UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CONSTRUCCIÓN DE UN MECANISMO DE PROCESAMIENTO DE PATRONES TEMPORALES APLICADO AL RECONOCIMIENTO DE VOZ

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

MANUEL ANÍBAL VALENZUELA RAMÍREZ

PROFESOR GUÍA:
JUAN D. VELÁSQUEZ SILVA

MIEMBROS DE LA COMISIÓN:
BENJAMÍN BUSTOS CÁRDENAS
JAIME SÁNCHEZ ILABACA

SANTIAGO DE CHILE
2015

RESUMEN DE LA MEMORIA PARA OPTAR AL
TÍTULO DE: Ingeniero Civil en Computación.
POR: Manuel Aníbal Valenzuela Ramírez
FECHA: 29/01/2015
PROFESOR GUÍA: Juan D. Velásquez Silva

CONSTRUCCIÓN DE UN MECANISMO DE PROCESAMIENTO DE PATRONES TEMPORALES APLICADO AL RECONOCIMIENTO DE VOZ

En el área de desarrollo de software para el control de aplicaciones y dispositivos electrónicos por voz, ha sido cada vez más común implementar mecanismos que cumplan esta función, considerando el procesamiento de señales sonoras para finalmente descubrir patrones que permitan la identificación y el uso de instrucciones.

El objetivo de este trabajo es la exploración de un mecanismo que implemente el procesamiento de la voz humana, extrayendo sus características fundamentales y utilizando estos datos para la identificación de patrones en el tiempo.

La hipótesis fundamental es que en la voz humana existen patrones en el tiempo, que podemos obtener y utilizar para la elaboración de instrucciones a ser ejecutadas por componentes de software.

Para lograr el objetivo se tomó como base la implementación de un mecanismo de obtención de espectros de frecuencias de la voz humana, considerando algoritmos y técnicas basadas en análisis espectral. Para el procesamiento de los patrones se desarrolló un mecanismo basado en redes neuronales, dada la naturaleza vectorial de los datos.

El trabajo, por tanto, se divide en dos grandes tareas. La primera es la obtención de los datos relevantes a la voz, de acuerdo con espectros de frecuencias obtenidos a partir de filtros basados en Wavelet transformadas. La segunda tarea es la implementación de una red neuronal no supervisada, basada en mapas auto-organizativos (SOM), que permita el registro e identificación de patrones en el tiempo.

El resultado de este trabajo es un mecanismo que cumple parcialmente sus objetivos, dados los niveles de identificación de los fonemas y el costo computacional requerido.

AGRADECIMIENTOS

Este trabajo no habría sido posible sin el apoyo de mi esposa Viviana y de mi hijo Agustín, gracias por su paciencia y por la fortaleza que me dieron.

Gracias a mi profesor guía y amigo, Juan Velásquez, que sin su poyo no hubiera sido posible terminar este trabajo, mi sueño por casi 20 años.

Quiero de la misma forma demostrar a mis hijos Lía y Gabriel, que en la vida siempre se debe luchar por los sueños, si no quedan en simples fantasías.

Finalmente quiero agradecer a mis padres, que me apoyaron en seguir este camino, sé que desde el cielo, mi viejo sonreirá feliz.

TABLA DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

1. INTRODUCCIÓN

En la actualidad, el uso de la voz humana para el control de software y el ingreso de datos en dispositivos electrónicos es una actividad común, gracias a que en los últimos años han existido grandes avances en las diferentes técnicas y herramientas para el desarrollo de esta tecnología (ver [W6] y [9]).

1.1 Motivación

En este contexto, es donde surge el interés en desarrollar un mecanismo de reconocimiento de patrones en relación al tiempo, aplicados a la voz humana. Con el objetivo de aportar a esta necesidad cada vez más demandada por instituciones tanto del ámbito público como privado.

Esta propuesta permite explorar alternativas de implementación, focalizando el trabajo en el uso de bancos de filtros para la obtención de espectros de frecuencia y el uso de un algoritmo de aprendizaje basado en redes neuronales para el registro de los datos y su posterior reconocimiento.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un mecanismo para la obtención e identificación de patrones temporales aplicados a la voz humana, considerando el uso de fonemas como la base de la identificación, y la exploración de un mecanismo de segmentación en el espacio espectral para la identificación de fonemas en palabras y frases.

1.2.2 Objetivos específicos

Los objetivos específicos de este trabajo son:

1. Implementar una técnica para la transformación de señales acústicas en espectros de frecuencias, basados en Wavelets.

2. Implementar un algoritmo que permita la segmentación y la clasificación de las señales sonoras en el espacio espectral.

3. Implementar un algoritmo para el aprendizaje y reconocimiento de patrones temporales asociados a la voz humana, basados en redes neuronales.

4. Obtener la verificación operacional y la verificación de la efectividad del mecanismo a partir del análisis de los resultados.

1.3 Metodología

La metodología considera los siguientes puntos:

1. **Revisar el estado del arte referente a las distintas técnicas para la generación de espectros de frecuencias asociados a la voz humana.**
   - Investigar las técnicas de análisis espectral utilizados en la actualidad.
   - Investigar el pre-procesamiento de señales acústicas .
   - Investigar la representación de patrones temporales aplicados a la voz humana y su relación con los fonemas en idioma español.
   - Estudiar los algoritmos y las librerías existentes en distintas plataformas de software, para la implementación de los programas requeridos.

2. **Revisar el estado del arte en el procesamiento de datos por medio de mecanismos basados en redes neuronales.**
   - Investigar los últimos avances en el uso de clustering y redes neuronales.
   - Estudiar los algoritmos y las librerías existentes en paquetes de software matemáticos, para la implementación de los programas requeridos.
   - Describir el estado del arte en el uso de técnicas basadas en redes neuronales.

3. **Implementar módulo con algoritmos de análisis espectral.**
   - Diseñar y desarrollar los programas y funciones requeridas.
   - Implementar los mecanismos y métricas para la evaluación de la calidad.
   - Evaluar el rendimiento de los algoritmos y de sus resultados.
   - Ajustar los parámetros operacionales y determinar sus rangos de validez.

4. **Implementar módulo para reconocimiento de patrones de voz.**
   - Desarrollar los programas y funciones requeridas.
   - Implementar los mecanismos y métricas para la evaluación de la calidad del reconocimiento de los patrones.

5. **Probar y validar el correcto funcionamiento del mecanismo desarrollado.**
   - Definir un plan de pruebas para la validación del mecanismo.
   - Ejecutar el plan de pruebas.

6. **Evaluación del resultado e identificación de mejoras posibles.**
   - Evaluar el mecanismo de obtención de espectros y procesamiento de datos acústicos, identificando posibles mejoras.
   - Evaluar el mecanismo de procesamiento de patrones, identificando posibles mejoras.

2. REQUERIMIENTOS

En este capítulo se presentan los **requerimientos funcionales** y **no funcionales** del mecanismo que se implementa en este Trabajo de Título.

