasB_1	Bel Canto Flute Academy	FT	CL
JS Bach Cello Suite No 3, BWV1009: Prelude for solo recorder	9RQ1P4v9yr8	Glen Shannon	FD	FD
Engels Nachtegaeltje for descant recorder solo from Der Fluyten....	eKc8mY3Rly8	bethanieliuny	FD	FD
Year mark for recorder solo - winter photos : IV. Ein Hauch von....	E0XmL-j4IQg	natthapong sirisit	FD	FD
"A Song for Japan" Tenor Recorder Solo	J_PxivjIz80	despicablesquid	FD	FD
Georg Philipp Telemann:-Fantasia n.1 for solo recorder	rjzpl-TXBv8	TOMMASO ROSSI	FD	FD
Stefan Temmingh (recorder) plays Telemann's Fantasia no. 10	O6X3Bg7Z8qo	Dr. Andreas Janotta ARTS MANAG...	FD	FD
A. Bova, Matin de printemps, for solo recorder	c53wVuJKdFs	unagondolaunremo	FD	FD
Fantaisie No. 1 in C major for Solo Recorder - Vivace	q9cTTWCasCQ	Jiří Stivín - Tema	FD	FD
Cantus for solo recorder: I. Syng	1pthW4Vku4	Caroline Eidsten Dahl - Tema	FD	FD
Lizard: Variants, Op. 166 No. 2, for solo recorder	rmAofC7qwNw	John Turner - Tema	FD	FD
Utah All-State 2020 Clarinet Audition   Nicholas Zhang	fKnmTEaEaA0	nicholasz2510	CL	CL
2020 SDYS Audition Excerpts	9HdyvVSEx6o	Brian Lee	CL	CL
Jose Franch-Ballester plays Sonatina for Clarinet Solo (Full) by....	cTDaPHNHX7M	Jose Franch-Ballester	CL	CL
Jonathan Jones - Widmann: Fantasie solo Clarinet	4IXPdLFslJo	Jonathan Jones	CL	CL
I. Stravinsky 3 Pieces for Solo Clarinet - Eduardo Sanchez Moreno	nN29MPXy70E	Eduardo Sanchez Moreno	CL	CL
Monologue Nr. 3 - E. von Koch for clarinet solo	0u8gQg5BXZ8	Javi Olmeda	CL	CL
ORQUESTAL SOLO EXCERPTS   SOLOS ORQUESTA  ....	kWZ52ipwCfU	JOSÉ MARÍA SANTANDREU - clarinet	CL	CL
I. Stravinsky: 3 Pieces for clarinet solo (Jaehee Choi/ 최재희)	dKg0OEzVGZA	nfamusic0105	CL	CL
Stravinsky Firebird Clarinet Solo	x7yiHwxutnQ	SeanOsborn	CL	CL
Valentino Bucchi , Concierto para Clarinete Solo. Movimientos I y II	H8k5zSK19j0	Duván Andrés Aristizábal	CL	CL
Prelude from Cello Suite No. 1 by J.S. Bach - Bass Clarinet (HD)	3pAdLMSGpbY	Marcelo Maldonado	CJ	CJ
Bass Clarinet Blues Improvisation solo by Cornelius Boots live in...	ciGuMya0yYQ	Edmund Welles: the bass clarinet quart	CJ	CJ
Flight of the Bumblebee on Bass Clarinet	UGeLOC7eTtg	Christopher Howard	CJ	CJ
So you want to be a Bass Clarinet player: Patrol Russe	BBlhFHKd_kw	Earspasm Music	CJ	CJ
G. Mattalano - Gibran, for solo bass clarinet	OJrmUrbC7CE	Elisa Marchetti	CJ	CJ
Bach Suite n°1, Prelude	6oRvq1As_JM	Nicolas Baldeyrou	CJ	CJ
G. Verdi Ernani Act III Bass clarinet solo	eWbzV-d1ZbE	Adriano Ricci	CJ	CL
Perfis, for bass clarinet solo.	AD3YqYSrSO0	Germano Blume	CJ	CJ
Shostakovich Symphony no. 8, Bass Clarinet solo	45ImgzlaABI	ClarinetJohnny	CJ	CJ
Astor Piazzolla - Tango Etude Nro 3 - Bass Clarinet	4qzA17JoKGE	SeBass	CJ	CJ
Sax Acapella	1U9CyX7wElg	Jani Ho - Tema	SX	VN
solo saxo, Eduardo Zapata, jazz.	Z7Ug4jm8Xqw	Eduardo Zapata Doria	SX	VN
More Than Anything - acapella tenor sax cover.	ooQ5KGTgq-o	Justin Hamby	SX	TA

Título del video	Dirección URL youtube.com/watch?v=	Autor / Usuario	Categoría real	Respuesta del sistema
Pescador de hombres ( FELICES PASCUAS ) saxo cover a capella	n-07-odGZ1I	Christian Cross	SX	SX
Ivan Renta, tenor sax solo	89A0vvoel6M	congahead	SX	TA
Coro 02 - Toda a glória a Jesus - Sax Alto Acapella	onPuZLgv584	Wagner Costa Sax	SX	VN
Chullita quiteño a capella instrumental saxofón por Rafinsky	b14XZagn3fk	Rafinsky Albarran Sax	SX	SX
Tenor Sax solo impro. Test Yanagisawa T9937, Olivier Waleckx	A_BevyMKp7w	Dragon Khan	SX	TN
Improvisation n°1 - A Cappella (Saxophone)	2UL8DdgXa9Q	Basile Godtbil	SX	TP
SAX A CAPELA (Leia a descrição)	X4bU9ZR1Vfo	Denis Campos	SX	TA
Carl Philipp Emanuel Bach/Sonata en sol menor oboe solo/E....	_c6fX117M8o	Eduardo Martínez Caballer	OB	OB
Swan Lake Oboe Solo	gXWZCtjgDms	Andrew Parker	OB	OB
Dorati: 5 (Cinq) Pieces For Solo Oboe - 1. La Cigale et la fourmie	gUBoPhApX68	Diana Doherty - Tema	OB	OB
Wood Song for Solo Oboe	UbCK2u9H_Xk	Lindabeth Binkley - Tema	OB	OB
Telemann Fantaisie no. 2 for solo oboe	gDTwZRGtXl	Bernice Lee	OB	OB
Blush for Solo Oboe by Jean Ahn	cOeQxf2Mx0	Jung Choi	OB	OB
Amethyst Deceiver for Solo Oboe	U5s6dYHCDAg	James Turnbull - Tema	OB	OB
Six Metamorphoses after Ovid for Solo Oboe, Op. 49: I. Pan. ....	PqIWHsUBijQ	Céline Moinet - Tema	OB	OB
J. S. Bach - Partita for solo oboe - Sarabande	_g3-CBqOcnq	D_Bunc Oboe	OB	OB
Syrinx for solo oboe	6VUF89ZSvUg	Christophe patrix	OB	OB
Dvořák "New World" - English Horn Solo	fjoPMmQ-1Fo	Vivien Kong	CI	CI
R.Wagner : English Horn solo from Tristan und Isolde / Sho Music...	WPGZt04tiis	Sho Music Festival Online	CI	CI
English Horn Audition 2011	GvB33tAy7QQ&	DRAWRoss	CI	CI
YouTube Symphony Orchestra 2011 audition - English Horn	qo5bOJrgXpA	Simon Lee	CI	CI
Moritzburg Festival Audition - English Horn	xilHudLc26o	Moritzburg Festival Audition	CI	CI
Cheng Wai Lam Asian Youth Orchestra 2017 Audition: English Horn	cKzH_DbMtzl	Ivan Cheng	CI	CI
Parable for Solo English Horn	CkTiWmDB8bA	Kevin Chavez	CI	CI
A Certain Tune. 5 Songs for English horn Solo by Juliana Hall	FnqNAqGV27Y	Margaret Marco	CI	CI
Dvorak Symphony No. 9 "From the New World" Mvmt II English ....	7hRh3Xrwwl0	Tiffany Huang	CI	CI
MET Orchestra Musician Pedro Diaz performs the plaintive ....	WyGoU13zcZc	MET Orchestra Musicians	CI	CI
¿Cómo suena un fagot?	I-QID08ZYP4	Miguel Ángel Balaguer	FA	FA
Marcolm Arnold - Fantasía para fagot solo	sT2wkQV2wPo	Gabriel O.	FA	FA
J. S. Bach, Allemande de la partita para fagot solo BWV 1013	eLpIIMiY2w	Andrea Perez	FA	FA
Bram van Sambeek - J.S. Bach/ Uit: Partita voor fluit Sarabande	oOfzMjwQP0s	vprou vrije geluiden	FA	FA
Laura J C Sonata para fagot solo, Harald Genzmer	1JFwICbsUs	Laura Jiménez Celorio	FA	FA
Partita for Solo Bassoon - Gordon Jacob (KyuSun Pyo) 바순 표규선	T7Jm30gNmS8	바순표규선 KyuSun PYO	FA	FA
Shao-Jung Wang - J.S. Bach: Partita for bassoon solo BWV1013	LoyNk8Y51s4	Shao-Jung Wang	FA	FA
Katia Osorio - Bassoon - OA Audition 2020	Ox1oC1pxrgo	Katia Osorio	FA	FA
Aleksandr Popov, Bassoon. YouTube Symphony 2011 Audition	En3ZQDepsno	yes	FA	FA
Catherine Chen - YouTube Bassoon Audition 2011.mov	C1Ms4wQNfis	cychen92	FA	FA
Jarrett Grempel Contrabassoon Excerpts	lIQA-2ssqgQ	Jarrett Grempel	CF	CF
"Oak Demon" for solo contrabassoon - Dave Volpe	9lIhe3vEp-0	Anthony Parnther	CF	CF
Schulhoff Bass Nightingale for Contrabassoon	Df6GWZ0cl1U	Fiona Butterworth	CF	CF
Contrabassoon Plays Beethoven	nBHzyTLTH4I	Brian Peterson	CF	CF
Tan Dun Contrabassoon	Qef8aUIBj8	Anrijs Ivanovskis	CF	CF
Verdi "Don Carlos" contrabassoon solo Andrei Huzeyeu	5RzK5pmKfe0	gvarnerius	CF	CF
More Bach on Contrabassoon	wgd_ad-QnUw	JayBassoon	CF	CF
Barry Gray basslines on a contrabassoon	vNfbBFi4NiU	Alfie4	CF	CF
Tim Wells Quarantine 2 - Contrabassoon	OrHuxoEPTx0	Tim Wells	CF	CF
Symphoria 2nd Bassoonist and Contrabassoonist Jessica King At...	GpiyBneue_E8	Symphoria	CF	CF
"Black eyes". Trumpet solo acapella.	4fVUIZleYps	Anzor Abdelhammad	TP	TP
nothing's gonna change my love for you trumpet solo	NOO7029QeY	trumpet chie	TP	TP
New World Symphony Trumpet Audition, 2012 - Nick Hall	hNK3jYrp3kU	Nick Hall	TP	TP
LA BIKINA   RAYMUNDO CORONADO trumpet solo	tiRmbTFYaZ8	Mariachi Channel	TP	TP
Trumpet - The Technician	scrj-c2gt0s	Green Bay Solo & Ensemble Project -	TP	TN
2011 YouTube Symphony Orchestra Audition - Trumpet	4I1VjI4G9_M	BughouseMASTER	TP	TP
David Sampson - SOLO - for Trumpet Unaccompanied - Fabrizio...	KXzRVtpaKRY	Fabrizio Iacoboni	TP	TP
Summon the Heroes Trumpet Solo - Trint Castle	eCQO6k5Qrmg	Trint Castle	TP	TP
Trumpet solos from JFK performed by Alex Brain	MKB2IWo9S-M	Trumpet Brain	TP	TP
Solo Trumpet (Unaccompanied)	wjcmuV150F0	Justin Powell	TP	TP
PMF horn audition 2019 - Kaylet Torrez	eNGHFCEzPPQ	Kaylet Torrez	TA	TA
Gabriella Sá- French Horn- YOA Audition 2018	kQigSdKq-2w	Gabriella Sá	TA	TA
Verbier Festival Orchestra Audition 2014. Miguel Morales, Horn	GGVUUB4Nhgk	Miguel Morales Llopis	TA	TA
Horn Audition	ndH0G-Aq-yE	Kaylet Torrez	TA	TA
David Cooper French horn Youtube Symphony 2011	rv9U5KldMc	David Cooper	TA	TA
Alma Horn Audition 2020	9JyuETHsJc	Elizabeth Vredevelt	TA	TA
Daniel Itzkowitz - French Horn - OA Audition 2020	Mz0Ta4o-MZ8	Daniel Itzkowitz	TA	TA
Ian Mayton Youtube Symphony Horn Audition 2010-11	JVm_q8_YApk	Ian Mayton	TA	TN
Horn audition	4S8L4ODqCTE	Gadiel Caleb Ocasio	TA	TA
Allie DeGraffenried BYU-Idaho Horn Audition	oBGsF7nhdxY	Allie DeGraffenried	TA	TA
YouTube Symphony Orchestra Audition 2011 Trombone.mp4	mZNNWTrvGoA	Mrvuvuzelaman	TN	TN
Trombone Orchestral Excerpts Mock Audition	JIU-nBftgqY	Ohtae Kwon	TN	TN
Noah Roper - Bass Trombone - OA Audition 2019	CYjNb4oADko	Noah Roper	TN	TN
Bass Trombone audition tape for the College Summer Music ....	HBhdD4IOoto	Jahleel Smith	TN	TN
Bass trombone audition excerpts: "Symphony no. 5", by Gustav ...	fBDECs4xe2Y	Christian Jones Bass trombone	TN	TN
Brass Music: III. Elegy for Mippy II for Solo Trombone	wh23GCR9QPU	Joseph Alessi - Tema	TN	TN
YTSO Audition 2011 Trombone Jorge Navarro	BWnLOTVuxAQ	80jnavarrom	TN	TN
YouTube Symphony 2011 Bass Trombone audition	2ApqVonJSMY	Last Train Brass	TN	FA
"Mosaic" for solo trombone. Composed by Erik Ørum von ....	X7r2p6mJd9Q	erik ørum von spreckelsen	TN	TN
YouTube Symphony Orchestra 2011 Audition Bass Trombone. ....	2LBsevI3Cd8	Sebastian Cifuentes	TN	TN
Air Suite J.S Bach	83zyulbrIn0	Tubist Jamal	TB	TB

