

# Herramienta para mejorar la afinación vocal en Android

Ramón Menchén, Ana M. Barbancho, M. Inés Herrero, Lorenzo J. Tardón, Isabel Barbancho  
menchenmartin@gmail.com, abp@ic.uma.es, ines@ic.uma.es, lorenzo@ic.uma.es, ibp@ic.uma.es

Universidad de Málaga, ATIC Research Group, ETSI Telecomunicación,  
Dpt. Ingeniería de Comunicaciones, Campus Teatinos, 29071 Málaga, Spain

**Resumen**—In this paper, a tool to improve vocal tuning in Android devices is presented. This application aims to offer exercises to practice and improve singing skills. The designed tool includes two main functionalities: sound synthesis, to provide with singing sound references, and fundamental frequency analysis, to analyze the sound and check if the user sings the right musical note. The well-known Yin algorithm has been selected to perform the fundamental frequency analysis.

Three different singing exercises are included: sing single notes, sing intervals and sing a note in order to complete a chord. The system also includes a graphical interface in which musical notation is employed to write down the singing sound. The system has been evaluated in order to test out its correct performance regarding both the analysis and synthesis of musical sounds.

## I. INTRODUCCIÓN

La música es el lenguaje universal por excelencia el cual ha sido utilizado por el ser humano independientemente de su nivel educativo y cultural [1]. Una de las formas básicas y naturales de interactuar con la música es cantar. Todo el mundo puede cantar, pero cantar de manera musicalmente afinada, requiere cierta práctica.

Actualmente, gracias a los avances tecnológicos el público general tiene acceso a múltiples dispositivos electrónicos como el ordenador, las tablets, los teléfonos móviles, etc. Esta disponibilidad ha supuesto una revolución en muchos ámbitos, incluido el de la educación, ámbito en el que cada día aparecen más aplicaciones cuyo objetivo es la docencia y, dentro de este ámbito, el aprendizaje musical también se ve favorecido.

En esta comunicación se presenta una herramienta para mejorar la afinación al cantar. Esta aplicación tiene como público objetivo tanto usuarios con muy pocos conocimientos musicales como usuarios con ciertos conocimientos, que quieran mejorar y practicar distintos aspectos de afinación de la voz. La gran ventaja de esta aplicación frente a muchas otras, es la interactividad que proporciona, dado que evalúa de manera interactiva la afinación de las notas cantadas por los usuarios.

La herramienta presentada en esta comunicación se ha desarrollado en Android. Android es un sistema operativo personalizable y fácil de utilizar que incluyen más de mil millones de dispositivos de todo el mundo [2]. La aplicación está pensada para usuarios de todas las edades e incluye tres tipos de ejercicios: cantar notas individuales, cantar intervalos y cantar una nota para completar un acorde.

La estructura de esta comunicación es la siguiente: en el Apartado II, se presenta la estructura de la herramienta desarrollada. A continuación, en el Apartado III, se presentan

los subsistemas de procesado de señal que ha sido necesario desarrollar para la aplicación. En el Apartado IV, se enumeran las clases Android utilizadas para poder desarrollar estos subsistemas. El Apartado V, describe el funcionamiento de la herramienta, basado en los distintos tipos de ejercicios musicales implementados. La evaluación del funcionamiento de la aplicación se muestra en el Apartado VI. Por último, en el Apartado VII, se presentan las conclusiones extraídas de este trabajo.

## II. DESCRIPCIÓN DEL SISTEMA

La estructura de herramienta para mejorar la afinación vocal, se presenta en la figura 1. Tal como se muestra en esta figura, la herramienta está dividida en tres tipos de ejercicios (Notas, Intervalos y Acordes) y dispone de dos subsistemas auxiliares (detección de frecuencia fundamental y generación de sonidos).