2.1 Requerimientos funcionales

Se requiere implementar un mecanismo que permita completar el siguiente caso de uso:

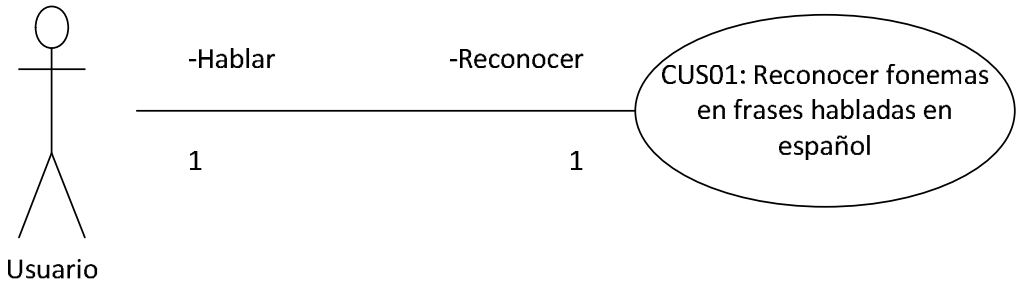

Ilustración 5: Caso de uso principal definido para el mecanismo.

CUS01: Reconocer fonemas en frases habladas en español
En este contexto, el requerimiento funcional principal considera que el usuario debe ser capaz de utilizar algún mecanismo que le permita reconocer fonemas en frases compuestas y presentar los resultados de una forma clara y medible.

El usuario debe ser capaz de presentar al mecanismo, los registros de voz en algún formato digital y de realizar los casos de uso que son descritos.

En este análisis se considera al usuario como aquella persona que requiere reconocer fonemas específicos en frases compuestas por medio de grabaciones de voz digital. El mecanismo debe ser capaz de completar este caso de uso de alto nivel.

Si se realiza un análisis más detallado, se pueden identificar dos casos de uso necesarios para completar el caso de uso principal:

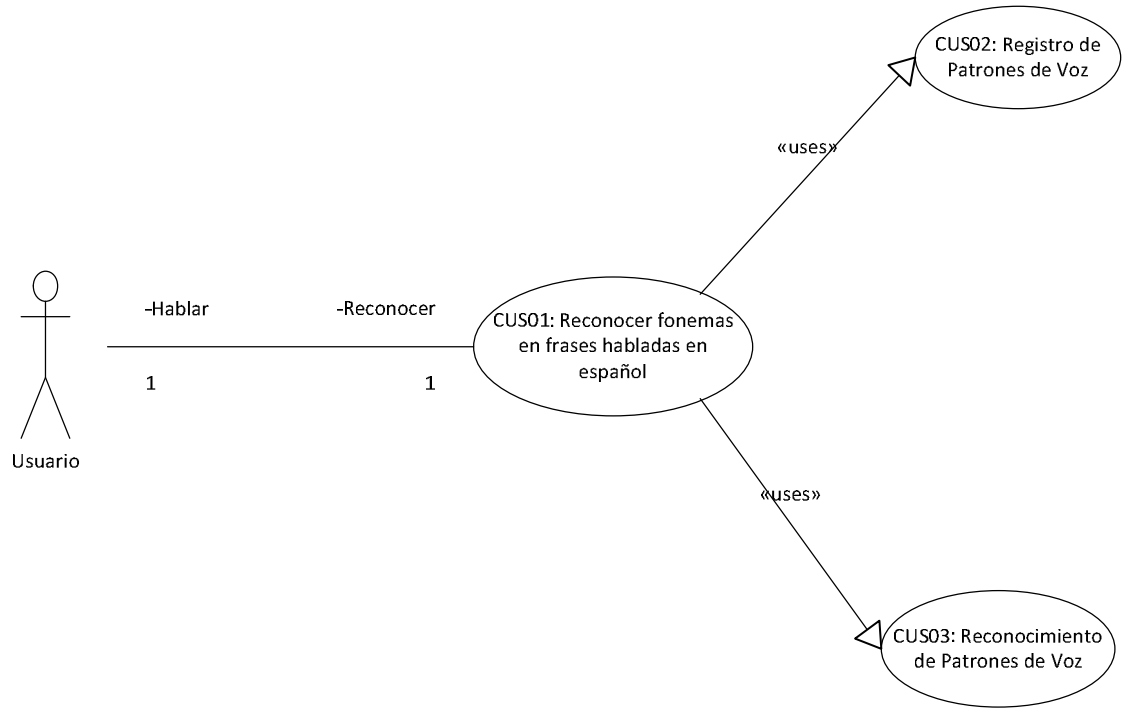

Ilustración 6: Caso de uso principal y casos de uso relacionados.

CUS02: Registro de Patrones de Voz

Este caso de uso describe la creación de un registro de fonemas para ser utilizado en el proceso de reconocimiento de voz. Cada fonema debe ser identificado, de tal manera que la respuesta del mecanismo pueda ser visible para el usuario al momento de proceder al reconocimiento.

El registro de patrones de voz y sus etiquetas deben ser implementados en el mecanismo de forma persistente, para que el usuario lo pueda aplicar a distintas frases compuestas en el futuro.

Este registro se debe hacer en función de grabaciones de voz en algún formato digital y considerando un procedimiento de asignación de etiquetas para su posterior identificación.

CUS03: Reconocimiento de Patrones de Voz

En este caso de uso se describe la actividad del reconocimiento. El usuario debe generar una frase compuesta y el mecanismo debe dar por respuesta los fonemas identificados.

El usuario debe ser capaz de crear frases complejas a raíz de un registro de voz grabado en algún formato digital en el que el mecanismo pueda ser aplicado para presentar los fonemas identificados.

El reconocimiento de voz en frases compuestas requiere de un conjunto de casos de usos adicionales que describan las actividades necesarias a ser implementadas en el

mecanismo, como lo son la transformación de señales sonoras y la segmentación de las frases compuestas para la identificación de los fonemas.

Al revisar en mayor detalle los requerimientos, es posible identificar nuevos casos de uso:

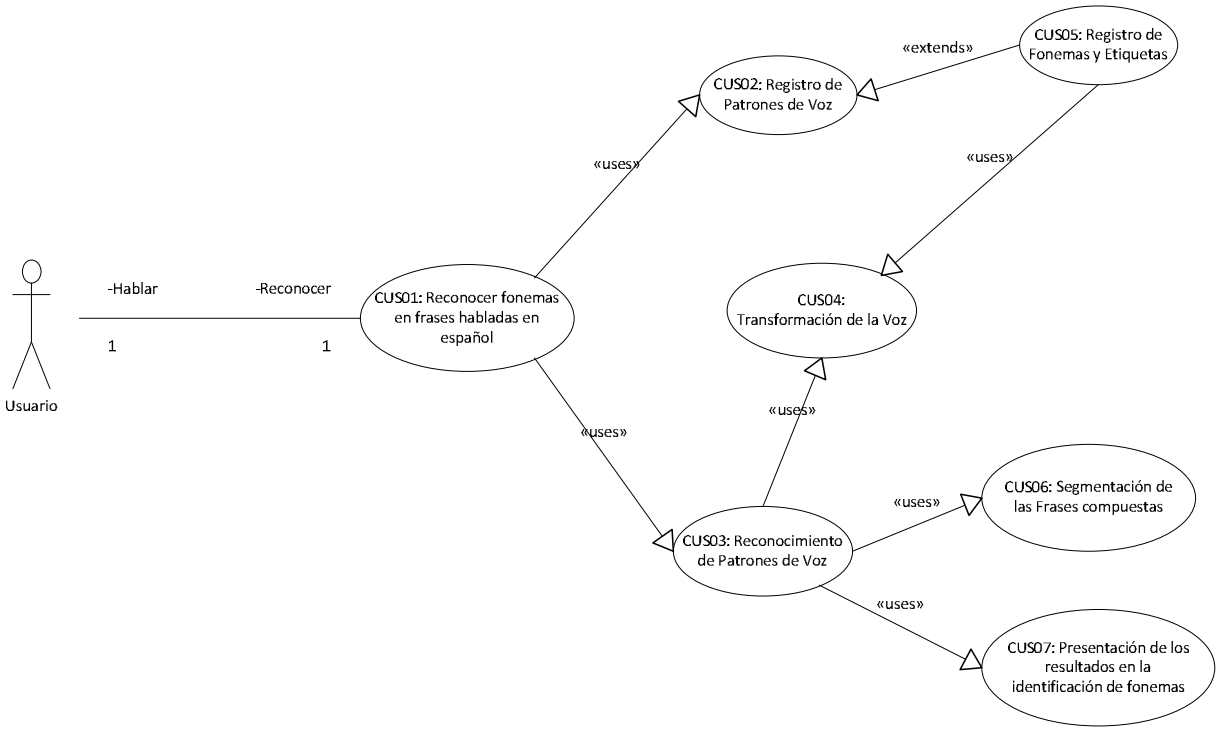

Ilustración 7: Casos de uso identificados para las funciones que el sistema debe realizar.

CUS04: Transformación de la Voz

Este caso de uso describe la necesidad de transformar las señales sonoras de voz en algún formato que permita su manipulación por parte del mecanismo implementado.

Esta transformación de la voz permite que los registros de audio puedan ser analizados por el mecanismo y así completar los objetivos del caso de uso principal.

Aunque el usuario del sistema no tendrá interacción directa con este caso de uso, es un requerimiento funcional del mecanismo.

CUS05: Registro de Fonemas y Etiquetas

Este caso de uso describe el registro por parte del usuario de un conjunto de fonemas (tokens) y el uso de un procedimiento basado en etiquetas para su posterior identificación.

El objetivo de este caso de uso consiste en identificar la función de los registros de voz de los fonemas específicos, creando un catastro o base de conocimiento con los fonemas de interés que serán estudiados por el usuario.

El usuario utilizará este catastro como base para el análisis de frases compuestas, lo que le permitirá identificar los fonemas.

Este caso de uso se requiere la transformación de la voz para completar su objetivo.

CUS06: Segmentación de las Frases Compuestas
Este caso de uso describe la segmentación necesaria de las palabras y frases compuestas, con el fin de identificar los fonemas que el usuario ha catalogado en el mecanismo.

La segmentación es un requerimiento funcional que el mecanismo deberá implementar al momento de ser presentada una frase compuesta, de tal manera que sea capaz de aislar las palabras de la frase y aplicarles las funciones de reconocimiento.

CUS07: Presentación de los resultados en la identificación de Fonemas
Este caso de uso describe la necesidad de elaborar un mecanismo que le permita al usuario verificar los fonemas identificados en las frases compuestas y validar el comportamiento del mecanismo.

La presentación de los resultados al usuario es un requerimiento funcional que debe considerar al menos los siguientes datos:

A. Fonemas reconocidos y su cantidad
Considera los fonemas reconocidos en la frase presentada y la cantidad de ocurrencias.

B. Fonemas no reconocidos y su cantidad
Considera los fonemas no reconocidos en la frase presentada y la cantidad de ocurrencias.

Los casos de uso descritos serán referenciados en los capítulos siguientes utilizando la sigla que lo identifica.

2.2 Requerimientos no funcionales

El mecanismo de reconocimiento de patrones temporales considera, por definición, los siguientes requerimientos no funcionales:

- El mecanismo a implementar utilizará registros de voz en formato digital, cuyo proceso de grabación estará fuera de las funciones integradas. Es decir, el mecanismo utilizará archivos de voz con los fonemas y las frases compuestas.
  Las muestras digitalizadas serán grabadas a una frecuencia de 8 kHz, para establecer un nivel de calidad similar al de la telefonía.

- El mecanismo a implementar deberá utilizar algún algoritmo para la obtención de espectros de frecuencia que permita establecer de patrones únicos basados en la voz humana.
- El mecanismo a implementar utilizará un procedimiento para el aprendizaje y el reconocimiento. Para ello, se ocupará un algoritmo basado en redes neuronales, dado que se requiere la identificación de patrones en un conjunto de datos basados en vectores (tramas), en el que la identificación es un proceso complejo.
- El mecanismo a implementar deberá considerar aspectos como bajo costo computacional (ver Sección 5.7) y uso de un paquete matemático que provea las librerías necesarias para las transformaciones y el procesamiento requerido.
- El mecanismo a implementar deberá ser paramétrico. Es decir, debe considerar un conjunto identificado de parámetros que gobiernen su comportamiento.
- El mecanismo a implementar estará compuesto por dos módulos. Un módulo para completar las funciones de catastro de fonemas y un módulo para el reconocimiento de los fonemas en frases compuestas.

Estas definiciones fueron establecidas en conjunto con el profesor guía, con el objetivo de limitar el alcance del mecanismo y para que los objetivos fueran adecuados en un Trabajo de Título de ingeniería.

3. MARCO CONCEPTUAL

Durante los últimos años han aparecido un conjunto de herramientas de software que permiten la captura, procesamiento y tratamiento de la voz humana, llegando ser una herramienta muy utilizada en múltiples dispositivos electrónicos con distintos fines.

El procesamiento de la voz requiere que el registro de voz capturado sea transformado en espectros de frecuencias capaces de ser organizados en paquetes etiquetados. De este modo se establece la base de un procedimiento de aprendizaje, primero, y de un mecanismo de reconocimiento, después.

3.1 Conceptos importantes

Análisis espectral: se refiere al procedimiento de transformación de señales de entrada en un conjunto de datos compuestos, que permiten el análisis detallado de cada una de sus partes. En este caso, las dimensiones relevantes son frecuencia y tiempo (ver [13] y [6]).

Fonética Acústica: la fonética tiene como objetivos determinar el modo en que los sonidos del habla se emplean con fines comunicativos en las lenguas naturales y explicar los mecanismos que condicionan tanto su producción como su percepción en el proceso del habla. En particular, la fonética acústica estudia la transmisión de un mensaje, en tanto que constituye una señal sonora (ver [21] y [26]).

Patrón: se refiere a la identificación de una estructura asociada a un concepto que pueda ser utilizada en procedimientos de reconocimiento como un elemento común en ese espacio (ver [W2], y [19]).

Fonema: se refiere al significado que se entrega a un sonido en el contexto del habla y que en combinación con otros fonemas es capaz de formar palabras y frases entendibles para los participantes. Todo ello en un idioma en concreto (ver [21] y [26]).

Pre-énfasis: se refiere al conjunto de transformaciones necesarias para procesar las señales de voz antes de llevar a cabo la generación de espectros de frecuencia. En general, estas transformaciones son utilizadas para dar énfasis a determinadas frecuencias (ver [27] y [19]).

Tramas: se refiere a un vector con información relevante. Este vector se obtiene como resultado de la aplicación de filtros Wavelet a los registros de voz originales en formato WAV, que son utilizados en el mecanismo de aprendizaje y reconocimiento.

Cuantización: se refiere al proceso de agrupar las señales sonoras transformadas en el espacio espectral, según las similitudes de sus características formantes (ver [W2], [23] y [16]).