Título del video	Dirección URL youtube.com/watch?v=	Autor / Usuario	Categoría real				Respuesta del sistema
NYO 2019 Tuba Excerpt Audition	OA4Ef9Dixfg	Destin Hoskins	TB				TB
TUBA AUDITION HONG KONG PHILHARMONIC	9GpG7L_hpKE	Sergio Sanjuan	TB				TB
Youtube Symphony Orchestra Tuba Audition 2011	IGRfvpk-gAA	sebastiantuba94	TB				TB
Hong Kong Philharmonic Orchestra Tuba Audition 2020	Kd_CiAQbNS0	Ferran Martinez	TB				TB
Melton "Tuono", Orchestral Excerpts Test	LqOZVWbOSjE	Alessandro Fossi	TB				TB
Dillon Jefferies-Tuba-YOA Audition 2017	f9cUizWCcwk	Dillon Jefferies	TB				TB
Youtube Orchestra 2011-Tuba Audition	f6NLRQQG_I	Isaac Rodríguez Cotrofe	TB				TB
UNR Tuba Audition Joshua Petterson	26_9TEga8ic	Joshua Petterson	TB				TB
Carolina Crown Tuba audition 2016	hQ8J5BbBCJk	Eric Brown	TB				TB
Cirque du Soleil accordion waltz	hTz1rPcaG0s	Lucy Riddett	AC				AC
La Valse d'Amelie no acordeon - Yann Tiersen - DOUGLAS ...	q2UweAtj_Js	Douglas Borsatti	AC				AC
Resistiré/Arturo Ortega/Acordeón	KdUOMXl0kP0	Arturo Ortega Calderon	AC				AC
Llorando se fue (Lambada) en acordeón	wi_YqenPE5A	Clases de acordeón	AC				AC
amor y celos , vals acordeon jose maria	qkz4wq6ufM	Jose Maria Perak	AC				AC
Danza Kuduro (gespielt von Stefan Bauer)	lkzPnRRBP8M	Stefan Bauer	AC				AC
Guns N' Roses - Sweet Child O' Mine - Douglas Borsatti ....	qDTru6gLvOs	Douglas Borsatti	AC				AC
Tico-Tico	agXHZ7T0npc	marmir3	AC				AC
When You Want To Play Hip Hop, But You Just Have An Accordion	njL3f6ZUCmo	Accordionsimon	AC				AC
EI Gateado - ACORDEON A PIANO	fXciTpsZ8o	LeandroFilmaciones chamame	AC				AC
VIVALDI - WINTER (Four Seasons) Organ of Basílica de Santa ....	S_AzwJwy7Ns	scottbrothersduo	OR				OR
Pirates of the Caribbean - Davy Jones's theme cover church ....	D_-qS_3KXBA	Grissini Project	OR				OR
Notre-Dame pipe organ improvisation by Olivier Latry	Ye4r_csWJDw	Fugue State Films	OR				OR
GRIEG - IN THE HALL OF THE MOUNTAIN KING - ORGAN OF....	dBoGdJ54oWI	scottbrothersduo	OR				OR
Bohemian Rhapsody Queen on church organ played by Bert van...	Wd6qZS1_1aU	BertsBytes	OR				OR
Epic Halloween Organ Solo - Toccata in D Minor	bG-MI_EN-wU	The Tabernacle Choir at Temple Squar	OR				OR
Final Fantasy VII - Organ solo One-winged Angel by Grissini Project	wOnpKgL-NUQ	Grissini Project	OR				OR
Two Steps from Hell - Heart of Courage, Organ	i119LgMDIXI	Pipe organs & Movies	OR				OR
Benedikt Müller - Viva la Vida (Organ) by Coldplay	4q9mA53XHGM	Benedikt Müller	OR				OR
Pirates of the Caribbean Medley (on Organ) / Fluch der Karibik...	magnG5fJDIA	Ben das Brot	OR				OR
Flight of the Bumble Bee xylophone	nd9xha3vk58	Dano Cracchiolo	XI				XI
Thomas Pitfield "Sonata for Xylophone Solo" por Alberto Chiroli	EjqGLQk-ezo	Alberto Chiroli	XI				XI
Bena Havlu - Paganini Capriccio XXIV, Xylophone Solo	oc3rnVw-xpQ	Faust FaustRecords	XI				XI
Gypsy Rondo Xylo solo	wXGXnk8xF-k	Matthew Suzelis	XI				XI
Valse Brilliante - Xylophone Rag by G. H. Green	EHf-2FD0dF0	Austin Weeks	XI				XI
The Hummingbird, a xylophone solo by G.H. Green	loCRmBvtHZ4	Alex Frederick-Percussion	XI				XI
Xylophone solo - Bena Havlu - R. Rod : Oriental Dance	JSXGy1Bcr0	Faust FaustRecords	XI				XI
La Primavera "Antonio Vivaldi" para xilófono. MJMPCADI	LESc1ICiEo	Manuel Jesus Martínez Pastrana	XI				XI
Concierto para violín Am Bach Xilofono 1 movimiento	FKHPPDckWWhw	Josue Pedroza	XI				XI
Porgy and Bess (introduction)- xylophone solo	XwdDUTpMMi0	Michael LaMattina	XI				XI
G. F. Fuch - Allegro. Arnaud Sussmann, violin - Jose Franch-....	lxZWJ6Tn8Qs	Jose Franch-Ballester	VN	CL	VN	CL	
Café 1930, Astor Piazzolla (arr. Duo-B) - Duo-B	2wmxck5D9Vw	Duo-B	VN	VC	VN	VC	
Trumpet/Bass Jazz Duo Demo	jQ0aMd_CJ84	Hart Guonjian-Pettit	CB	TP	OB	TP	
Dussek Duo pour harpe et piano	wpGIKSsyMpU	ConservatoriMallorca	AR	PI	AR	PI	
Ukulele & recorder duo - El condor pasa	oAXashVPSpU	Arnaud Martin	UK	FD	FT	FD	
Duo For Recorder and Harpsichord -Matan Ben Zahav	RvR6uYqjDc	מתן בן זורב	CV	FD	FT	FD	
Duo Luftwirbel - Caprice for piccolo & xylophone by Casey ....	OgHPzxB1sM	CaseyCangelosi	FN	XI	FN	XI	
Postlude Duet: Flute/Organ	ntfCz-CYARM	William Lord	FT	OR	PI	FT	
Odessa Bulgarian - Oboe and accordion klezmer	WvY7aC87YU	Dexter's Right	OB	AC	FN	OB	
DeGruchy English Horn/Bassoon Duo - Range Rearrage (DeGruchy)	5A7sVCePdwc	Toni Raynolds	CI	FA	CI	FA	
The Swan (Le Cygne) - Harp and Bass Clarinet Duet	a7ySS8nl5E	The Four Graces	AR	CJ	TN	TB	
Duo for Contrabassoon and Piano	8K26bmxNpcM	Various Artists - Topic	PI	CF	PI	CF	
Duo Flute & Horn - Thursday Jazz Duetto	_LTWj7ohC28	urednik1	FT	TA	PI	TA	
Edicson Ruiz, Elgar Duetto for Double Bass and Trombon	24Xfw39v6IU	doublebassVenezuela	CB	TN	FA	TN	
Ammerland - (duo tuba e trombone)	t523Uz4QrF8	Marco Jesus	TN	TB	FA	TN	
Duet for xylophone and flute (2nd)	L7jNEK4fRDs	Drumatic	FT	XI	FT	XI	
ukulele and cello duet	1gN0PKbt8zU	glivvy	VC	UK	VC	UK	
Camille Saint-Saëns, Danse Macabre (excerpt) organ and ...	jVoHnykvDq4	Ensemble Doppipunto	CV	OR	PI	OR	
A. Piazzola: Libertango (Trumpet & Accordion)	T_vvrZHGk5c	WindBrass MusicHumor	TP	AC	PI	TP	
Béla Bartók: Romanian Dances for piccolo and piano	w6vKqBO-gU	Peter Verhoyen	PI	FN	PI	FN	
Dutch Suite for Tuba and Bassoon	E-TH0nPIA8M	Ken Drobnak	FA	TB	FA	TB	
Fun time with Steinmetz duo for Contrabassoon and Bassoon	46eW4jju60	Hans Agreda	FA	CF	FA	CF	
Alun Cook "Mad Swans" Duet for clarinet and bass-clarinet	IRByWqmsxqc	Nikolay Inkizhinov	CL	CJ	CL	CJ	
E. BOZZA, Shepherds of Provence - Duo for oboe and English horn	kyFYaHnfdBM	ClassicalMusic	OB	CI	OB	CI	
Frédéric Nicolas Duvernoy - Sonata No. 1 for Horn & Cello	t6qqnDjeHqU	mirinae0904	VC	TA	CF	TA	
César Franck Prélude, Fugue et Variation, op.18 (harp and organ)	B1JeaOEqJIM	Olesya Kravchenko	AR	OR	AR	GT	
Accordion and Violin Duo [Carte Blanche Jazz Band]	YvyYp0ZRfko	CarteBlancheJazzBand	VN	AC	VN	OR	
Sonata for Xylophone and Piano	TEikbL7KPHg	Various Artists - Topic	PI	XI	PI	XI	
Sam Pilafian Tuba & Catalin Rotaru Bass-Petite Fleur	laVrSb1Qay4	profsuperhero	CB	TB	AR	TB	
Action! Duet for Horn & Trombone	OAVIkY5ycpk	Rick Peperkamp	TA	TN	FA	TN	
Pieces for Piccolo and Contrabassoon	YbsN2CzEjU0	Fluteintimesaves9	FN	CF	FN	CL	
Clarinet & Trumpet Duet	bgNkxMA3II4	Margaret Gale	CL	TP	OB	TP	
Capriole for bass clarinet and cello by Kimmo Hakola	96MHfz4tXhs	Claire Bryant	VC	CJ	PI	CJ	
"all of me" cover ukulele and trumpet	JyXpBuwPrmM	Dos Almas	UK	TP	FN	TP	
Lied (arr. for english horn and harpsichord)	zI2AIaQsn9E	Thomas Stacy - Tema	CV	CI	OB	CI	
Telemann: Duet for Recorder and Violin in C, TWV 40: 103 - 1....	VnSf4bJ8ZiY	Michala Petri - Tema	VN	FD	VN	FD	
CPE Bach Oboe & Bassoon Duet	h3p5n-pAQ_0	JDW Sheet Music	OB	FA	FN	OB	
LISZT HUNGARIAN RHAPSODY NO. 2 - SCOTT BROTHERS ...	orneY4K9YtQ	scottbrothersduo	PI	OR	PI	OR	
Hanzhi Wang & Giovanni Punzi, Accordion and clarinet duo, ...	i6LZ1oZ4ebg	Hanzhi Wang	CL	AC	VN	AC	