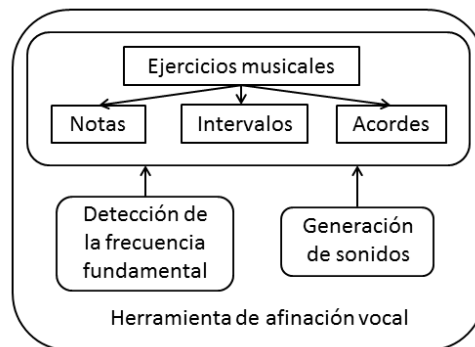


Fig. 1. Estructura de la herramienta de afinación vocal.

En una herramienta de afinación vocal, es fundamental detectar la nota que canta el usuario o, mejor dicho, la frecuencia fundamental para saber si está realizando de manera correcta o no el ejercicio. Además, dado que se quiere entrenar el oído del usuario así como mejorar sus habilidades para cantar en grupo, es fundamental disponer de un módulo de generación de sonidos afinados, que permita tener referencias sonoras exactas.

En el apartado siguiente (Apartado III), se detallan las características de los subsistemas de procesado de señal que se incluyen en la herramienta y que corresponden a la generación de sonidos por síntesis aditiva y a la detección de la frecuencia fundamental.

## III. SUBSISTEMAS DE PROCESADO DE SEÑAL

El desarrollo de nuestra aplicación interactiva para la mejora de la afinación de la voz, necesita dos componentes fundamentales: un sintetizador de sonidos afinados para proporcionar referencias sonoras correctas y un estimador de la

frecuencia fundamental de la nota que canta el usuario. Hay que destacar, que la estimación de la frecuencia fundamental de la nota que canta el usuario, es una de las características principales que distingue esta herramienta de muchas otras aplicaciones de aprendizaje musical que hay disponibles.

#### A. Síntesis de Sonido

Entre los distintos tipos de síntesis de audio que existen, para esta aplicación se ha seleccionado la síntesis aditiva [3]. En la síntesis aditiva el sonido se construye sumando, por cada armónico que tenga el sonido, un tono puro cuya amplitud varía en el tiempo según el armónico que se considere. Con este tipo de síntesis, se controla la posición de cada uno de los armónicos que componen los distintos sonidos musicales.

En esta aplicación, no es necesario imitar el timbre de ningún instrumento concreto. Lo único que se busca, es un sonido en el que sea sencillo identificar la frecuencia fundamental, que resulte agradable de escuchar y con la mínima complejidad posible.

Para realizar la síntesis aditiva de un sonido armónico, hay que decidir, fundamentalmente, tres elementos:

- El número de armónicos a sintetizar por cada sonido. Tras realizar múltiples pruebas, se ha comprobado heurísticamente que tres armónicos son suficientes para generar un sonido con un timbre agradable.
- La envolvente de la amplitud, en tiempo, de cada uno de los armónicos. Se ha decidido, por simplicidad, utilizar la misma para todos los armónicos. En la Fig. 2(b) se presenta la aproximación a la envolvente de amplitud que se ha utilizado. Esta aproximación es lineal a tramos de forma que su generación sea muy sencilla.
- La amplitud relativa entre los distintos armónicos. En la Fig. 2(a) se muestra el perfil de amplitud de los tres armónicos elegidos.

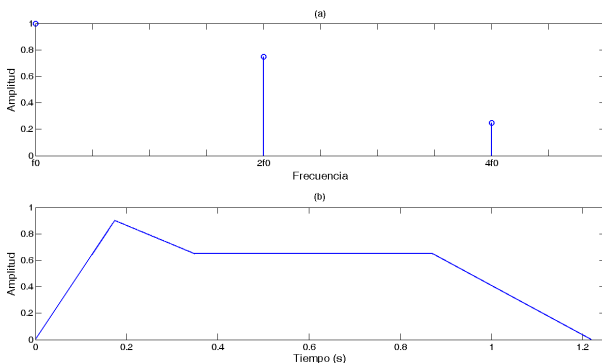


Fig. 2. a) Perfil de amplitud de los armónicos. b) Envolvente de amplitud de los armónicos

En el procesado correspondiente a esta síntesis aditiva de generación de sonido, se han empleado los siguientes parámetros de configuración:

- Frecuencia de muestreo: se ha utilizado 44100Hz para lograr una síntesis de buena calidad.
- Configuración de canal: la configuración de canal para el objeto *AudioTrack* será mono.
- Formato de audio: las muestras estarán codificadas en PCM de 16 bits, dado que es el formato con mayor calidad para el que se garantiza el buen funcionamiento en todos los dispositivos.