Clustering: se refiere al procedimiento de agrupación de vectores, segúnen una definición de distancia y en relación a una implementación de red neuronal. (Ver [W1], [W4], [14] y [13]).

Redes Neuronales: se refiere a un mecanismo de procesamiento de la información que emula en forma muy básica el comportamiento de las neuronas. Considera mecanismos de aprendizaje y de reconocimiento (ver [10], [11], [12] y [6]).

Token: se refiere a un caracter o grupo de caracteres que serán utilizados en el mecanismo de aprendizaje para etiquetar el patrón que se está registrando y que posteriormente será utilizado en la presentación del resultado en el mecanismo de reconocimiento. Específicamente, se refiere a los fonemas, clasificados en fonemas vocálicos y fonemas consonánticos (ver sección 4.4).

Nodo: corresponde a las unidades mínimas de procesamiento en una red neuronal. Esta unidad está compuesta por los parámetros que permiten el aprendizaje. En la mayoría de los mecanismos, estos parámetros son denominados pesos. Una red neuronal está compuesta por múltiples nodos que trabajan en forma colectiva (ver [10]).

STFT: Short Time Fourier Transform es una variación de la transformada de Fourier, implementada para procesamiento computacional en el procesamiento de sonidos.

3.2 Estado del Arte – Detección de Patrones aplicados a la Voz

Los sistemas de reconocimiento de patrones temporales y su aplicación al análisis de la voz existen dentro de la rama de la computación desde hace algunas décadas (ver [W6] y [9]). Dichos sistemas han experimentado grandes cambios debido a las técnicas y herramientas utilizadas.

Uno de los principales problemas al procesar una señal de audio tiene que ver con los datos existentes dentro de dicha señal que no aportan información útil. Esto es, el ruido que, de una manera u otra, siempre está presente dentro del reconocimiento de voz. La eliminación total o parcial del ruido permite un mejor procesamiento y por lo tanto un mejor resultado al reconocer una señal de audio.

Existen varios métodos para la eliminación del ruido:

- Sustracción Espectral (ver [22] y [7]). Con éste método es posible aproximar una señal sin ruido a partir de la original con ruido, estimando el ruido que pueda existir de acuerdo con los silencios dentro de la señal.
- Filtro de Wiener (ver [W7]). Es uno de los métodos más conocidos debido a que opera según el cálculo del error cuadrático medio. Esté filtro tiene varias modificaciones, dependiendo de la aplicación específica que se le dé.
- Estimadores de Ephraim y Malah (ver [3]). Al igual que el filtro de Wiener, esta técnica utiliza el error cuadrático medio para la amplitud de la señal.

Estos tres métodos son los más simples y los que arrojan mejores resultados en aplicaciones.

Un segundo problema relevante a resolver es el proceso de segmentación de las señales en el espacio espectral. Esta segmentación permite identificar los paquetes de vectores que contienen información relevante con algún significado. En este caso particular, los fonemas. Existen consideraciones en [W2], [25] y en [19] que muestran alternativas para ser utilizadas en este trabajo.

El tercer tema relevante al analizar una señal tiene que ver con la cantidad de datos a procesar. Una señal puede ser muy grande y muchos de sus componentes pueden ser irrelevantes, por lo que se debe encontrar un medio para reducir la cantidad de datos a analizar, sin perder la información relevante del mensaje. Es necesario considerar distintos niveles de calidad al momento de la captura de los datos o utilizar técnicas basadas en compresión difusa (ver [14], [4] y [8]).

En el contexto de la extracción de características en una señal de audio, existen dos grupos de técnicas basadas principalmente en la transformada de Fourier y la transformada wavelet (ver [25], [22],[20] y [18]).

La transformada de Fourier constituye una herramienta mediante la cual podemos obtener información sobre cómo está distribuida la energía de una señal a través de sus distintas componentes de frecuencia. Su perfecta resolución en frecuencia la convierte en una herramienta muy útil para el análisis de señales estacionarias. Sin embargo, no puede ser aplicada con el objeto de obtener información precisa respecto de cuándo o dónde las diferentes componentes de frecuencia se encuentran en la señal, como es el caso de señales no estacionarias cuyo contenido espectral varía con el tiempo. La transformada de Fourier en señales no estacionarias posee una muy pobre resolución en cuanto al tiempo (ver [18]).

La transformada Wavelet constituye una técnica que ha sido propuesta como una poderosa herramienta en el análisis sobre el comportamiento local de una señal. Al igual que la transformada de Fourier, utiliza una función ventana que encuadra una señal dentro de un intervalo y focaliza el análisis sólo en ese segmento de la señal.
Dado que el objetivo de este trabajo es la implementación de un mecanismo para el análisis de patrones temporales basados en la voz, se requiere una herramienta que permita identificar características detalladas en el tiempo y en la frecuencia. El desarrollo y evolución de las técnicas basadas en Wavelets pueden ser revisadas en [25], [22] y [20].

Por último, en la implementación de los mecanismos de aprendizaje y reconocimiento de patrones se propone el uso de algoritmos basados en redes neuronales. Los avances en el uso de este tipo de procesamiento de información pueden ser revisados en [W1] y en [28].

Uno de los usos de las redes neuronales (ver [4]) está relacionado con el reconocimiento de hablantes. En este trabajo se utilizan también coeficientes de predicción lineal y una red neuronal basada en el algoritmo conocido como Backpropagation.

Lo expuesto anteriormente es solo una muestra de lo que se ha realizado dentro del campo del reconocimiento de voz. Existen más técnicas que pueden ser utilizadas para la reducción de ruido, la compresión de señales y el reconocimiento de patrones, dependiendo del problema a resolver.

Todos estos avances responden a que actualmente no existe un sistema de reconocimiento de voz cien por ciento efectivo, por lo que hay mucho por hacer en esta área.

3.3 Métricas de calidad y procedimiento de análisis comparativo

El objetivo de esta sección es describir las técnicas utilizadas para el análisis de resultados.

3.3.1 Conceptos de Precision/Recall y F-Measure

Precision/Recall
Las técnicas precision/recall son dos medidas ampliamente usadas en el área de Information Retrieval (IR) para evaluar el resultado de una búsqueda. Estas medidas se aplican sobre un conjunto de documentos y una búsqueda. Es necesario que la relevancia de los documentos sea conocida, suponiendo que un documento es completamente relevante o no lo es.

Con estas técnicas se puede saber qué tan completa ha sido la búsqueda (cuántos documentos relevantes ha retornado) y qué tan precisa (cuántos documentos irrelevantes ha retornado).

La definición de ambos conceptos es:

$$Recall \triangleq \frac{|\{documentos\ relevantes\} \cap \{documentos\ encontrados\}|}{\{documentos\ relevantes\}}$$

$$Precision \triangleq \frac{|\{documentos\ irrelevantes\} \cap \{documentos\ encontrados\}|}{\{documentos\ irrelevantes\}}$$

Según la teoría de IR, la definición es:

- Recall
  A: conjunto de documentos relevantes recuperados
  B: conjunto de documentos relevantes no recuperados
  recall = |A|/(|A| + |B|)
- Precision
  A: conjunto de documentos relevantes recuperados
  C: conjunto de documentos irrelevantes recuperados
  precision = |A|/(|A| + |C|) = |A|/(Todos los documentos recuperados)
  donde |X| es la cantidad de elementos.