Título del video	Dirección URL youtube.com/watch?v=	Autor / Usuario	Categoría real		Respuesta del sistema	
Contrabassoon and Harp Satie	23HGvJw4VaQ	Rebecca Noreen	AR	CF	AR	CF
Ragtime Robin - George Hamilton Green (Xylophone+Ukulele) flute and tuba	yh6djWRsklc	Jeremy Lawi	UK	XI	FD	XI
Footprints for Trombone and Trumpet: V. In the Sand	qcDFLJSMCBA	Азат Килов	FT	TB	PI	VA
Bass Clarinet and Bassoon duet Misty Mountains	pKz5FHWlUpQ	Brad Edwards - Tema	TP	TN	TA	TB
The Nightingale by Deborah Henson - Recorder/Harp Cover	PUcHGmY02o	The junk rat	CJ	FA	CJ	CF
Samuel Stuchbery; Op. 4 - Cathedral Music (Duet for Violin and...	fIS86oAU9c	Biegel Family Music channel	AR	FD	UK	FT
Carl Philipp Emanuel Bach. Sonata in G minor for Oboe and ...	aroZ1s4_BFc	Samuel Stuchbery	VN	TA	VN	PI
Consonance (from "Fullmetal Alchemist Brotherhood") English...	HUhx3ZovcRg	Пана Ген	CV	OB	FT	OB
Duet for Tuba and Piccolo	ZJHObiYvI5w	IsrafelCello	VC	CI	PI	CI
Agnus Dei - G.Puccini ( double bass and clarinet duo)	cqaCC4rAnPk	Klaus Bruengel - Tema	FN	TB	GT	FN
Harp, Cello & Violin Trio	0Zp7JkBtaKE	Zappawoman - Valentina Ciardelli	CB	CL	PI	CL
Francis Poulenc - Trio for Piano, Oboe & Bassoon	WuSDpOhq6Jk	Deon Entwisle	VN	VC	AR	VA
A. Vivaldi, Trio in a minor for recorder, cello, and harpsichord, RV 86	mClyflwkD6c	pelodtelperro	PI	OB	FA	PI
Trio for Piccolo, Bassoon and Piano (2010) : Allegretto	l50S70Hf8Jk	mfufanc	VC	CV	FD	VC
Pastorale by Eric Ewazen (Trio of flute, french horn, and piano)	jIOa4Ag4O7c	Lior Eitan - Tema	PI	FN	FA	PI
Les Cordes Basses: Josef Myslivecek, Trio para violín, cello y ...	UroA1a9PYtE	Joon Lee	PI	FT	TA	PI
I'm yours by Jason Mraz. Ukelele, Violin and Cello	m0W3qX3ja_I	Ranlofer62	VN	VC	CB	VN
Divertimento-Trio for Oboe, English horn, and Bassoon	vDjrZ4n7_mw	Agez Producciones	VN	VC	UK	VN
Darius Milhaud: Suite op.157b for Clarinet, Violin and Piano -...	d0cDC4i09hU	Joseph A. Ryker	OB	CI	FA	OB
Veloce (Suite for Flute, Bass Clarinet and Piano) - Claude Bolling	TKYTU9M8oDI	Ensemble der KammerMusikKöln	VN	PI	CL	VN
Christmas Oboe, Cor Anglais and Contrabassoon	xJJIDgZRQ5Q	Sibylle Marquardt	PI	FT	CJ	PI
Trio for Trumpet, Trombone and Piano, Op. 285: Moderato ma ....	Dq7QLWVHD2o	Royal Philharmonic Orchestra	OB	CI	CF	VA
International Low Brass Trio plays Triangles for Horn, Trombone, ...	9QmiHMxwTXo	Wim Van Hasselt - Tema	PI	TP	TN	PI
Toccata for Violin, Xylophone and Piano	ShCakrwHHKc	International Low Brass Trio	TA	TN	TB	PI
Piazzolla Winter, Four Seasons (violin, accordion and piano trio)	_4LcXe98Wkc	gracehyma	VN	PI	XI	PI
Trio for Violin, Cello, and Organ, Op. 55: I. -	olStQa5NRZ8	Yueun Gemma Kim Violin	VN	PI	AC	AR
Trio for Piano, Violin & Double Bass, Op. 38: I. Adagio - Allegro ....	rcfXuu1Cjto	Clyde Beavers - Tema	VN	VC	OR	VN
Tjasha Gafner -Haydn Trio in D-Major: for Flute, Cello& Harp	M3LaxwMgHqu	Trio Sonata Concert - Tema	VN	CB	PI	PI
Suasana Hari Raya [Clarinet, Ukulele & Bass Clarinet Trio Cover!]	2NdEmp-KZY4	The International Harp Contest in Israel	VC	AR	FT	AR
Sonata in E Minor for Oboe, Violoncello and Harpsichord, BWV...	awRVeCTr6Kc	Elle Jimmy Tan	UK	CL	CJ	CB
Gaelic Fantasy for piccolo flute, violin & organ (piano)	sKyzR0USeQM	Ramón Ortega Quero - Tema	VC	CV	OB	PI
Telemann Trio sonata in E minor for recorder, flute and continuo	WLHskOyXCpE	Hans-Andre Stamm	VN	FN	OR	VC
Szilárd Mezei: Hep 7B (for bass clarinet, contrabassoon and ....	zvhwYJyAm0	Marta Femenia	CV	FT	FD	PI
Sonata for Horn, Tuba & Piano (1995) by Gary Kulesha _ ...	RjJ7IPZwVh0	Szilárd Mezei - Composer & Improviser	PI	CJ	CF	CB
Poulenc Trio for horn, trumpet, and trombone	3cBgQsvPSnk	Sérgio Carolino	PI	TN	TB	AR
Harry Potter for harp, xylophone and violin	KLU5PbVcMY	Kurt Civilette	TP	TA	TN	PI
trio con abbandono: "Pequena Czarda"	RvMTJbJmTHk	Music Play	VN	AR	XI	VN
Michael Haydn: Divertimento (for violin, cello, double bass)	xvUPlrhAp_A	Kaleidos Musikeditionen	VC	CL	AC	CL
HAYDN LONDON TRIO - ENSEMBLE FRAGONARD FRANCE 2...	jTQdbR6av3w	novnor	VN	VC	CB	VN
Ensemble Dialoghi - Reinecke Trio (1/4) for Piano, Clarinet and ...	wk8veywsQq8	Marc Zuili	VN	VC	FT	VN
Trio for Piano, Oboe and English Horn, Op. 118 (C. Reinecke)	N1Mw4waSg24	Pierre-Antoine Tremblay	PI	CL	TA	PI
Trio for Flute, Violin, Tuba	4ZJQ2i2mr8	Cremona Academy	PI	OB	CI	GT
Trio for bassoon, violin and cello: I	Roxe7xt2FFQ	Stella G Gitelman Willoughby	VN	FT	TB	VN
Eric Ewazen - Trio for Trumpet, Violoncello, and Piano	_gfvAGfp2_0	Valery Popov - Tema	VN	VC	FA	VN
Astor Piazzolla: Ave Maria (cello, flute, accordion version)	-rPTxpHczpo	Friesen Studios	VC	PI	TP	PI
Ave Maria - Harp, Flute, Violin & Cello Quartet	9Zv9mhavDMI	Rebecca X Nazar	VC	FT	AC	VC
Metallica - Nothing Else Matters (Quartet cover - Violin, Flute, ...	Xg_toBFUrvW	The Euphonious	VN	VC	AR	FT
Couperin / The Harpsichord Quartet, 1953: Le Parnasse ou ...	K_m5gQ1SapM	Y Michelan	VN	CB	PI	FT
Adios Nonino Astor Piazzolla Quartet: Violin, Piano, Accordion,...	7E9Zp01vpG8	davidhertzberg	VC	CV	FT	OB
Cassazione Quartet for Oboe, Clarinet, Bassoon and Horn (Rev....	x5J5MguOGIs	domenico zezza	VN	CB	PI	AC
Wind Quartet • Villa-Lobos • Quarteto Assai	xYoie20MR7M	Round Top Festival Institute	CL	OB	CI	TA
Bill Douglas - Quartet for Flute, Oboe Bassoon and Piano, II. Full...	U-OKz59ZoSrs	Villa-Lobos Channel	FT	OB	CL	FA
Paul Hindemith - Quartet for clarinet, violin, cello and piano (3 mvt)	0QhQ5EvUApE	Y Ishikawa	PI	FT	OB	FA
Quartet for Flute, Violin, Cello and Piano: I Alla Marcia	IzuXjXXK9_0	SilhouettesEnsemble	VN	VC	PI	CL
Quartet for Trumpet, Horn, Trombone, and Piano	7HcSVFNng2s	Eleonore Pameijer, Ursula Schoch, Ste	VN	VC	PI	FT
Quartet for Flute, Clarinet, Horn and Bassoon, Op. 50, No. 1: III. ...	-b6ZkX3Xex0	Sandra Woodward - Tema	PI	TP	TA	TN
Trio III in G Major for Flute, Violin, Cello and Harpsichord - I. ...	LC9MApVxwss	Orchestra Rossini Quartet - Tema	FT	CL	FA	TA
"Fade to Black" by Metallica, for clarinet, bassoon, trombone, and...	RaUIMdtF5XY	Musopen	VN	VC	CV	FT
Walter Rabl Quartet for Violin, Clarinet, Cello and Piano Op. 1, 2nd	cSBRb4KF5xs	Ceylan Yagmur	VC	CL	FA	TN
BOZZA Sonatine for Two Trumpets, Horn, Trombone, and Tuba	PLIEctbL58U	Else Ensemble	VN	VC	PI	CL
Liszt Premiere Elegie for cello, harp, piano & organ, S. 130	oID9nJb7KHY	Curtis Institute of Music	TP	TA	TN	TB
Quatuor Parisienne for Flute, Oboe, Clarinet and Bassoon	f_u-ELjaa1E	Sam Ou	VC	AR	PI	OR
Sonata in C for Trumpet, violin, Oboe and Continuo, by Gottfried...	K0Rq2sBquqQ	Charles Fernandez	FT	OB	CL	FA
Hindemith: Quartet for Clarinet, Violin, Cello and Piano	tt_Wx9gQySU	Rodney Leinberger	VN	CB	OB	TP
Ensemble Corrélatif, Eugène Bozza - 'Trois pièces pour une....	ybrzFSo-tMA	NIKITA MNDOYANTS	VN	VC	PI	CL
Just the Way You Are (Bruno Mars) cover - violin / cello / harp /	enXi09_t57c	musicomchannel	FT	CL	OB	FA
Doppler Nocturne for Flute, Violin, Cello and Piano	4qsJNEOjOs	Deans' Duets	VN	VC	AR	PI
Quartet for Clarinet, Violin, Cello & Piano: I. Moderato, flowing	Gvly751gjI4	Mary Kay Robinson	VN	VC	PI	FT
Michael Haydn Quartet for English horn, Violin, Cello and Bass, 1...	8nb2U7xhL7I	American Chamber Ensemble - Tema	VN	VC	PI	CL
Karl Goepfert Quartet for Winds, Op.93	ccwPKB71GfA	leokonda2	VN	VC	CB	CI
	4_WAdbVRdLM	Jimis Oxiavlist	FT	CL	OB	FA