- Tamaño de buffer: es el propio de la clase *AudioTrack*.
- Modo de lectura: configurado en modo Streaming, para escuchar las señales en tiempo real mientras se van sintetizando.

Dado que el coste computacional de la síntesis de audio es muy alta, esta tarea se ha enviado a un hilo independiente, mediante la clase *Thread*. Además, se ha permitido que, si el usuario utiliza auriculares, pueda cantar a la vez que escucha el sonido sintetizado. Esta es una característica muy interesante tanto para afinar una nota simple como para cantar intervalos o completar acordes.

#### B. Estimación de la frecuencia fundamental

La estimación de la frecuencia fundamental de un sonido es una tarea que ha sido estudiada durante décadas [4] y, en especial, para el caso de la voz [5]. Actualmente, se encuentran en la literatura una gran variedad de métodos. De todos ellos, el que se ha seleccionado para esta aplicación es el algoritmo Yin [6], que fue desarrollado en 2002 por Cheveigné y Kawahara. Este algoritmo se basa en el método de la autocorrelación [7], pero introduce muchas mejoras en su funcionamiento. En la figura 3, se muestra el diagrama de bloques del algoritmo Yin utilizado. En este diagrama aparece reflejada la función de media normalizada acumulativa, que es la responsable de reducir los errores respecto a la función de autocorrelación clásica.

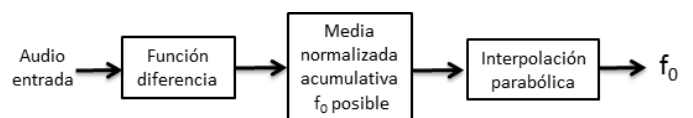


Fig. 3. Diagrama de bloques del algoritmo de Yin [6] utilizado.

Para la estimación de la frecuencia fundamental con este algoritmo, se han empleado los siguientes parámetros de configuración:

- Frecuencia de muestreo: se ha utilizado 11025Hz, ya que es una de la frecuencias recomendadas por Android, para todas las versiones y dispositivos, y es suficiente para detectar frecuencias fundamentales por debajo de 1046.5Hz, que es la correspondiente a un  $Do_6$ . Esta es la nota más aguda que se puede esperar de un cantante.
- Configuración de canal: la configuración de canal para el objeto *AudioRecord* es mono, dado que es cómo se capta la señal habitualmente.
- Formato de audio: las muestras estarán codificadas en PCM 16 bits.
- Modo de lectura: se configura en modo streaming de forma que se podrán analizar las muestras en tiempo real.

#### IV. PRINCIPALES CLASES ANDROID UTILIZADAS

Una vez identificados los subsistemas de procesado de señal necesarios en la aplicación, se han seleccionado las clases Android que mejor se adaptan a las necesidades de interacción con el audio de la herramienta desarrollada.

- Las principales clases utilizadas en la aplicación son [8]:
- *AudioTrack*: es la clase Java que permite manejar y reproducir recursos de Audio. Esta clase permite reproducir señales mientras se van introduciendo muestras

en el objeto que la instancia. Esta clase se utiliza para sintetizar los sonidos necesarios para la aplicación.

- *AsynTask*: es una clase abstracta, de la que debe heredar cualquier otra clase que pretenda derivar alguna tarea a un hilo de procesos, que no sea el hilo principal de la aplicación. Se utiliza para realizar la detección de tono de la nota cantada por el usuario en un hilo independiente.
- *Thread*: es una clase similar a la anterior, pero permite infinitos procesos independientes. Se utiliza para la síntesis de sonidos.
- *AudioRecord*: permite manejar señales que son captadas por el micrófono del dispositivo.
- *SharedPreferences*: es la interfaz para acceder y modificar los datos de preferencia de la aplicación, para los cuales sólo hay una única instancia que se comparte a lo largo de todas sus clases.