Ambos indicadores, tanto precision como recall, están comprendidos entre los valores 0 y 1, indicando un valor porcentual de la medida.

Generalmente, se recomienda usar ambas medidas de manera paralela, pues revelan propiedades distintas. Existe una correlación negativa entre ellos: al ampliar los criterios de búsqueda, la precisión (precision) disminuye mientras que la completitud (recall) aumenta. En cambio, al ser más exigente (restrictivo) se pone en evidencia un comportamiento contrario. En un caso extremo, si una búsqueda devuelve solamente todos los documentos relevantes existentes, significa que tiene un valor de recall de 1 y el valor del indicador de precision es de 1.

F-Measure
Cuando se quiere utilizar una medida que considere los indicadores precision y recall, se usa la media armónica entre ambos indicadores, que se define de la siguiente forma:

$$F \triangleq \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

3.3.2 Aplicación de la técnica de evaluación

Los conceptos de precision, recall y F-measure hoy son ampliamente utilizados en los trabajos de Web Minning.

Al aplicarlos al mecanismo de reconocimiento de patrones temporales se consideran las siguientes definiciones:

- **Token:** unidad lingüística utilizada en el proceso de aprendizaje, que considera fonemas.

- **Frase:** conjunto controlado de palabras, que es presentado al mecanismo de reconocimiento.

- **Recall**
  A: tokens relevantes recuperados en una frase
  B: tokens relevantes no recuperados en una frase
  recall = |A|/(|A| + |B|)

- **Precision**
  A: tokens relevantes recuperados en una frase
  C: tokens irrelevantes recuperados en una frase
  precision = |A|/(|A| + |C|) = |A|/(Todos los tokens recuperados)

- **F-Measure**
  F-Measure = 2*(precision*recall)/(precision+recall)

La frase que se utilizará en el proceso de reconocimiento posee una colección de tokens claramente catalogados, por lo que puede ser evaluada.

Para el ejemplo, se consideran las siguientes definiciones:

Tokens = {a,e,i,o,u,ta,te,ti,to,tu,ma,me,mi,mo,mu,sa,se,si,so,su,…}

Frase = "Mi casa está totalmente sola"

A: cantidad de tokens recuperados en forma correcta

B: cantidad de tokens en la frase que no fueron identificados

C: cantidad de tokens identificados por el mecanismo, que no corresponden a los tokens existentes. Es decir, identificación errónea.

Un ejemplo que representa este comportamiento del mecanismo es:

Tokens totales en la frase = {mi,ca,sa,es,ta,to,tal,men,te,so,la}
A = {mi,ca,sa,to,te,so,la}, ||A|| = 7
B = {es,ta,tal,men}, ||B|| = 4
C = {mon,le}, ||C|| = 2

En este caso, los valores de los indicadores son los siguientes:
recall = 7/(7+4) = 0,63
precision = 7/(7+2) = 0,77
F-Measure = 2*(0,63*0,77)/(0,63+0,77) = 0,693

4. FUNDAMENTOS TEÓRICOS

El objetivo de este capítulo es presentar en términos generales los fundamentos teóricos, las técnicas y los algoritmos que serán evaluados para la implementación del mecanismo.

4.1 Ondas sonoras

El sonido se puede definir como la decodificación que efectúa el cerebro de las vibraciones percibidas a través de los órganos de la audición.

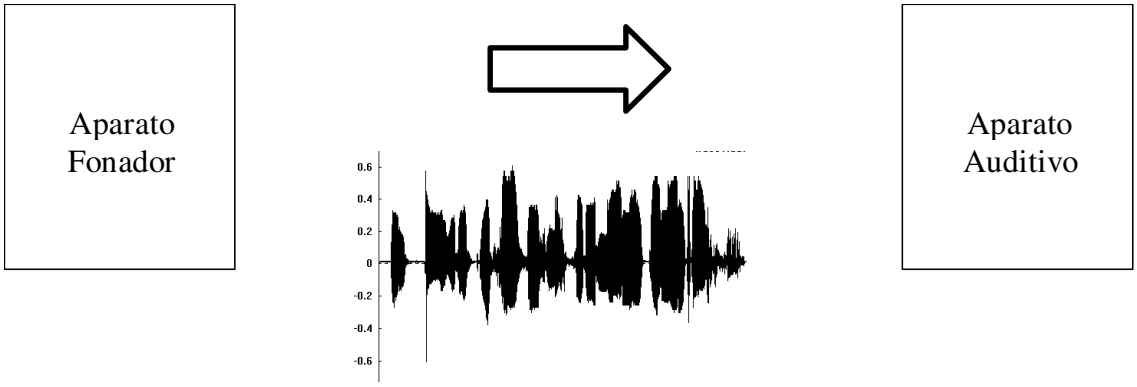

Ilustración 8: Relación entre el aparato fonador y el aparato auditivo.

El sonido se define utilizando un modelo basado en ondas que fluyen en un medio, en este caso el aire. Los órganos especializados son los encargados de producr y recibir estos sonidos para darles un significado de acuerdo con el contexto.Las ondas sonoras son interpretadas a través de movimientos oscilatorios cuasi periódicos que presentan las siguientes características:

- F : Frecuencia: oscilación por unidad de tiempo.
- P : Período: tiempo de una oscilación completa.
- λ : Longitud de Onda: distancia entre dos puntos fijos de una señal.
- A : Amplitud: máxima distancia entre la posición de reposo y la posición de oscilación.

$F = \frac{1}{P}$, relación entre frecuencia y periodo.

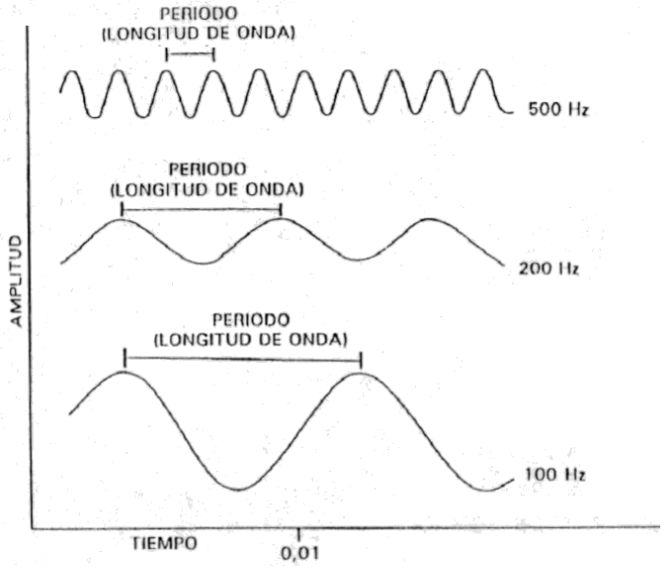

Ilustración 5: Relación entre frecuencia, longitud de onda y periodo, parámetros utilizados en el tratamiento de señales acústicas. Figura extraída de [6].

Toda onda que repite periódicamente su perfil se puede descomponer en un número limitado de sinusoides que tengan amplitud, frecuencia y fase diferentes. La impresión auditiva de la frecuencia fundamental se denomina tono.