## Anexo C: Código MATLAB de extracción de vectores de características

### Funciones de análisis

```
function output = calculo_vectores(audio,fs)
% Obtiene los vectores de características de una señal de audio
% audio: señal de audio obtenida mediante audioread
% fs: frecuencia de muestreo (obtenida mediante función audioread)

% Calcula los coeficientes estáticos
CE = analisis_estatico(audio,fs);
% Obtiene los coeficientes de grupo
CG = analisis_grupal(CE);
% Agrupa los coeficientes y devuelve los vectores de características en
% filas
output = [CE(:,1:42), CG];
end

function output = analisis_estatico(audio,fs)
% Obtiene los coeficientes estáticos del vector de características
% audio: señal de audio
% fs: frecuencia de muestreo
% Creamos barra de progreso
wb = waitbar(0,'Extrayendo los vectores de características...')
if size(audio,2) == 2
% Pasamos de stereo a mono
audio = audio(:,1) + audio(:,2);
end
if fs ~= 44100
% Muestreamos a fs = 44100
audio = interpolacion_fs(audio,fs);
end
% Filtrado-pre-énfasis
audio = filtrado_preenfasis(audio);
% Normalización volumen
audio = normalizar_volumen(audio);
% Aplicamos zero_padding para cuadrar con las ventanas
audio = zero_padding(audio);
% Cargamos el banco de filtros Mel y los coeficientes de la DCT
%(necesarios en llamadas a función coef_mfcc)
load filtrosMel.mat filtrosMel;
load coeficientes_DCT.mat coeficientes_DCT;
% Obtenemos el vector de frecuencias
vect_f = vector_frecuencias;
% Numero de harmónicos a considerar en la extracción de coeficientes
maxh = 15;
% Obtenemos la matriz M_i,j = i/j que permite calcular la posición del
% resto de armónicos de un pico considerando que este pudiera ser
% cualquiera de los 15 primeros armónicos, y que se utiliza en la
% función anidada analisis_harmonicos (para optimizar el código)
matriz_harmonicos = 1:maxh;
matriz_harmonicos = meshgrid(matriz_harmonicos,matriz_harmonicos);
matriz_harmonicos = matriz_harmonicos./matriz_harmonicos';
% Obtenemos el número de ventanas largas (V = 2)
N = max_segmentos(audio,2);
% Obtenemos el número de ventanas cortas (V = 1)
M = max_segmentos(audio,1);
% Inicializamos vector extractos (suponiendo 12 notas por ventana)
F = zeros(round(12*N),47);
c = 0;
% Bucle for a través de cada ventana larga
for k=1:N
% Extracción segmento
x = extraccion_segmento(audio,k,2);
% Enventanamiento
x = enventanado_hamming(x);
% Obtiene FFT
X = respuesta_frecuencial_normalizada(x);
% Obtiene las frecuencias fundamentales de baja frecuencia
[f0,ene] = frecuencias_fundamentales(X,2,[1,matriz_harmonicos]);
% Bucle for a través de cada frecuencia fundamental
for f=1:length(f0)
c = c + 1;
F(c,1) = f0(f); % COL 1 frecuencia fundamental
F(c,43) = k; % COL 43 número ventana
end
end
end
```

```

F(c,44) = numero_semitono(f0(f)); % COL 44 numero semitono
F(c,45) = ene(f); % COL 45 energia nota
Xesp = espectro_especifico(X,f0(f),1,vect_f); % espectro especifico nota
F(c,[2:22,47]) = caracteristicas_harmonicos(Xesp,f0(f),vect_f);
% COL 2-14 amplitud relativa harmónicos 1 a 13
% COL 15 suma total amplitudes harmónicos 1 a 15
% COL 16 crest factor máximo
% COL 17 centroide espectral (relativo)
% COL 18-19 segundo y tercer triestimulo
% COL 20 extensión espectral
% COL 21-22 pendiente espectral y error de ajuste
% COL 47 amplitud estimada nota
F(c,23:42) = mfcc_coef(Xesp,filtrosMel,coeficientes_DCT);
% COL 23-42 coeficientes MFCC 1 a 20
end
% Actualización barra de progreso
if rem(k,100) == 0
    try
        waitbar(k/(N+M),wb,'Extrayendo los vectores de características...');
    catch
    end
end
end
% Eliminamos filas no usadas
F(c+1:end,:) = [];
% Eliminamos vectores que no formen parte de un grupo de 5 vectores
% consecutivos con el mismo semitono y numeramos los grupos
% encontrados empezando por el 1 en COL 46
F = agrupar_vectores(F,1);
% Inicializamos vector extractos (suponiendo 12 notas por ventana)
G = zeros(round(12*M),36);
c = 0;
% Bucle for a través de cada ventana corta
for k=1:M
    % Extracción segmento
    x = extraccion_segmento(audio,k,1);
    % Enventanamiento
    x = enventanado_hamming(x);
    % Obtiene FFT
    X = respuesta_frecuencial_normalizada(x);
    % Obtenemos la ventana equivalente de baja frecuencia
    if k == 1
        v = 1;
    elseif k == M || k == M - 1
        v = N;
    else
        v = k - 1;
    end
    % Obtenemos las frecuencias fundamentales de baja frecuencia
    flf = F(F(:,43) == v,1);
    % Obtiene las frecuencias fundamentales de alta frecuencia
    [f0,ene] = frecuencias_fundamentales(X,1,flf,matriz_harmonicos);
    % Bucle for a través de cada frecuencia fundamental
    for f=1:length(f0)
        c = c + 1;
        G(c,1) = f0(f); % COL 1 frecuencia fundamental
        G(c,43) = k; % COL 43 numero ventana
        G(c,44) = numero_semitono(f0(f)); % COL 44 numero semitono
        G(c,45) = ene(f); % COL 45 energia nota
        Xesp = espectro_especifico(X,f0(f),1,vect_f); % espectro especifico nota
        G(c,[2:22,47]) = caracteristicas_harmonicos(Xesp,f0(f),vect_f);
        % COL 2-14 amplitud relativa harmónicos 1 a 13
        % COL 15 suma total amplitudes harmónicos 1 a 15
        % COL 16 crest factor máximo
        % COL 17 centroide espectral (relativo)
        % COL 18-19 segundo y tercer triestimulo
        % COL 20 extensión espectral
        % COL 21-22 pendiente espectral y error de ajuste
        % COL 47 amplitud estimada nota
        G(c,23:42) = mfcc_coef(Xesp,filtrosMel,coeficientes_DCT);
        % COL 23-42 coeficientes MFCC 1 a 20
    end
end
% Actualización barra de progreso
if rem(k,100) == 0
    try
        waitbar((N+k)/(N+M),wb,'Extrayendo los vectores de características...');
    catch
    end
end

```

```

        end
    end
end
% Eliminamos filas no usadas
G(c+1:end,:) = [];
% Eliminamos vectores que no formen parte de un grupo de 5 vectores
% consecutivos con el mismo semitono y numeramos los grupos
% encontrados (en COL 46) empezando por el siguiente numero disponible a los
% encontrados en las frecuencias bajas.
if size(F,1) > 0
    G = agrupar_vectores(G,max(F(:,46))+1);
else
    G = agrupar_vectores(G,1);
end
% Cierre barra de progreso
try
    close(wb);
catch
end
output = [F;G];
end

```

```

function output = analisis_grupal(coef_estaticos)
% Obtiene los coeficientes estáticos del vector de características
% coef_estaticos: salida de la funcion analisis_estatico (46 columnas)

% Numero de vectores
N = size(coef_estaticos,1);
% Inicializamos la matriz de características de grupo
C = zeros(N,28);
% Obtenemos el numero de grupos existentes
G = max(coef_estaticos(:,46));
% Bucle para cada grupo
for k = 1:G
    % Extraemos los indices de vectores de este grupo
    Gind = find(coef_estaticos(:,46) == k);
    % Numero de ventanas en el grupo
    M = length(Gind);
    % COL 1 desv.estándar frecuencia fundamental
    C(Gind,1) = std(coef_estaticos(Gind,1));
    % COL 2-10 media amplitud relativa harmonicos (1-9)
    for h=2:10
        C(Gind,h) = mean(coef_estaticos(Gind,h));
    end
    % COL 11-12 medias triestímulos 2 y 3
    C(Gind,11) = mean(coef_estaticos(Gind,18));
    C(Gind,12) = mean(coef_estaticos(Gind,19));
    % COL 13-14-15 Desviaciones típicas triestímulos 1-2-3
    C(Gind,13) = std(coef_estaticos(Gind,2));
    C(Gind,14) = std(coef_estaticos(Gind,18));
    C(Gind,15) = std(coef_estaticos(Gind,19));
    % COL 16-17 Media y desviación típica del espectroide central
    C(Gind,16) = mean(coef_estaticos(Gind,17));
    C(Gind,17) = std(coef_estaticos(Gind,17));
    % COL 18-19 Media y desviación típica del espectroide central absoluto
    C(Gind,18) = mean(coef_estaticos(Gind,17).*coef_estaticos(Gind,1));
    C(Gind,19) = std(coef_estaticos(Gind,17).*coef_estaticos(Gind,1));
    % COL 20-21 Media y desviación típica de la extensión espectral
    C(Gind,20) = mean(coef_estaticos(Gind,20));
    C(Gind,21) = std(coef_estaticos(Gind,20));
    % COL 22-23 Media y desviación típica de la pendiente espectral
    C(Gind,22) = mean(coef_estaticos(Gind,21));
    C(Gind,23) = std(coef_estaticos(Gind,21));
    % COL 24 Media del error ajuste pendiente espectral
    C(Gind,24) = mean(coef_estaticos(Gind,22));
    % Calculamos el flujo espectral de cada vector
    SF = flujo_espectral(coef_estaticos(Gind,2:14));
    % COL 25-26 Media y desviación típica del flujo espectral
    C(Gind,25) = mean(SF(2:end));
    C(Gind,26) = std(SF(2:end));
    % Extraemos las amplitudes
    A = coef_estaticos(Gind,47);
    % COL 27-28 Pendiente local y su error (coef.estáticos nuevos)
    for j = 1:M
        % Extraemos el vector fila de amplitudes desde 7 muestras
        % anteriores hasta 7 muestras posteriores
        AL = A(max(1,j-7):min(M,j+7));
    end
end