## V. FUNCIONAMIENTO DE LA APLICACIÓN

La aplicación se estructura en tres ejercicios de dificultad creciente, de manera que cada uno de ellos constituya un reto para el usuario:

- Notas individuales (sección A).
- Intervalos (sección B).
- Acordes musicales (sección C).

Estos tres niveles, dan lugar a tres pantallas diferentes pero con estética y lógica muy parecidas.

Por otra parte, es necesario almacenar una serie de variables propias de cada usuario como son las notas límite de su tesitura y los tipos de intervalos a practicar. En esta aplicación la tesitura se puede variar entre  $Mi_2$  y  $Do_6$ , para tener en cuenta tanto voces masculinas como femeninas.

En cuanto a los intervalos, se han seleccionado 11 tipos diferentes de intervalos (2º Mayor, 2º menor, 3º Mayor, 3º menor, 4º Justa, 5º Justa, 6º Mayor, 6º menor, 7º Mayor, 7º menor), que son los más utilizados en música [9]. Por lo tanto, se incluye una pantalla de opciones, que se representa en la Fig. 4 (a).

Además, como en cualquier aplicación habitual, se añade una pantalla de menú para seleccionar los ejercicios así como una pantalla de ayuda.

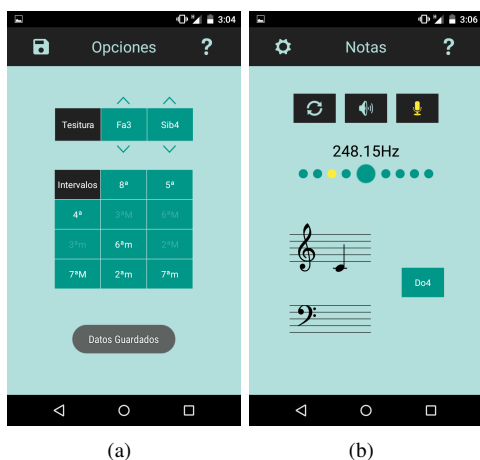


Fig. 4. (a) Pantalla de opciones de usuario. (b) Pantalla Notas mientras el usuario está cantando un  $Do_4$ .

En los tres ejercicios, el usuario ha de cantar una nota determinada y la aplicación ha de evaluar si lo hace de forma

correcta. Por ello, en todas las pantallas de ejercicios (Fig. 4(b), Fig. 6(a) y Fig. 6(b)) se ha incluido, además de la frecuencia en hertzios detectada, una barra de puntos que se van encendiendo en función de la frecuencia, con la que, gráficamente, el usuario puede ver cómo de cerca o lejos está de la afinación justa.

En la Fig. 5, se muestra la barra de detección de frecuencias, junto con el criterio de encendido de los puntos. Es de destacar, que se da por buena la afinación si se canta entre un tercio de semitono superior e inferior respecto a la nota exacta. El resto de puntos se encienden en los sucesivos tercios de tonos superiores e inferiores, excepto los puntos de las esquinas que no tienen límite inferior y superior, respectivamente, de filtrado. Utilizar el filtrado tomando como referencia el semitono, hace que la impresión de frecuencia se asemeje a la del oído humano, aunque esto haga que el ancho de los filtros varíe con la frecuencia fundamental a cantar, por la naturaleza logarítmica de las escalas y el oído humano.



Fig. 5. Barra de detección de frecuencias.

### A. Notas musicales individuales

El primer y más sencillo de los ejercicios de canto que se pueden realizar, consiste en escuchar una nota, y a continuación repetir, cantando, dicho sonido. Por lo tanto, el primer ejercicio de esta herramienta consiste en cantar una nota que ha sonado previamente.