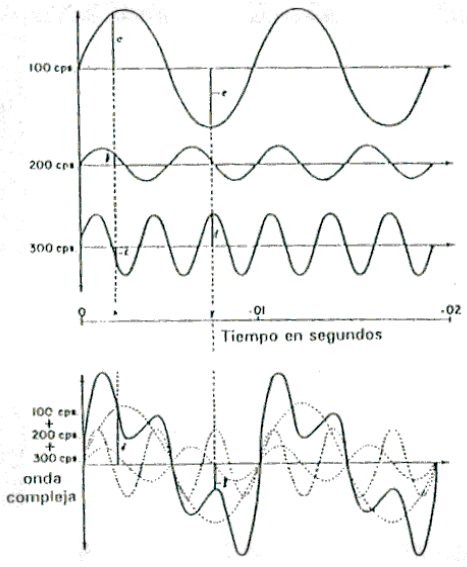

Ilustración 6: Ejemplo de composición de ondas sinusoidales que dan como resultado una onda compleja. Imagen obtenida de [6]

Las frecuencias pueden ser aisladas y representadas en espectrogramas que identifican las oscilaciones en el tiempo.

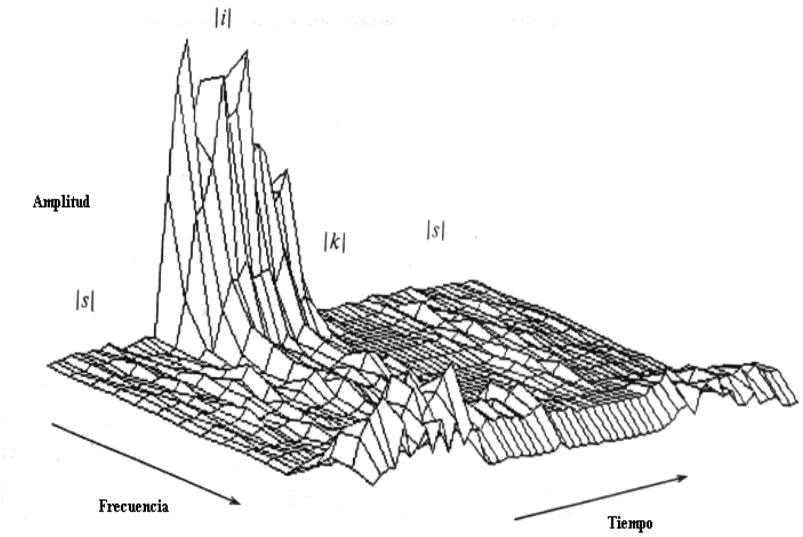

Ilustración 7: Espectrograma en el que los parámetros de amplitud, frecuencia y tiempo son representados con una gráfica 3D. Esta figura fue obtenida de [13] y traducida al español.

4.2 Análisis espectral en una dimensión

El análisis espectral es una técnica que permite transformar las señales sonoras en elementos discretos que pueden ser cuantificados y tratados numéricamente en función del tiempo. En este trabajo serán evaluadas las técnicas basadas en transformadas Wavelet.

4.2.1 Transformadas Wavelet

Esta técnica se desarrolló como una alternativa para superar los problemas de resolución de la transformada de Fourier, haciendo posible una buena representación de una señal en forma simultánea tanto en tiempo como en frecuencia. De este modo, se puede establecer el intervalo de tiempo en el cual aparecen determinadas componentes espectrales (ver [17]).

La transformada Wavelet filtra una señal en el dominio del tiempo mediante filtros paso bajo y paso alto que eliminan componentes de alta o baja frecuencia. Este procedimiento se vuelve a repitir para las señales resultantes del proceso de filtrado anterior.

Se puede ver claramente, si se toma como ejemplo una señal con frecuencias de hasta 1000 Hz. En la primera etapa de filtrado, la señal se divide en dos partes y se pasa a través de un filtro paso bajo y un filtro paso alto. Con esta operación, se obtienen dos versiones diferentes de la misma señal: una que corresponde a las frecuencias entre 0 y 500 Hz. (paso bajo) y otra que corresponde a las frecuencias entre 500-1000 Hz. (paso alto). Posteriormente, se toma cualquiera de las dos versiones (comúnmente la versión del filtro paso bajo) y se hace nuevamente la misma división. Esta operación se denomina descomposición.

De esta forma, y suponiendo que se ha tomado la versión de la señal correspondiente al filtro paso bajo, se tendrían tres conjuntos de datos, cada uno de los cuales corresponde a la misma señal pero a distintas frecuencias: 0-250 Hz., 250-500 Hz. y 500-1000 Hz. A continuación se vuelve a tomar la señal correspondiente a la parte del filtrado de paso bajo y se pasa por los filtros paso bajo y paso alto, de modo que se obtengan 4 conjuntos de señales correspondientes a las frecuencias 0-125 Hz., 125-250 Hz., 250-500 Hz. y 500-1000Hz. El proceso continúa hasta que la señal se ha descompuesto en un cierto número de niveles previamente definidos.

Finalmente se cuenta con un grupo de señales correspondientes a diferentes bandas de frecuencia. Para cada una de estas bandas se conocen sus respectivas señales, y si se juntan todas y se presentan en una gráfica tridimensional, se tendría tiempo en un eje, frecuencia en el segundo y amplitud en el tercer eje, lo que nos permite establecer qué frecuencias existen para un tiempo dado.

Sin embargo, el "principio de incertidumbre" de Heisenberg establece que no puede conocerse la información de tiempo y frecuencia de una señal en un cierto punto del plano tiempo-frecuencia. En otras palabras, no pueden determinarse exactamente qué frecuencias existen en un instante dado, por lo que sólo es posible conocer qué bandas de frecuencias existen en un determinado intervalo de tiempo.

Este problema de resolución es la razón principal por la cual es reemplazada la STFT por la WT, puesto que la STFT trabaja con una resolución fija para todos los tiempos, mientras que la WT trabaja con una resolución variable (ver [24]).

Con la WT, las altas frecuencias tienen mejor resolución en el tiempo mientras que las bajas frecuencias tienen mejor resolución en el dominio de la frecuencia. Esto significa que una determinada componente de alta frecuencia puede localizarse mejor en el tiempo (con menor error relativo) que una componente de baja frecuencia. Por el contrario, una componente de baja frecuencia puede localizarse mejor en frecuencia comparado con una componente de alta frecuencia.

4.2.2 La Transformada Wavelet Continua

La transformada Wavelet Continua (CWT) fue desarrollada como una técnica alternativa a la STFT para superar el problema de resolución.

El análisis wavelet se realiza de manera similar al análisis STFT. La señal es multiplicada por una función (función wavelet), de manera similar a la función ventana en la STFT, y la transformada se calcula separadamente para distintos segmentos de la señal, en el dominio del tiempo.

La transformada Wavelet Continua se define como:

$$C(\tau, s) = \int_{-\infty}^{+\infty} f(t)\psi^*_{\tau,s}(t)dt$$

Dónde:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{|s|}}\psi\left(\frac{t-\tau}{s}\right)$$

Como se observa en la ecuación anterior, la señal transformada es una función de dos variables, τ y s, que identifican los parámetros de traslación y escala respectivamente. $\psi_{\tau,s}(t)$, s(t) es la función de transformación que se denomina "wavelet madre". Este nombre deriva de dos importantes propiedades del análisis wavelet:

- El término wavelet significa "onda pequeña". "Pequeña" indica el hecho de que esta función (ventana) es de longitud finita (compactamente soportada) y "onda", la condición de que esta función es de naturaleza oscilatoria.
- El término **madre** da a entender que las funciones con diferentes regiones de actuación que se usan en el proceso de transformación provienen de una función principal o **wavelet madre**. Es decir, la wavelet madre es un prototipo para generar las otras funciones ventanas.