```

```

        [AP,APE] = pendiente_coeficientes(AL');
        C(Gind(j),27) = AP;
        C(Gind(j),28) = APE;
    end
end
output = C;
end

function output = agrupar_vectores(input,num_inicio)
% Elimina aquellos vectores de entrada que no formen parte de un conjunto seguido (en el
tiempo) con el mismo numero de semitono fundamental. Además, asigna un nuevo campo llamado
numero de grupo, con la función de etiquetar a cada conjunto de vectores seguidos con el
mismo semitono con un único número de identificación.
% input: conjunto de vectores a agrupar o descartar
% num_inicio: numero inicial para contar grupos

    % Ordenamos la matriz M por el número de semitono y luego por numero
    % ventana
    input = sortrows(input,[44,43]);
    k = 1;
    num_grupo = num_inicio - 1;
    % Asignamos numero de grupo a los vectores aceptados
    % Ponemos numero de grupo = 0 a los vectores que se van a rechazar por
    % no formar parte de un grupo de 5 ventanas consecutivas con el mismo
    % semitono detectado.
    while k <= size(input,1)
        % Obtenimos semitono y num.ventana inicial a comprobar
        stn = input(k,44);
        nvt = input(k,43);
        % Observamos si las siguientes 5 filas tienen el mismo semitono y
        % son consecutivas
        if k <= size(input,1)-4
            % Semitonos 9 muestras posteriores
            stn_post = input(k+1:k+4,44);
            % Dif num. ventanas 9 muestras posteriores]
            nvt_dif_post = input(k+1:k+4,43) - nvt - (1:4)';
            % Condicion mantener muestras como nota
            if all(stn_post == stn) && all(nvt_dif_post == 0)
                % Nuevo numero de grupo
                num_grupo = num_grupo + 1;
                % Avanza k hasta nuevo semitono asignando el num_grupo
                while input(k,44) == stn && k <= size(input,1)
                    input(k,46) = num_grupo;
                    k = k + 1;
                    if k > size(input,1) || input(k,43) - input(k-1,43) ~= 1
                        break
                    end
                end
                % Condición eliminar muestras de esta nota por no tener 5
                % consecutivas
            else
                input(k,46) = 0;
                k = k + 1;
            end
        else
            input(k,46) = 0;
            k = k + 1;
        end
    end
    % Elimina los vectores a descartar
    input(input(:,46) == 0,:) = [];
    output = input;
end

```

## Funciones de apoyo

```

function output = interpolacion_fs(audio,fs)
% Remuestrea un archivo de sonido para que pase a tener una frec. muestreo de 44100Hz.
% audio: señal de audio
% fs: frecuencia de muestreo de la señal de audio

    % Frecuencia de muestro del sistema
    f = 44100;
    % Ratio racional frecuencia del sistema y frecuencia de muestreo
    [N,D] = rat(f/fs);

```

```

    % Remuestramos de acuerdo a la nueva tasa
    output = resample(audio,N,D);
end

function output = respuesta_frecuencial(input)
% Obtiene la respuesta en frecuencia (módulo) para frecuencias positivas de un filtro o
señal. El número de puntos del resultado es  $2^{14}+1 = 16385$  muestras.
% input: señal o respuesta impulsional del filtro h[n]

    % Número de muestras de la FFT (32768)
    M = 2^15;
    % Calculamos la FFT del filtro
    HF = fft(input,M);
    % Seleccionamos solo las frecuencias no negativas
    HF = HF(1:M/2+1);
    % Obtenemos el modulo
    HF = abs(HF);
    output = HF;
end

function output = respuesta_frecuencial_normalizada(input)
% Obtiene la respuesta en frecuencia (módulo) para frecuencias positivas de un filtro o señal
y normalizada (dividiendo por el numero de muestras de la señal y multiplicando por 2), para
que el valor del pico coincida con la amplitud de las señales sinusoidales. El número de
puntos del resultado es  $2^{14}+1 = 16385$  muestras.
% input: señal o respuesta impulsional del filtro h[n]

    % Número de muestras de la FFT (32768)
    M = 2^15;
    % Calculamos la FFT del filtro
    HF = fft(input,M);
    % Seleccionamos solo las frecuencias no negativas
    HF = HF(1:M/2+1);
    % Obtenemos el modulo
    HF = abs(HF);
    % Dividimos por el número de muestras de la señal y multiplicamos por 2
    % Así, una señal sinusoidal se convierte en un lóbulo con pico
    % igual a la amplitud
    HF = HF/length(input)*2;
    output = HF;
end

function output = vector_frecuencias()
% Genera el vector de frecuencias de  $2^{14}+1 = 16385$  puntos entre 0 y fs/2, con fs = 44.1 kHz

    % Numero de puntos FFT
    M = 2^15;
    % Numero de puntos frecuencias positivas FFT
    N = M/2+1;
    % Frecuencia de muestreo del sistema
    fs = 44100; % (Hz)
    % Obtenemos el vector de frecuencias
    output = linspace(0,fs/2,N);
end

function n = indice_frecuencia(f)
% Dada una frecuencia cualquiera obtiene la posición de la frecuencia más cercana dentro del
vector de frecuencias utilizado para todas las FFT del sistema (y que se obtiene mediante la
funcion vector_frecuencias). Si la frecuencia es superior a fs/2 retorna el máximo.
% f: frecuencia (Hz)

    % Configuración del sistema
    fs = 44100; % (Hz) frecuencia muestreo
    N = 2^15; % numero puntos FFT
    % Límite máximo
    f(f > fs/2) = fs/2;
    n = round(f/fs*N) + 1;
end

function n = numero_semitono(f)
% Obtiene el número de semitono más cercano asociado a una frecuencia determinada. Siendo
n = 0 la nota LA-4 (440 Hz). La salida, n, indica la nota correspondiente al número de
semitonos de distancia respecto a LA-4
% f: frecuencia

    n = round(12*log(f/440)/log(2));
end

```

```

function n = numero_nota(f)
% Obtiene un código asociado al nombre de la nota asociada a una frecuencia siendo LA=0,
LA#=1, SI=2, DO=3, DO#=4, RE=5, RE#=6, MI=7, FA=8, FA#=9,
% f: frecuencia

    semitono = numero_semitono(f);
    n = mod(semitono,12);
end

function output = espectro_especifico(FFT,f0,V,vect_freq)
% A partir de una FFT y una frecuencia fundamental f0, devuelve la versión de la FFT en la
que se han anulado todos los componentes que no esten en el entorno de los primeros 15
harmónicos de f0.
% Para ventana corta V=1, la distancia a un armónico debe ser < 64 Hz
% Para ventana larga V=2, la distancia a un armónico debe ser < 24 Hz
% FFT: FFT sobre la que limpiar el espectro
% f0: frecuencia fundamental que marcará los espacios del espectro a
% mantener
% V: tipo de ventana (V=1 corta, V=2 larga).
% vect_freq: vector de frecuencias de la FFT

    % Obtiene la distancia máxima a harmónicos
    if V == 1
        eps = 64/f0; % Hz (ventana corta)
    else
        eps = 24/f0; % Hz (ventana larga)
    end
    % Creamos vector de unos y ceros para limpiar la FFT
    g = vect_freq/f0;
    g1 = (g - fix(g))<=eps); % Frec. por encima de armonico
    g2 = (g - fix(g))>=1-eps); % Frec. por debajo de armonico
    g3 = (vect_freq < 15.5*f0); % Frec. por debajo de armonico 16
    g4 = (vect_freq > f0/2); % Frec. anteriores a armonico 1
    % Variable booleana indicando si se anula o mantiene un componente
    k = (g1 | g2) & g3 & g4;
    output = FFT.*k';
end

function [pks,locs] = picos_significativos(FFT)
% Realiza la misma tarea que la función findpeaks de MATLAB, pero descarta todos aquellos
picos cuyo valor esté por debajo de la suma de la media más la mitad de la desviación
estandar de las muestras. Devuelve amplitud y posición (índice) de los picos
% FFT: señal de la que se obtendrán los picos y posiciones significativas

    % Obtiene la media de la FFT
    mu = mean(FFT);
    % Obtiene la desviación típica de la FFT
    sigma = std(FFT);
    % Anula las muestras de la FFT por debajo de mu + sigma/2
    FFT(FFT < mu + sigma/2) = 0;
    % Devuelve los picos de la señal resultante
    [pks,locs] = findpeaks(FFT);
end

function [] = filtros_mel()
% Obtiene el banco de filtros Mel utilizado para el cálculos de los coeficientes MFCC
(función coef_mfcc). Se obtienen 40 filtros paso-banda que recorren el espectro entre los 0 y
% los 20000 Hz y se guarda su respuesta frecuencial en una matriz de 2 filas (una para los
filtros pares y otra para los impares) con el nombre filtrosMel.mat.

    % Frecuencia inicial
    f1 = 0;
    % Frecuencia final
    f2 = 20000;
    % Numero de filtros
    N = 40;
    % Funcion de conversion frecuencia a mel
    fmel = @(f) 1125*log(1 + f/700);
    % Funcion de conversion mel a frecuencia
    fHz = @(fm) 700*(exp(fm/1125) - 1);
    % Frecuencias-mel inicial y final
    f1m = fmel(f1);
    f2m = fmel(f2);
    % Bordas y centros de los filtros en frecuencia-mel (equiespaciados)
    fextm = linspace(f1m,f2m,N+2);

```

```

% Bordes y centros de los filtros en Hz
fext = fHz(fextm);
% Vector de frecuencias FFT
f = vector_frecuencias;
% Generamos el array de los filtros (40 filas, una por filtro) inicializado a 0
filtrosMelseparados = zeros(40,length(f));
% Asignamos valores fila por fila
for k = 1:40
    % Frecuencia inicial, maxima y final del filtro triangular
    fin = fext(k);
    fmax = fext(k+1);
    ffin = fext(k+2);
    for p = 1:length(f)
        if (fin <= f(p) && f(p) <= fmax)
            filtrosMelseparados(k,p) = (2*(f(p) - fin))/(fmax - fin);
        elseif (fmax < f(p) && f(p) <= ffin)
            filtrosMelseparados(k,p) = (2*(ffin - f(p)))/(ffin - fmax);
        end
    end
end
% Generamos el array para los filtros pares/impares (2 filas, una por paridad)
filtrosMel = zeros(2,length(f));%
% Obtenemos los filtros impares
k = 1:2:40;
filtrosMel(1,:) = sum(filtrosMelseparados(k,:));
% Obtenemos los filtros pares
k = 2:2:40;
filtrosMel(2,:) = sum(filtrosMelseparados(k,:));
% Guardamos los filtros calculados
save filtrosMel.mat filtrosMel
end

```

```

function [] = coeficientes_DCT()
% Obtiene el conjunto de valores cos(n*pi/M*(k-0.5)) con n entre 1 y 20 y M entre 1 y 40 que
se utilizan para realizar la transformada discreta del coseno durante el cálculo de los MFCC
% Se programa para evitar calcularlos en cada iteración
% Los coeficientes se guardan como archivo coeficientes_DCT.mat

[k,n] = meshgrid((1:40),(1:20));
coeficientes_DCT = cos(n*pi./40.*(k-0.5));
% Guardamos los filtros calculados
save coeficientes_DCT.mat coeficientes_DCT
end