El funcionamiento del ejercicio de notas individuales es el siguiente. Al usuario, tras haber configurado su tesitura en la pantalla de opciones, se le presenta en la pantalla principal una nota aleatoria dentro de su tesitura para cantar (Fig. 4 (b)). La nota a cantar aparece tanto escrita en pentagramas como con su nombre. A continuación, el usuario puede cantar de inmediato la nota pulsando el botón del micrófono o pulsar el botón del altavoz para escuchar previamente cómo suena. Con la barra de detección de frecuencias, se muestra lo cerca o lejos que está la nota de aquella que hay que cantar.

### B. Intervalos musicales

Cuando se quiere cantar correctamente, además de repetir una nota que ha sonado previamente, hay que ser capaz de interpretar una nota bien entonada, a partir del sonido de otra diferente que se ha interpretado previamente. En otras palabras, se trata de cantar intervalos [9]. Por lo tanto, el siguiente ejercicio de canto que se incluye en esta aplicación es el de cantar intervalos.

El funcionamiento del ejercicio de intervalos es el siguiente: configurar tanto la tesitura como el tipo de intervalos que quiere practicar, al usuario se le presentan en los pentagramas las notas que componen el intervalo a practicar (Fig. 6(a)). A la derecha de la pantalla se le indica también, el tipo de intervalo y la nota a cantar. A continuación, si se pulsa el botón altavoz se escucha la nota de referencia. Además, si se

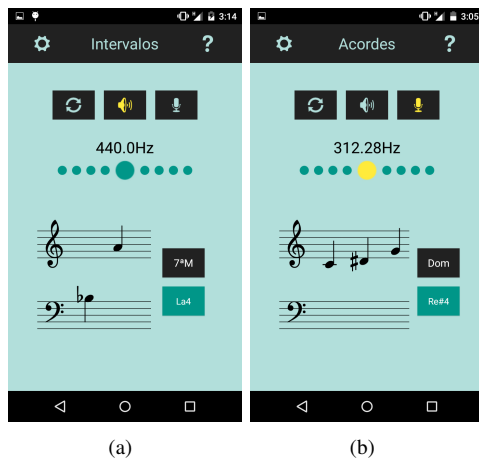


Fig. 6. (a) Pantalla Intervalos cuando se emite un Sib<sub>3</sub> de referencia para cantar un intervalo de séptima mayor. (b) Pantalla Acordes cuando suena el acorde de Do menor y hay que completarlo con la tercera (nota Re#<sub>4</sub>).

prefiere se puede pulsar el botón del nombre del intervalo y sonarán las dos notas a la vez. Cuando el usuario se siente preparado, pulsa el botón micrófono para cantar la nota que se le solicita. De nuevo, con la barra de frecuencias se comprueba la separación entre la nota cantada y la nota a interpretar.

### C. Acordes musicales

En los dos ejercicios anteriores, se han practicado notas sueltas o intervalos, pero lo normal en música es que suenen más de dos notas diferentes a la vez. Por tanto, un individuo debe ser capaz de cantar una nota escuchando dos o más notas diferentes de manera simultánea [9].

La suma de tres o más notas diferentes se denomina acorde. Existen muchos tipos de acordes, pero los más comunes son los perfectos mayores y menores, que son los más simples y consonantes que existen [10]. Por ello, este es el tipo de acordes que se van a practicar en esta aplicación.

El funcionamiento del ejercicio de acordes es el siguiente: Tras configurar la tesitura, al usuario se le presentan las tres notas que componen el acorde a practicar (Fig. 6(b)), de las cuales deberá cantar una de ellas. A la derecha de la pantalla se le indica, el tipo de acorde que forman y la nota que tiene que cantar. Si se pulsa el botón altavoz se escuchan las dos notas que acompañan a la nota a cantar. Además, si pulsa el botón del nombre del acorde, el usuario puede escuchar las tres notas a la vez. El usuario, al igual que en los casos anteriores, ha de pulsar el botón micrófono para cantar y podrá observar cómo está cantando la nota que completa el acorde, desde el punto de vista de la entonación.