La transformación, finalmente aplicada, corresponde a la combinación de las dos primeras, traslación y cambio de escala.

Traslación	Cambio de escala	Traslación y cambio de escala
$\psi(t-b)$	$\frac{1}{\sqrt{a}}\psi\left(\frac{t}{a}\right)$	$\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$

Algunas wavelets madres clásicas son:

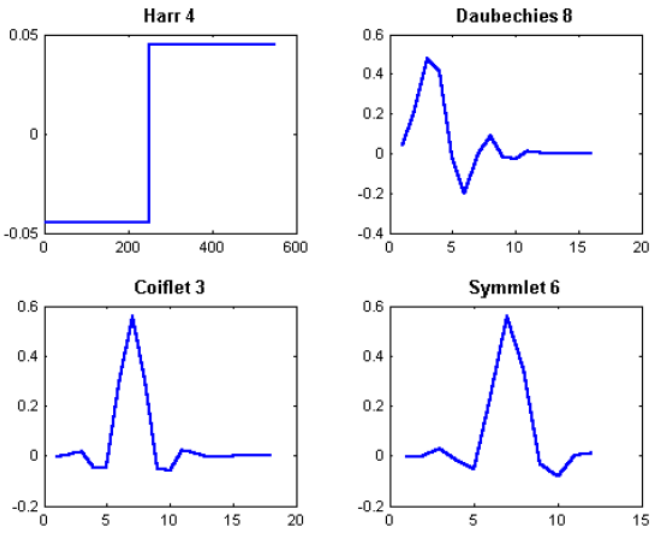

Ilustración 8: Cuatro funciones madre para el uso de transformadas Wavelet. Esta imagen fue obtenida en [2]

Traslación
El término **traslación** se usa con el mismo sentido que es usado en la STFT y está relacionado con la localización de la ventana a medida que ésta se desplaza a través de la señal. Este término, por tanto, identifica la información del tiempo en el dominio transformado. Sin embargo, no se tiene un parámetro que indique la frecuencia, como si se tenía antes en el caso de la STFT. Por ello, la transformada wavelet tiene un parámetro de "escala" que se define como:

$$\text{Escala} = \frac{1}{\text{frecuencia}}$$

Escala
En el análisis wavelet, el parámetro escala es análogo al parámetro escala utilizado en los mapas. Tal como en este último caso, las altas escalas corresponden a una visión global no detallada (de la señal) y las bajas escalas, a una vista detallada.

De igual manera, en términos de frecuencia, las bajas frecuencias (altas escalas) corresponden a una información global de la señal que comúnmente abarca toda la señal, mientras que las altas frecuencias (escalas bajas) corresponden a una información detallada de una característica oculta en la señal, que comúnmente dura un tiempo relativamente pequeño.

Conjunción traslación y escala
En señales que corresponden a fenómenos o aplicaciones reales, las escalas bajas (altas frecuencias) no tienen una larga duración en la señal, sino que aparecen de tiempo en tiempo como picos o "spikes". En cambio, las altas escalas (bajas frecuencias) comúnmente aparecen todo el tiempo de duración de la señal.

El escalamiento como operación matemática produce una dilatación o una compresión de una señal. Las altas escalas corresponderán a señales dilatadas y las escalas pequeñas a señales comprimidas. Todas las señales mostradas en la figura nacen de la misma señal, es decir son versiones comprimidas o dilatadas de la misma función.

En términos de funciones matemáticas, f(t) es una función dada y f(st) corresponderá a una versión contraída (comprimida) de f(t), si $s > 1$ y a una versión expandida (dilatada) de f(t), si $s < 1$.

Sin embargo, en la definición de la transformada Wavelet, el término de escalamiento aparece en el denominador y, por lo tanto, la situación es opuesta a la descrita en el párrafo anterior. Es decir, escalas $s > 1$ dilatan la señal, mientras que escalas $s < 1$ comprimen la señal.

La relación entre la escala y la frecuencia establece que las escalas menores corresponden a altas frecuencias y las escalas mayores corresponden a bajas frecuencias.

Debido a que la WT incluye información relacionada con el tiempo y la frecuencia, la representación gráfica de esta transformada se realiza en un plano denominado plano tiempo-escala, tal y como se representa en la siguiente figura:

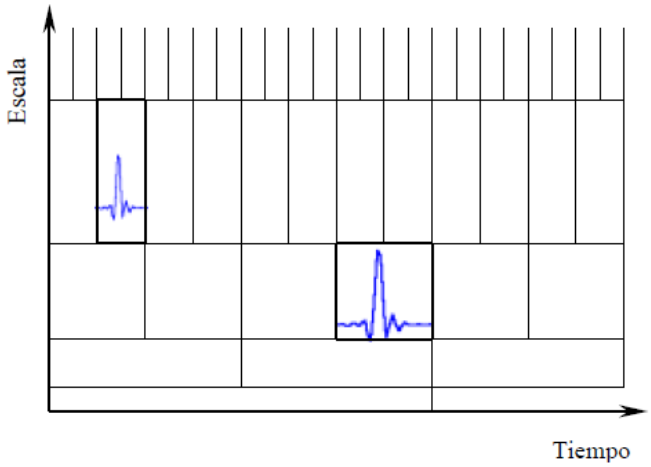

Ilustración 9: Espectrograma basado en Wavelets.

Independientemente de las dimensiones de cada división, las superficies de éstas, tanto para el caso de la STFT como para el caso de la WT, son iguales y están determinadas por el principio de incertidumbre de Heisenberg. Es decir, el área de cada división es fija para cada función ventana (STFT) o para cada wavelet madre (CWT), aun cuando diferentes ventanas o wavelet madres pueden representar diferentes áreas.

Las dos operaciones básicas de escalado y traslación definen el enrejado del plano tiempo-escala. En caso de tener buena resolución temporal, la wavelet madre, representada en el eje inferior, se estrecha, con lo que se pierde resolución en la frecuencia. Si la wavelet madre se ensancha, se pierde resolución en el tiempo y se gana en la frecuencia. Así, variando la anchura y desplazándola por el eje temporal, se calcula el valor correspondiente a cada celda.

4.2.3 Tipos de Transformadas Wavelet

Existen tres tipos de transformada Wavelet*:* Continua (CWT), Semi-discreta (SWT) y Discreta (DWT). La diferencia entre ellas radica principalmente en la forma en que los parámetros de desplazamiento y escala son discretizados.

Transformada Wavelet Discreta.
Sea la señal a analizar f[n] una función discreta. En este caso la transformada Wavelet de esta señal viene dada por:

$$C[j,k]=\sum_{n\in Z} f[n]\psi_{j,k}[n]$$

Donde $\psi_{j,k}$ es una Wavelet Discreta definida como:

$$\psi_{j,k}[n] = 2^{-\frac{j}{2}} \cdot \psi\left[2^{-j}n - k\right]$$

La transformada inversa se define de forma similar:

$$f[n] = \sum_{j \in Z} \sum_{k \in Z} C[j,k] \cdot \psi_{j,k}[n]$$

Para efectos computacionales, la implementación de DWT se basa en el uso de análisis de multi-resolución, utilizando baterías de filtros (ver [24] y [29]).