```

```

function output = transformada_DCT(X,coeficientes_DCT)
% Obtiene los coeficientes 1 a 20 de la transformada DCT de una secuencia de longitud N,
según el método de la norma ETSI ES 201 108.
% X: secuencia sobre la que obtener la transformada
% coeficientes_DCT: matriz de valores cos(n*pi/M*(k-0.5)) utilizados en el cálculo de la DCT

% Replicamos la entrada 20 veces (una por coeficiente)
X = repmat(X',20,1);
% Calculamos los coeficientes
output = sum(X.*coeficientes_DCT,2)';
end

```

## Funciones de preprocesado

```

function output = filtrado_preenfasis(input)
% Realiza el filtrado_preenfasis de la señal input.
% input: señal a filtrar

% Obtenemos la respuesta impulsional del filtro
filtro = [1 -0.95];
% Filtramos el señal mediante convolución
output = conv(input,filtro);
end

```

```

function output = normalizar_volumen(signal)
% Amplifica la señal de forma que la amplitud media RMS quede fijada en 0.05

amp_RMS = sqrt(mean(signal.^2));
output = 0.05/amp_RMS*signal;
end

```

## Funciones de segmentación

```
function output = extraccion_segmento(signal,n,V)
% Extrae las muestras del n-ésimo fragmento de una señal
% signal: señal original de la que extraer el segmento
% n: índice del segmento
% V: tipo de ventana (1: corta, 2: larga)

% Muestras inicial y final
ini = muestra_inicio_segmento(n,V);
fin = muestra_final_segmento(n,V);
% Extracción del segmento
output = signal(ini:fin);
end
```

```
function output = muestra_inicio_segmento(n,V)
% Obtiene el índice de la primera muestra del segmento n-ésimo del proceso de segmentación de la señal.
% n: número de segmento
% V: tipo de ventana sistema (1: corta, 2: larga)

% Configuración sistema
fs = 44100; % (Hz) frecuencia muestreo
if V == 1 % (ventana corta)
    Tx = 20e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.5; % solapamiento
else % (ventana larga)
    Tx = 50e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.8; % solapamiento
end
% Calcula muestra inicial
output = round((n - 1)*(1 - sigma)*Nx + 1);
end
```

```
function output = muestra_final_segmento(n,V)
% Obtiene el índice de la primera muestra del segmento n-ésimo del proceso de segmentación de la señal.
% n: número de segmento
% V: tipo de ventana sistema (1: corta, 2: larga)

% Configuración sistema
fs = 44100; % (Hz) frecuencia muestreo
if V == 1 % (ventana corta)
    Tx = 20e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.5; % solapamiento
else % (ventana larga)
    Tx = 50e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.8; % solapamiento
end
% Calcula muestra final
output = round((n - 1 + 1/(1-sigma))*(1-sigma)*Nx);
end
```

```
function output = max_segmentos(signal,V)
% Obtiene el número máximo de segmentos que se pueden extraer de una señal.
% signal: señal de la que calcular el número máximo de segmentos
% V: tipo de ventana sistema (1: corta, 2: larga)

% Configuración sistema
fs = 44100; % (Hz) frecuencia muestreo
if V == 1 % (ventana corta)
    Tx = 20e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.5; % solapamiento
else % (ventana larga)
    Tx = 50e-3; % (s) duración ventana
    Nx = fs*Tx; % muestras ventana
    sigma = 0.8; % solapamiento
end
% Número de muestras de la señal
```

```

N = length(signal);
% Calcula índice último segmento
output = (N - sigma*Nx)/(1 - sigma)/Nx;
end

```

```

function output = zero_padding(signal)
% Añade ceros al final de una señal hasta que su número de muestras sea múltiplo de la
distancia entre dos segmentos consecutivos. Para los dos tipos de ventana utilizados, la
distancia entre muestras consecutivas es de 441

```

```

% Distancia entre muestras consecutivas
D = 441;
% Longitud de la señal
N = length(signal);
% Residuo entre N y D
r = rem(N,D);
if r > 0
    % Muestras a añadir zero-padding
    Nzp = D - r;
    % Añadimos zero-padding (vector columna)
    signal = [signal;zeros(Nzp,1)];
end
% Caso extremo y raro: la longitud de la señal es inferior a 2205
% muestras (una ventana larga), añadimos más zero-padding.
if length(signal) < 2205
    Nzp = 2205 - length(signal);
    signal = [signal;zeros(Nzp,1)];
end

% Salida señal
output = signal;
end

```

```

function output = enventanado_hamming(input)
% Realiza un enventanado de tipo Hamming sobre la señal de entrada. Como la ventana de
Hamming no tiene por defecto el pico de su lóbulo principal normalizado 1, se divide la
ventana por su máximo según si usamos la ventana corta (882 muestras) o larga (2205) del
sistema. Estos coeficientes se han calculado previamente mediante la función
% picos_hamming y en el código se ponen directamente para evitar cálculos.
% input: señal a enventanar

```

```

% Longitud de la señal de entrada
N = length(input);
% Ventana de Hamming sin normalizar
H = hamming(N);
% Normalizamos ventana de Hamming
if N == 882
    H = H/0.539478458049887;
end
if N == 2205
    H = H/0.539791383219955;
end
% Realizamos el enventanamiento
output = input.*H;
end

```

```

function [] = picos_hamming()
% Obtiene el valor del pico del lóbulo principal de las ventanas de Hamming utilizadas en el
sistema, dividido por el número de muestras con la finalidad de no alterar los picos de los
armónicos al enventanar. Los valores obtenidos se usan en la función enventanado_hamming.

```

```

% Formato con más decimales
format long
% Configuración sistema
fs = 44100; % (Hz) frecuencia muestreo
% Ventana Hamming corta
Tx = 20e-3; % (s) duración ventana
Nx = fs*Tx; % muestras ventana
V = hamming(Nx); % ventana
TF = respuesta_frecuencial(V);
max_ventana_corta = max(TF)/Nx % Muestra coeficiente para ventanas cortas
% Ventana Hamming larga
Tx = 50e-3; % (s) duración ventana
Nx = fs*Tx; % muestras ventana
V = hamming(Nx); % ventana
TF = respuesta_frecuencial(V);

```

```

max_ventana_larga = max(TF)/Nx % Muestra coeficiente para ventanas largas
% Formato normal
format short
end

```

## Funciones de cálculo de características

```

function output = analisis_harmonicos(FFT,V,lf,M)
% A partir de una FFT obtiene una matriz de seis columnas
% output = [fh, k, A, f0, ns, nn] en las que para
% cada fila se obtiene la siguiente información relativa a las frecuencias fundamentales
detectadas en el extracto.
% fh: frecuencia de un pico significativo de la FFT tomado como referencia
% k: numero de harmonico (probamos cada pico con k=1:15)
% A: calculando el resto de harmonicos a partir de fh y k, se suman las
% energías de los 15 harmónicos. Resultado en decibelios.
% f0: posible frecuencia fundamental (fh/k)
% ns: numero de semitono asociado a f0
% nn: numero de nota asociado a f0
%
% FFT: funcion a analizar
% V: tipo de ventana segmentación (V=1 corta, V=2 larga)
% lf: frecuencias fundamentales de baja frecuencia encontradas
% con la ventana larga asociada a la ventana corta a analizar (solo se
% utiliza si V=1).
% M: matriz Mi,j = i/j utilizada para el analisis de harmónicos y que se calcula en la
función madre para optimizar tiempo ejecución.

% Anulamos el componente continuo (utilizaremos esa frecuencia f=0 para
% aquellos puntos que no podamos calcular o queramos ignorar).
FFT(1) = 0;
% Obtenemos los picos significativos de la FFT y sus posiciones
[~,lcs] = picos_significativos(FFT);
% Obtenemos el vector de frecuencias
f = vector_frecuencias;
% Frecuencia de muestreo del sistema
fs = 44100; % (Hz)
% Numero de harmonicos a considerar;
maxh = 15;
% Creamos los vectores de resultados, considerando 15 posiciones para
% cada posible pico
fh = zeros(maxh*length(lcs),1);
k = zeros(maxh*length(lcs),1);
A = zeros(maxh*length(lcs),1);
% Para cada pico calculamos las 15 hipótesis de posibles harmónicos
for i = 1:length(lcs)
% Matriz de posibles harmónicos
H = M*f(lcs(i));
% Si alguna de estas frecuencias es superior a fs/2 la ignoraremos
H(H > fs/2) = 1;
% Convertimos las frecuencias en posiciones de la FFT
H = indice_frecuencia(H);
% Evaluamos las posiciones H en la FFT
H = FFT(H).^2;
% Obtenemos la suma de las amplitudes de las posibles series de
% harmonicos
hA = sum(H,2);
% Obtenemos el resto de parámetros de este pico
hfh = f(lcs(i))*ones(15,1);
hk = (1:15)';
%Añadimos los resultados a los vectores generales
for j = 1:maxh
A((i-1)*maxh + j) = hA(j);
fh((i-1)*maxh + j) = hfh(j);
k((i-1)*maxh + j) = hk(j);
end
end
% Obtenemos el resto de vectores de salida
A = 10*log10(A);
f0 = fh./k;
ns = numero_semitono(f0);
nn = numero_nota(f0);

% Unimos todas las filas de resultados
M = [fh, k, A, f0, ns, nn];

```

```

% Obtenemos los semitonos de los picos de la FFT
semitonos = unique(numero_semitono(f(lcs)));
% Eliminamos aquellas filas de M cuya energía sea inferior a -30 dB
M(M(:,3) < -30,1) = 0;
% Eliminamos aquellas filas de M cuyo semitono de la frecuencia
% fundamental estimada no esté incluida en la lista
for i = 1:size(M,1)
    if all(semitonos ~= M(i,5))
        M(i,1) = 0;
    end
end
% Si la ventana es larga, eliminamos todas aquellas filas de M cuya
% frecuencia fundamental estimada sea igual o superior al semitono -9
% ya que esta ventana está pensada solo para detectar muy bajas.
% También se eliminan los semitonos inferiores a -33 ya que esta es la
% mínima frecuencia que el sistema podrá detectar
% frecuencias
if V == 2 % ventana larga (bajas frecuencias)
    for i = 1:size(M,1)
        if M(i,5) < -33 || M(i,5) > -10
            M(i,1) = 0;
        end
    end
end
% Si la ventana es corta se eliminan los semitonos inferiores a -9
% ya que esta es la mínima frecuencia que las ventanas cortas
% podrán detectar. También se eliminan los semitonos superiores a 50,
% ya que esta es la máxima frecuencia que se considera que un
% instrumento puede producir.
if V == 1 % ventana corta (medias-altas frecuencias)
    for i = 1:size(M,1)
        if M(i,5) < -9 || M(i,5) > 50
            M(i,1) = 0;
        end
    end
end
% Eliminamos las filas a descartar
M(M(:,1) == 0,:) = [];
% Ordenamos la matriz M por la suma de energías descendente
M = sortrows(M,-3);
% Eliminamos aquellas filas de M con misma nota de la frecuencia
% fundamental repetidas, excepto la que ofrece una suma de energías
% máxima.
i = 1;
while i <= size(M,1)
    nota = M(i,6);
    for j = (i+1):size(M,1)
        if M(j,6) == nota
            M(j,1) = 0;
        end
    end
    % Eliminamos las filas a descartar
    M(M(:,1) == 0,:) = [];
    i = i + 1;
end
% Eliminamos aquellas filas de M con frec.fundamental que es armónico
% de una frec.fundamental con mejor suma de energías
i = 1;
while i <= size(M,1)
    f0 = M(i,4); % frecuencia fundamental
    f0h = (1:10).*f0; % frecuencias 300 armónicos
    sth = numero_semitono(f0h); % semitonos armónicos
    j = i + 1;
    for j = (i+1):size(M,1)
        if any(sth == M(j,5))
            M(j,1) = 0;
        end
    end
    % Eliminamos las filas a descartar
    M(M(:,1) == 0,:) = [];
    i = i + 1;
end
% Si la ventana es corta, tendremos que eliminar aquellas filas de M
% con frec.fundamental que sea armónico de alguna de las frecuencias
% fundamentales de baja frecuencia hayadas por la ventana larga (lf)
% También se eliminan los semitonos inferiores a -9 ya que este es la
% mínima frecuencia que las ventanas cortas podrán detectar