## VI. EVALUACIÓN DEL SISTEMA

Se ha comprobado el correcto funcionamiento de todos los botones y enlaces entre pantallas, mediante la interacción con la aplicación en distintos dispositivos móviles. Sin embargo, para comprobar el funcionamiento de los subsistemas de procesamiento de señal, ha sido necesario utilizar el depurador de Eclipse para poder extraer los valores generados por la síntesis de sonido y los detectados por la aplicación para así analizarlos con una aplicación externa. En concreto, para estas comprobaciones se ha utilizado Matlab [11].

Se han sintetizado todas las notas y se han cantado todas las notas, dentro del rango de funcionamiento de la aplicación

(Mi<sub>2</sub> a Do<sub>6</sub>, comprobándose que tanto la síntesis como la detección es correcta. Para cantar las notas se han utilizado muestras de Vocaloid [12], para poder controlar la frecuencia exacta de las notas. En todos los casos se han obtenido resultados correctos tanto del sintetizador como del detector.

## VII. CONCLUSIONES

En esta comunicación se ha presentado una herramienta para mejorar la afinación vocal, programada en Android. Para conseguir una buena interactividad con el usuario, se incluye un sistema de síntesis de sonido que permite proporcionar referencias sonoras correctas, y un sistema para estimar en tiempo real la frecuencia fundamental de la nota que canta el usuario.

La técnica que se ha utilizado para sintetizar sonidos es la aditiva, dado que facilita controlar la posición de los armónicos del sonido a generar así como la envolvente de los mismos. El método que se ha implementado para estimar la frecuencia fundamental es el algoritmo Yin [6], que ha permitido tener una estimación robusta de las notas cantadas por los usuarios de la aplicación.

La aplicación se ha estructurado en tres ejercicios de dificultad creciente: notas individuales, intervalos y acordes musicales. La selección de estos ejercicios hace que la herramienta diseñada sea útil para usuarios con diversos niveles de canto. Por último, se ha comprobado que todas las opciones y funcionalidades de la aplicación, incluyendo los esquemas de procesamiento de señal, funcionan de forma correcta.

### AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad del Gobierno de España, Proyecto TIN2013-47276-C6-2-R. Este trabajo ha sido realizado en la Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

### REFERENCIAS

- [1] I. Shara, "La música es el lenguaje universal por excelencia," *Voces*, Enero, 2014.
- [2] J. Ribas-Lequerica, *Desarrollo de aplicaciones para Android*, Anaya, 2011.
- [3] R. Boulanger, *The Csound Book: Perspectives in Software Synthesis, Sound Design, Signal Processing and Programming*, The MIT Press Order, 1999.
- [4] E. Gómez, A. Klapuri, B. Meudic, "Melody description and extraction in the context of music content processing," *Journal of New Music Research*, vol.32, no. 1, pp. 23-40, 2003.
- [5] E. Molina, L.J. Tardón, A.M. Barbancho, I. Barbancho, "SiPTH: Singing Transcription Based on Hysteresis Defined on the Pitch-Time Curve," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* vol.23, no. 2, pp. 252 - 263, 2014.
- [6] A. De Cheveigné, H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol.111, no. 4, pp. 1917-1930, 2002.
- [7] L. Rabiner, R. Schafer, *Digital processing of speech signals*. Prentice-Hall Series in Signal Processing No. 7, Englewood Cliffs: Prentice Hall, 1978.
- [8] Android Developers, <http://developer.android.com>, last retrieved 02/02/2015, Official web site, 2015.
- [9] D. De Pedro, *Teoría completa de la Música vol. 1 y vol. 2 (Edición revisada)*, Real Musical-Carisch España, 2014.
- [10] J. Raso del Molino, *Compendio de Armonía Razonada, Volumen I, II y III*, Ediciones Si bemol, 1996.
- [11] The MatWorks, Inc., <http://es.mathworks.com>, last retrieved 07/04/2016, Official web site, 2016.
- [12] H. Kenmochi, H. Ohshita, "Vocaloid-commercial singing synthesizer based on sample concatenation," in *Interspeech 2007*, pp. 4009-4010, 2007