

# Especificación preliminar del proyecto

## Audio Fingerprint

**Nombre del grupo:** P3R.

**Integrantes:**

Daniel Aicardi

Edgardo Vaz

Melina Rabinovich

**Tutor:** Juan Cardelino

**Problema:** Identificar señales de voz y de audio a partir de su huella característica.

**Antecedentes:**

- *Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal.*

*Autores:* Kopparthi, Adapa, Barkana.

*Descripción:* Método que utiliza los cruces por cero y la energía de la señal para poder distinguir el umbral de decisión sonoro-sordo.

- *Construcción de la prótesis táctil "Pauper Tango".*

*Autores:* Rozo, Perry, Villegas.

*Descripción:* Extracción de frecuencia fundamental utilizando correlación.

- *YIN, a fundamental frequency estimator for speech and music.*

*Autores:* Alain de Cheveigne, Hideki Kawahara.

*Descripción:* Artículo original.

## Objetivos:

- Estudiar distintos algoritmos que permitan generar huellas descriptivas de señales de audio.
- Generar fingerprint de señales conocidas, que serán almacenadas en memoria.
- Definir un umbral que evite *falsos positivos* a la hora de reconocer la señal (mediante su fingerprint) que se está analizando, contra las que se tienen almacenadas en memoria.

## Alcance del proyecto:

Reconocer señales de audio mediante el uso de dos algoritmos para la generación de fingerprint. En una primera etapa solo para voz y en una segunda para audio en general.

### *Parámetros a definir:*

- Ancho de banda: Se elige a partir del máximo valor de frecuencia fundamental de la señal a tratar. Un posible valor se encuentra alrededor de  $\sim 4\text{KHz}$  para voz y  $\sim 22\text{KHz}$  para audio en general.
- Frecuencia de muestreo: La frecuencia de muestreo del codec del DSP56303 es configurable. Por esta razón se elegirá la frecuencia de muestreo de  $8\text{kHz}$  para señales de voz y de  $48\text{kHz}$  para señales de audio.
- Longitud de los tramos: Una posible elección es tomar intervalos de  $15\text{ms}$  para hallar la frecuencia fundamental  $f_0$ . Lo anterior permite dejar fuera la frecuencia  $50\text{Hz}$  de la red eléctrica, esto ocurre ya que la frecuencia fundamental mínima que puede hallarse en dicho intervalo es  $f_0 = 66,7\text{Hz}$  ( $1/15\text{ms}$ ).
- Solapamiento de tramos: Luego de calcular la frecuencia fundamental de un tramo, el próximo tramo es creado quitando las muestras de los primeros  $5\text{ms}$  del tramo anterior y agregando  $5\text{ms}$  de nuevas muestras.
- Función diferencia: El desplazamiento máximo de la señal está asociado con el período máximo ( $f_0 = 66,7\text{Hz}$ , mínimo) que puede encontrarse en un intervalo de tiempo. En nuestro caso, si elegimos  $T_{\text{max}} = 15\text{ms}$ , y  $f_s = 8\text{Khz}$  (señales de voz) --->  $T_{\text{max}} * f_s = 120$ , por lo tanto el desplazamiento máximo es de  $120$  muestras.  
Si  $f_s = 48\text{Khz}$  (señales de audio) --->  $T_{\text{max}} * f_s = 720$ , por lo tanto el desplazamiento máximo es de  $720$  muestras.

- Filtrado de la señal a la entrada: Se va a utilizar el filtro interno del codec que presenta las siguientes características: banda pasante  $0,45f_s$ , por lo que elimina todo lo que esté fuera del alcance audible ( $\sim 20\text{KHz}$ ), banda de transición  $0,45f_s$  a  $0,55f_s$ , banda suprimida mayor a  $0,55f_s$ , con  $74\text{dB}$  de caída.

#### *Algoritmo para generar huellas:*

- se almacenarán en un buffer las muestras obtenidas por el codec.
- se tomará un intervalo de muestras equivalente a  $15\text{ms}$ .
- a partir del intervalo anterior, calcularemos la frecuencia fundamental  $f_0$ , mediante dos métodos: *zero crossing rate* (cruces por cero), *estimador YIN* (función diferencia).
- luego de calcular la frecuencia fundamental de un tramo, el próximo tramo es creado quitando las muestras de los primeros  $5\text{ms}$  del tramo anterior y agregando  $5\text{ms}$  de nuevas muestras.
- se repetirá el paso anterior, hasta obtener la cantidad de valores de  $f_0$  equivalentes a  $3\text{ segundos}$  de la señal original ( $5\text{ms} \times q = 3\text{s}$ ), o sea  $q = 600$  muestras de  $f_0$ .
- el conjunto de  $f_0$  generadas, serán la huella de la señal de entrada.

Este algoritmo es aplicable tanto para la *señal incógnita* como para las *conocidas* que estarán almacenadas en memoria. La cantidad de muestras  $f_0$  que representan una huella es un parámetro, y puede no coincidir en ambos casos.

#### *Búsqueda de huellas en la base de datos:*

- a la *señal incógnita* de entrada, se le aplicará el algoritmo para generar huellas, se tomarán  $30$  muestras (este parámetro es una primera propuesta, pero será determinado mediante simulación) de  $f_0$  para esta señal.
- luego se calculará la autocorrelación entre dicha huella y las almacenadas en memoria del DSP.
- si el resultado de dicho cálculo supera cierto umbral (que será definido en forma empírica), se dará por hallada la huella en la base de datos.