```

```

if V == 1 % ventana corta (medias-altas frecuencias)
% Obtenemos la lista de semitonos que son armónicos de las
% frecuencias fundamentales de baja frecuencia detectadas
% El número máximo de frecuencias detectables es 12 (notas
% distintas)
lfh = ones(12,10);
for i = 1:length(lf)
    lfh(i,:) = (1:10).*lf(i); % frecuencias 300 armónicos lf
    % obtenemos los semitonos asociados
end
stfh = unique(numero_semitono(lfh));
% Eliminamos las filas candidatas con dicho num.semitono

for i = 1:size(M,1)
    if any(stfh == M(i,5))
        M(i,:) = 0;
    end
end
% Eliminamos las filas a descartar
M(M(:,1) == 0, :) = [];
end
output = M;
end

function output = caracteristicas_harmonicos(FFT, f0, f)
% Obtiene varias características del espectro de una nota.
% FFT: FFT de la señal a extraer las características
% f0: frecuencia fundamental asociada al vector de características
% f: vector de frecuencias de la FFT

% Vector de armónicos
fh = (1:15).*f0;
% Índice armónicos
ih = indice_frecuencia(fh);
% Valores absolutos de los armónicos
HA = FFT(ih)';
% HS: Suma amplitudes armónicos
HS = sum(HA);
% H(k) Amplitud relativa armónico k
H = HA./HS;
% AN: Amplitud estimada nota
AN = sqrt(sum(HA.^2));
% CF: Máximo valor del Crest Factor
CF = sqrt(2)*HS/AN;
% SC: Espectroide central normalizado
SC = f*FFT/sum(FFT)/f0;
% T2: Triestímulo 2 T3: Triestímulo 3
T2 = sum(HA(2:4))/HS;
T3 = sum(HA(5:15))/HS;
% SS: Extensión espectral
n = (1:15);
SS = sqrt(sum(H.*(n-SC).^2,2));
% SP: Pendiente espectral SPE: Error ajuste recta pendiente espectral
[SP, SPE] = pendiente_coeficientes(H);
output = [H(1:13), HS, CF, SC, T2, T3, SS, SP, SPE, AN];
end

function output = flujo_espectral(C)
% Obtiene el flujo_espectral obtenido a partir de varias series de armónicos incluidas como
filas de la matriz de entrada C. Dado que no se puede calcular para el primer vector, se
replicará el valor del flujo del segundo vector.
% C: serie de coeficientes de la que obtener su pendiente (conjunto vectores fila)

% Calculamos el flujo espectral para las filas 2:end de la entrada
SF = 1-sum(C(1:end-1, :).*C(2:end, :),2)./sqrt(sum(C(1:end-1, :).^2,2).*sum(C(2:end, :).^2,2));
% Duplicamos la primera fila para aproximar el flujo de la fila 1
SF = [SF(1, :); SF];
output = SF;
end

function [f0, ener] = frecuencias_fundamentales(FFT, V, lf, matriz_harmonicos)
% Obtiene las frecuencias fundamentales a partir de la FFT de un segmento (en vector columna)
y la suma de energías de sus armónicos. Tiene en consideración si la ventana es corta o
larga para incluir solo el rango de frecuencias aceptado para cada caso. Para las ventanas
cortas, además, es necesario indicar en la entrada las frecuencias fundamentales de baja
frecuencia ya encontradas en la respectiva ventana larga.

```

```

% FFT: función a analizar
% V: tipo de ventana segmentación (V=1 corta, V=2 larga)
% lf: frecuencias fundamentales de baja frecuencia encontradas con la ventana larga asociada
a la ventana corta a analizar (solo se utiliza si V=1).
% matriz_harmonicos: M_i,j = i/j utilizada para el analisis de harmónicos y que se calcula en
la función madre para optimizar tiempo ejecución.

    AH = analisis_harmonicos(FFT,V,lf,matriz_harmonicos);
    % Obtiene las frecuencias fundamentales
    f0 = AH(:,4); % las frecuencias fundamentales estan en columna 4 de AH
    % Obtiene la suma de energías de sus harmonicos
    ener = AH(:,3);
end

```

```

function output = mfcc_coef(FFT,filtrosMel,coeficientes_DCT)
% Obtiene los 20 primeros coeficientes MFCC a partir de un espectro normalizado dado, según
el método de la norma ETSI ES 201 108
% FFT: espectro del que obtener los MFCC
% filtrosMel: banco de filtrosMel utilizados en el cálculo de los MFCC, la primera fila
contiene la superposición de los filtros impares 1,3,5,...,39. La segunda fila contiene la
superposición de los filtros pares 2,4,6,...,40. Las muestras iniciales y finales de la FFT
asociadas a cada filtro se pueden consultar en la función rango_filtros_Mel.
% coeficientes_DCT: coeficientes para el cálculo de la transformada
% discreta del coseno.

% Obtenemos el espectro de potencia
FFT = FFT.^2;
% Multiplicamos el espectro de potencia por los filtros Mel impares
Pimp = FFT'.*filtrosMel(1,:);
% Aplicamos el espectro a cada uno de los filtros
% Multiplicamos el espectro de potencia por los filtros Mel pares
Ppar = FFT'.*filtrosMel(2,:);
% Obtenemos la energía de la salida de cada filtro
S = zeros(40,1);
S(1) = sum(Pimp(1:94));
S(2) = sum(Ppar(46:147));
S(3) = sum(Pimp(94:205));
S(4) = sum(Ppar(147:267));
S(5) = sum(Pimp(205:335));
S(6) = sum(Ppar(267:408));
S(7) = sum(Pimp(335:488));
S(8) = sum(Ppar(408:575));
S(9) = sum(Pimp(488:669));
S(10) = sum(Ppar(575:771));
S(11) = sum(Pimp(669:882));
S(12) = sum(Ppar(771:1003));
S(13) = sum(Pimp(882:1134));
S(14) = sum(Ppar(1003:1277));
S(15) = sum(Pimp(1134:1431));
S(16) = sum(Ppar(1277:1599));
S(17) = sum(Pimp(1431:1782));
S(18) = sum(Ppar(1599:1980));
S(19) = sum(Pimp(1782:2195));
S(20) = sum(Ppar(1980:2429));
S(21) = sum(Pimp(2195:2682));
S(22) = sum(Ppar(2429:2958));
S(23) = sum(Pimp(2682:3258));
S(24) = sum(Ppar(2958:3583));
S(25) = sum(Pimp(3258:3936));
S(26) = sum(Ppar(3583:4320));
S(27) = sum(Pimp(3936:4736));
S(28) = sum(Ppar(4320:5189));
S(29) = sum(Pimp(4736:5680));
S(30) = sum(Ppar(5189:6214));
S(31) = sum(Pimp(5680:6794));
S(32) = sum(Ppar(6214:7424));
S(33) = sum(Pimp(6794:8108));
S(34) = sum(Ppar(7424:8851));
S(35) = sum(Pimp(8108:9658));
S(36) = sum(Ppar(8851:10534));
S(37) = sum(Pimp(9658:11486));
S(38) = sum(Ppar(10534:12519));
S(39) = sum(Pimp(11486:13642));
S(40) = sum(Ppar(12519:14862));
% Obtenemos los logaritmos base 10 de las energías de los filtros con un minimo de -22.
Slog = max(log10(S),-22);
% Finalmente, aplicando la transformada DCT obtenemos los coeficientes MFCC

```

```

output = transformada_DCT(Slog,coeficientes_DCT);
end

function [] = rango_filtros_Mel()
% Obtiene los indices inicial y final del vector de frecuencias de la FFT
% que corresponden a cada filtro Mel.

% Frecuencia inicial
f1 = 0;
% Frecuencia final
f2 = 20000;
% Numero de filtros
N = 40;
% Funcion de conversion frecuencia a mel
fmel = @(f) 1125*log(1 + f/700);
% Funcion de conversion mel a frecuencia
fHz = @(fm) 700*(exp(fm/1125) - 1);
% Frecuencias-mel inicial y final
f1m = fmel(f1);
f2m = fmel(f2);
% Bordes y centros de los filtros en frecuencia-mel (equiespaciados)
fextm = linspace(f1m,f2m,N+2);
% Bordes y centros de los filtros en Hz
fext = fHz(fextm);
% Tabla indices inicial y final
k = 1:40;
ind_inicial = indice_frecuencia(fext(k));
ind_final = indice_frecuencia(fext(k+2));
table(k',ind_inicial',ind_final','VariableNames',{'Filtro Mel','inicio','fin'})
end

function [pendiente,error_medio] = pendiente_coeficientes(C)
% Obtiene la pendiente de la recta de regresión de las series de coeficientes (vectores fila)
% indicada en la entrada, así como el error medio de la recta de regresión.

% C: serie de coeficientes de la que obtener su pendiente (conjunto vectores fila)

% Sistema lineal  $K1 \cdot n + K2 = coef(n)$  o bien  $A * K = C$ 
% donde  $A = [1 \ 1; 2 \ 1; 3 \ 1; \dots \ N \ 1]$ 
%  $K = [K1 \ K0]$ 
%  $C = [coef(1) \ coef(2) \ coef(3) \ \dots \ coef(N)]$ 

% Longitud C
N = size(C,2);
% Matriz A
A = [(1:N)',ones(N,1)];
K = A\C';

% La pendiente del sistema es el coeficiente K1
pendiente = (K(1,:))';
% Obtenemos las aproximaciones que ofrece la recta de regresión
Cestim = A*K;
% Calculamos los errores
Cerror = abs(Cestim' - C);
% Obtenemos el error medio
error_medio = (sum(Cerror')/N)';
end

```

## Anexo D: Código MATLAB del prototipo e interfaz gráfica

Esta sección contiene el código de la MATLAB App que se ha diseñado como prototipo e interfaz gráfica de pruebas. Las secciones comentadas en inglés han sido generadas por defecto y de forma automática por la aplicación *APP Designer* de MATLAB.

Este programa está relacionado con el código de extracción de características (Anexo C) a raíz de la llamada a la función `calculo_vectores`.

Para que el programa funcione es necesario entrenar primeramente el clasificador SVM Cubic y asignar al modelo generado (variable tipo `struct`) el nombre `trainedModel13K`, y, a su vez, guardar el modelo con el nombre de archivo `trainedModel13K.mat`. En el instalable entregado con este TFG se incluye ya el clasificador entrenado con el corpus de archivos de entrenamiento (Anexo A).

```
classdef app1 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        InstrumentClassifierUIFigure matlab.ui.Figure
        RespuestaPanel matlab.ui.container.Panel
        Respuesta matlab.ui.control.TextArea
        Grafico matlab.ui.control.UIAxes
        AnalizarnuevoarchivoButton matlab.ui.control.Button
        NumInstrumentos matlab.ui.container.ButtonGroup
        Button1 matlab.ui.control.RadioButton
        Button2 matlab.ui.control.RadioButton
        Button3 matlab.ui.control.RadioButton
        Button4 matlab.ui.control.RadioButton
        TipoAnalisis matlab.ui.container.ButtonGroup
        ButtonRapido matlab.ui.control.RadioButton
        ButtonCompleto matlab.ui.control.RadioButton
    end

    % Variables de programa
    properties (Access = private)
        fileWAV; % Nombre archivo .wav a analizar
        pathWAV; % Path archivo .wav a analizar
        resultados = zeros(1,26); % Vector resultados
        categorias; % Etiquetas categorías
        numIns = 1; % Numero de instrumentos
        vectores = []; % Vectores de características extraídos
        clasificador; % Estructura clasificador
        tipoAnalisis = 0; % Tipo análisis 0: rapido 1: completo
        error_archivo; % true si el archivo a analizar no es válido/definido
    end

    methods (Access = private)

        function [] = actualiza_plot(app)
            % Actualiza el gráfico

            if sum(app.resultados) ~= 0
                % Calcula la ratio de vectores clasificados en cada
                % categoría
                app.resultados = app.resultados/sum(app.resultados);
            end
            % Dibujo del gráfico
            barh(app.Grafico,app.resultados*100)
            app.Grafico.XLim = [0 100];
            app.Grafico.XGrid = 'on';
            app.Grafico.XMinorGrid = 'on';
            app.Grafico.XTick = 0:10:100;
            app.Grafico.XLabel.String = "Vectores clasificados (%)";
            app.Grafico.YLabel.String = "";
            app.Grafico.YTick = 1:26;
            app.Grafico.YTickLabel = app.categorias.etiquetas;
            % Actualización respuesta APP
            actualiza_respuesta(app);
        end
    end
end
```

```

end

function [] = actualiza_respuesta(app)
% Actualiza la respuesta de la APP

if sum(app.resultados) == 0
% Caso sin respuesta
app.Respuesta.Value = "";
else
% Obtención 4 categorías con más vectores clasificados
[~,r] = maxk(app.resultados,4);
s1 = app.categorias.etiquetas(r(1));
s2 = app.categorias.etiquetas(r(2));
s3 = app.categorias.etiquetas(r(3));
s4 = app.categorias.etiquetas(r(4));
% Muestra respuesta según numero instrumentos solicitado
switch app.numIns
case 1
app.Respuesta.Value = s1;
case 2
app.Respuesta.Value = [s1,s2];
case 3
app.Respuesta.Value = [s1,s2,s3];
case 4
app.Respuesta.Value = [s1,s2,s3,s4];
end
end
end

function [] = calcula_vectores(app,audio,fs)
% Obtiene los vectores de características e inicia el proceso
% de clasificación
app.vectores = calculo_vectores(audio,fs);
clasifica_vectores(app);
end

function [] = clasifica_vectores(app)
% Clasifica los vectores
num_vect = size(app.vectores,1);
if num_vect == 0
% Caso sin vectores disponibles
msgbox('No se ha podido extraer ningún vector de características. Prueba con
un audio de mayor calidad');
else
% Creación barra de progreso
wb = waitbar(0,'Clasificando el archivo...');
clasificacion = [];
% Clasificamos los resultados de 100 en 100 para actualizar
% barra de progreso
max_i = ceil(num_vect/100);
for i = 1:max_i
ind_min = (i-1)*100 + 1;
ind_max = min(num_vect,i*100);
vect_parcial = app.vectores(ind_min:ind_max,:);
clasif_parcial =
app.clasificador.trainedModel3K.predictFcn(vect_parcial);
clasificacion = [clasificacion;clasif_parcial]; %#ok<AGROW>
try
% Actualiza la barra de progreso
waitbar(i/max_i,wb,'Clasificando el archivo...');
catch
end
end
for k = 1:26
% Contamos los vectores clasificados en cada categoría
app.resultados(k) = sum(clasificacion == k);
end
try
% Cierra la barra de progreso
close(wb);
catch
end
% Calcula la ratio de vectores clasificados en cada
% categoría
app.resultados = app.resultados/sum(app.resultados);
% Actualiza el gráfico
actualiza_plot(app);

```

```

end

end

function [] = analizar_archivo(app)
% Analiza el archivo WAV seleccionado
if isequal(app.fileWAV,0)

else
try
app.error_archivo = false;
[audio,fs] = audioread([app.pathWAV,app.fileWAV]);
catch ME
% Caso error archivo seleccionad
app.error_archivo = true;
msgbox(ME.message)
app.Grafico.Title.String = app.fileWAV;
app.resultados = zeros(1,26);
app.actualiza_plot
end
if app.error_archivo == false
% Caso archivo correcto
% Actualiza nombre gráfico y lo inicia a 0
app.Grafico.Title.String = app.fileWAV;
app.resultados = zeros(1,26);
app.actualiza_plot
drawnow
if app.tipoAnálisis == 0
% Caso análisis rápido
% Reducción del audio a 60 minutos de duración
% mediante 10 segmentos de 6 segundos distribuidos
N = size(audio,1); % Longitud audio
M = round(N/10); % Redondeo N/10
T = round(6*fs); % Numero muestras 6 segundos
if N > 11*T % Solo se reduce si el audio dura más de 66 segundos
muestras = []; % Vector muestras a mantener
for k = 0:9
muestras = [muestras, (M*k+1):(M*k+T)]; %#ok<AGROW>
end
audio = audio(muestras,:);
end
end
calcula_vectores(app,audio,fs);
end

end

end

end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
app.Respuesta.Value = " ";
app.NumInstrumentos.SelectedObject = app.Button1;
app.TipoAnálisis.SelectedObject = app.ButtonRapido;
app.fileWAV = " ";
app.pathWAV = " ";
app.categorias = load('etiquetas.mat');
app.actualiza_plot
app.Grafico.Title.String = app.fileWAV;
app.clasificador = load('trainedModel3K.mat');
app.error_archivo = true; % inicio sin archivo definido
end

% Button pushed function: AnalizarnuevoarchivoButton
function AnalizarnuevoarchivoButtonPushed(app, event)
[fileSel,pathSel] = uigetfile('.wav','Seleccione archivo audio a analizar');
if isequal(fileSel,0)

else
app.fileWAV = fileSel;
app.pathWAV = pathSel;
analizar_archivo(app)
end
end

```

```

end

% Selection changed function: NumInstrumentos
function NumInstrumentosSelectionChanged(app, event)
    selectedButton = app.NumInstrumentos.SelectedObject;
    switch selectedButton
        case app.Button1
            app.numIns = 1;
        case app.Button2
            app.numIns = 2;
        case app.Button3
            app.numIns = 3;
        case app.Button4
            app.numIns = 4;
    end
    actualiza_respuesta(app)
end

% Selection changed function: TipoAnálisis
function TipoAnálisisSelectionChanged(app, event)
    selectedButton = app.TipoAnálisis.SelectedObject;
    switch selectedButton
        case app.ButtonRapido
            app.tipoAnálisis = 0;
            tipo = 'rápido';
        case app.ButtonCompleto
            app.tipoAnálisis = 1;
            tipo = 'completo';
    end
    if app.error_archivo == false
        answer = questdlg(['¿Deseas volver a analizar el archivo ', app.fileWAV, '
mediante análisis ', tipo, '?'], 'Cambio tipo de análisis', 'Sí', 'No', 'No');
        switch answer
            case 'Sí'
                app.resultados = zeros(1,26);
                app.actualiza_plot
                drawnow
                analizar_archivo(app)
            end
        end
    end

end

end

end

% Component initialization
methods (Access = private)
    % Create UIFigure and components
    function createComponents(app)
        % Create InstrumentClassifierUIFigure and hide until all components are created
        app.InstrumentClassifierUIFigure = uifigure('Visible', 'off');
        app.InstrumentClassifierUIFigure.Position = [100 100 720 492];
        app.InstrumentClassifierUIFigure.Name = 'Instrument Classifier';
        % Create RespuestaPanel
        app.RespuestaPanel = uipanel(app.InstrumentClassifierUIFigure);
        app.RespuestaPanel.TitlePosition = 'centertop';
        app.RespuestaPanel.Title = 'Respuesta';
        app.RespuestaPanel.Position = [21 333 170 120];
        % Create Respuesta
        app.Respuesta = uitextarea(app.RespuestaPanel);
        app.Respuesta.Editable = 'off';
        app.Respuesta.HorizontalAlignment = 'center';
        app.Respuesta.FontSize = 20;
        app.Respuesta.FontColor = [0 0 1];
        app.Respuesta.BackgroundColor = [0.8 0.8 0.8];
        app.Respuesta.Position = [1 1 170 99];
        % Create Grafico
        app.Grafico = uiaxes(app.InstrumentClassifierUIFigure);
        title(app.Grafico, 'Title')
        xlabel(app.Grafico, 'X')
        ylabel(app.Grafico, 'Y')
        app.Grafico.Position = [218 16 491 462];
        % Create AnalizarnuevoarchivoButton
        app.AnalizarnuevoarchivoButton = uibutton(app.InstrumentClassifierUIFigure,
'push');
        app.AnalizarnuevoarchivoButton.ButtonPushedFcn = createCallbackFcn(app,
@AnalizarnuevoarchivoButtonPushed, true);

```

```

    app.AnalizarnuevoarchivoButton.Position = [21 45 170 60];
    app.AnalizarnuevoarchivoButton.Text = 'Analizar nuevo archivo';
    % Create NumInstrumentos
    app.NumInstrumentos = uibuttongroup(app.InstrumentClassifierUIFigure);
    app.NumInstrumentos.SelectionChangedFcn = createCallbackFcn(app,
@NumInstrumentosSelectionChanged, true);
    app.NumInstrumentos.TitlePosition = 'centertop';
    app.NumInstrumentos.Title = 'Número de instrumentos';
    app.NumInstrumentos.Position = [22 250 170 50];
    % Create Button1
    app.Button1 = uiradiobutton(app.NumInstrumentos);
    app.Button1.Text = '1';
    app.Button1.Position = [11 4 26 22];
    app.Button1.Value = true;
    % Create Button2
    app.Button2 = uiradiobutton(app.NumInstrumentos);
    app.Button2.Text = '2';
    app.Button2.Position = [49 4 24 22];
    % Create Button3
    app.Button3 = uiradiobutton(app.NumInstrumentos);
    app.Button3.Text = '3';
    app.Button3.Position = [85 4 26 22];
    % Create Button4
    app.Button4 = uiradiobutton(app.NumInstrumentos);
    app.Button4.Text = '4';
    app.Button4.Position = [122 4 29 22];
    % Create TipoAnálisis
    app.TipoAnálisis = uibuttongroup(app.InstrumentClassifierUIFigure);
    app.TipoAnálisis.SelectionChangedFcn = createCallbackFcn(app,
@TipoAnálisisSelectionChanged, true);
    app.TipoAnálisis.TitlePosition = 'centertop';
    app.TipoAnálisis.Title = 'Tipo de análisis';
    app.TipoAnálisis.Position = [21 136 170 74];
    % Create ButtonRápido
    app.ButtonRápido = uiradiobutton(app.TipoAnálisis);
    app.ButtonRápido.Text = 'Rápido';
    app.ButtonRápido.Position = [11 28 60 22];
    app.ButtonRápido.Value = true;
    % Create ButtonCompleto
    app.ButtonCompleto = uiradiobutton(app.TipoAnálisis);
    app.ButtonCompleto.Text = 'Completo';
    app.ButtonCompleto.Position = [11 6 73 22];
    % Show the figure after all components are created
    app.InstrumentClassifierUIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = app1

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.InstrumentClassifierUIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.InstrumentClassifierUIFigure)
    end
end
end
end

```