### *Estimación de operaciones del DSP:*

- se tomará como hipótesis el hecho de que una multiplicación y una suma se realizarán en paralelo (un ciclo).
- se definen los siguientes parámetros:

$f_s$ : frecuencia de muestreo.

$t_{i15}$ : intervalo total de 15ms.

$t_{i5}$ : intervalo de 5ms.

$\#h$ : número de huellas almacenadas en memoria.

$N$ : cantidad de muestras de  $f_0$  (de señal incógnita).

$q$ : cantidad de muestras de  $f_0$  (de señal conocida, en memoria).

- para hallar una huella de la *señal incógnita*, se realizarán para el peor caso alrededor de:

$$op\_h = (t_{i15} + (N-1)*t_{i5}) * f_s \text{ operaciones.}$$

Con los parámetros estimados hasta el momento:

$$op\_h = (15ms + 29*5ms)*48kHz = 7680 \text{ operaciones}$$

correspondiente al audio.

- para buscar una huella en memoria, el peor caso se da cuando se debe recorrer toda la base de datos, de modo que se realizarán aproximadamente:

$$\#op = op\_h * q / N * \#h \text{ operaciones.}$$

Con los parámetros estimados hasta el momento:

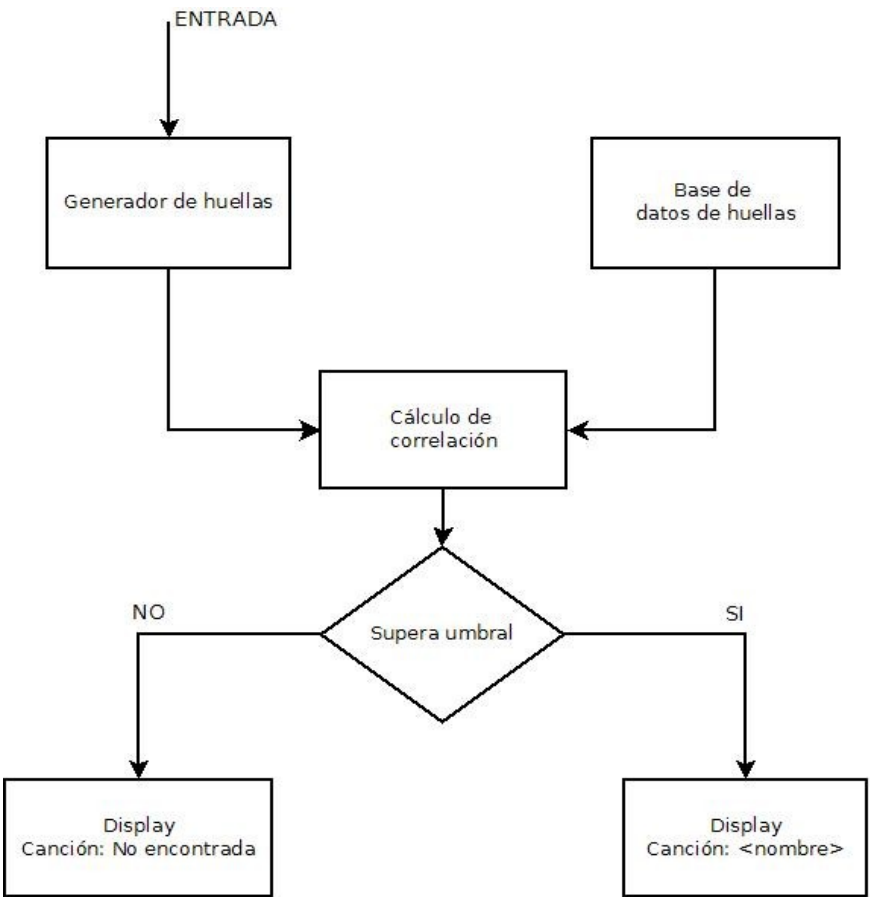
$$\#op = 7680 * (600/30) * 10 = 1536000 \text{ operaciones.}$$

### *Display:*

Mientras se efectúa la búsqueda en la base de datos, el display desplegará el mensaje: *buscando...*

Una vez reconocida la señal, se desplegará en el display el nombre de la canción a la cual pertenece esa huella. En caso de no reconocer la señal como una de las contenidas en la base de datos se desplegará: *canción no encontrada*.

**Diagrama de bloques (conceptual):**



## Planificación:

### *Cronograma:*

Nombre	fecha	Descripción
Entrega prop. definitiva	<b>08/10/09</b>	
Algoritmo ZCR	12/10/09	Implementación y pruebas en matlab/octave
Algoritmo YIN	18/10/09	Implementación y pruebas en matlab/octave
Depurado	22/10/09	Pruebas de huellas creadas con los algoritmos (previo a DSP)
1º Informe de avance	<b>02/11/09</b>	
Código C	09/11/09	Pasaje a código C para uso en el DSP
Pruebas en DSP	16/11/09	Busqueda de las huellas en el DSP
2º Informe de avance	<b>16/11/09</b>	
Correcciones	18/11/09	Depurado y corrección de errores
Redacción de info final	20/11/09	
Entrega de documentación	23/11/09	
Presentación	<b>30/11/09</b>	

### *Hitos:*

➤ 08/10/09:

Entrega propuesta definitiva.

➤ 02/11/09:

Presentación 1 de avance del proyecto.

➤ 16/11/09:

Presentación 2 de avance del proyecto.

➤ 30/11/09:

Presentación final.