4.2.4 El análisis multi-resolución

Aun cuando la Transformada Wavelet Continua puede evaluarse computacionalmente de manera discreta, esto no constituye realmente una transformada discreta, sino una serie wavelet o la versión muestreada de la CWT, con la desventaja de que la información que entregan es altamente redundante para la reconstrucción de la señal. Esta redundancia significa, además, un aumento considerable del tiempo de cálculo. Por este motivo, se utiliza la Transformada Wavelet Discreta (DWT) que es capaz de entregar suficiente información tanto para el análisis como para la reconstrucción de una señal, y con una significativa reducción del tiempo de procesamiento.

Para ser útil, la teoría de wavelets debe disponer de algoritmos rápidos para su uso en computadores, pues permite encontrar los coeficientes Wavelet C[j,k] y reconstruir la función que representan.

Existe una familia rápida de algoritmos basados en el análisis multi-resolución o MRA. El análisis multi-resolución, o algoritmo piramidal, se desarrolló para descomponer señales de tiempo discreto. La idea es la misma que en la CWT, obtener una representación tiempo-escala de una señal discreta. En este caso, filtros con distintas frecuencias de corte son usados para analizar la señal en diferentes escalas. La señal se pasa a través de filtros paso alto para analizar las componentes de alta frecuencia, y se pasa a través de filtros paso bajo para analizar las componentes de baja frecuencia. Estas operaciones cambian la resolución de la señal, y la escala se cambia mediante operaciones de interpolación y sub-muestreo (ver [2]).

El análisis multi-resolución de *Mallat* (ver [17]) se relaciona con este algoritmo piramidal. En este caso, se incluyen filtros de espejo en cuadratura (denominados QMF). Por tanto, la representación tiempo-escala de una señal digital se obtiene mediante técnicas de filtrado digital. El proceso de descomposición comienza pasando la secuencia discreta correspondiente a la señal a través de un filtro paso bajo de

media banda con respuesta al impulso h[n]. El filtrado de la señal corresponde a la operación matemática de convolución de ésta con h[n].

4.2.5 Codificación de sub-bandas

La idea básica consiste en obtener una representación tiempo-escala de una señal mediante técnicas de filtrado digital. En el caso discreto, se utilizan filtros con diferentes frecuencias de corte para analizar la señal en las diferentes escalas; de este modo la señal se pasa a través de una serie de filtros paso alto para analizar las altas frecuencias y de filtros paso bajo para analizar las bajas frecuencias.

La resolución varía por la operación de filtrado, mientras que la escala varía mediante operaciones de sub-muestreo (interpolar, sub-muestrear), que reducen la tasa de muestreo o eliminan algunas muestras de la señal. Por ejemplo, sub-muestrear por dos significa tomar una de cada dos muestras de la señal. El sub-muestreo por un factor "n" reduce el número de muestras de la señal "n" veces. Interpolar una señal, en cambio, significa incrementar la tasa de muestreo agregando nuevas muestras a la señal. Por ejemplo, interpolar por "2" significa agregar una nueva muestra, usualmente un cero o un valor interpolado entre dos muestras de la señal. Por lo tanto, interpolar una señal por un factor "n" aumenta el número de muestras en la señal por un factor "n".

Aun cuando no es la única elección posible, los coeficientes de la DWT comúnmente se calculan mediante una escala diádica, es decir, $s_0 = 2$ y $\tau = 1$, de manera que $s = 2^j$ y $\tau = k2^j$. Como la señal ahora es una función discreta en el tiempo, los términos función y secuencia se usarán indistintamente en este análisis y la señal se denotará como x[n], donde "n" es un número entero.

El procedimiento para obtener la DWT comienza al pasar la señal (secuencia) a través de un filtro digital de paso bajo y media banda con respuesta al impulso h[n]. Este proceso de filtrado consiste en realizar matemáticamente la convolución de la secuencia con la respuesta al impulso del filtro, y se define como:

$$x[n] * h[n] = \sum_{-\infty}^{\infty} x[k] \cdot h[n-k]$$

Un filtro paso bajo de media banda elimina todas las frecuencias que están por encima de la mitad de la mayor frecuencia de la señal. En el caso que se tomó como ejemplo, si la señal tiene como máximo una componente de 1000 Hz., este filtro eliminaría todas las frecuencias sobre los 500 Hz.

En señales discretas, la frecuencia se expresa en radianes, por lo que la frecuencia de muestreo de la señal es igual a 2π en términos de la frecuencia radial. Esto supone que la componente de mayor frecuencia que existe en la señal será de π radianes, si el muestreo se realiza a la frecuencia de Nyquist, que corresponde al doble de la máxima

frecuencia que existe en la señal. La frecuencia de Nyquist, por tanto, corresponderá a π rad/s en el dominio discreto de la frecuencia.

Una vez que la señal ha pasado por el filtro paso bajo de media banda, la mitad de las muestras se pueden eliminar de acuerdo con la regla de Nyquist, ya que la señal ahora tiene la mayor frecuencia en $\pi/2$ radianes en vez de π radianes. Se elimina, entonces, una de cada dos muestras de la señal (sub-muestreo por 2), lo que provoca que se reduzca el número de puntos a la mitad y se duplique la escala de la señal.

El filtrado paso bajo elimina la información de alta frecuencia, pero deja la escala invariable, puesto que solamente el proceso de sub-muestreo la altera. Y como la resolución está vinculada con la cantidad de información en la señal, ésta se ve alterada por las operaciones de filtrado que eliminan la mitad de las frecuencias, lo que, además, podría interpretarse como la pérdida de la mitad de la información.

Por lo tanto, se puede concluir que la resolución se reduce a la mitad después de la operación de filtrado. Sin embargo, el proceso de sub-muestreo posterior al filtrado no afecta a la resolución, ya que al eliminar la mitad de las componentes espectrales, la mitad del número de muestras se hacen redundantes también, por lo que la mitad de las muestras pueden eliminarse sin ninguna pérdida de información.

En resumen, el filtrado paso bajo reduce a la mitad la resolución, pero no altera la escala.

El procedimiento anterior puede expresarse matemáticamente como:

$$y[n] = \sum_{-\infty}^{\infty} h[k] \cdot x[2n - k]$$

La DWT analiza la señal descomponiéndola en una aproximación y en un detalle (nivel), considerando diferentes bandas de frecuencias con distintas resoluciones para cada nivel. Para ello, se emplean dos conjuntos de funciones denominadas funciones de escalamiento y funciones wavelets, que están asociadas a filtros paso bajo y paso alto, respectivamente.

La descomposición de la señal en diferentes bandas de frecuencia se obtiene mediante un sucesivo filtrado de paso bajo y paso alto. La señal original x[n], por tanto, se pasa a través de un filtro paso alto de media banda g[n] y de un filtro paso bajo h[n]. Y después de este filtrado pueden eliminarse la mitad de las muestras de acuerdo a la regla de Nyquist, ya que la señal ahora tiene una frecuencia superior de $\pi/2$ radianes en vez de π. Para ello se eliminan una de cada dos muestras (sub-muestreo por 2).

De esta manera se ha constituido el primer nivel de descomposición, lo que atemáticamente puede expresarse como:

$$y_{high}[k] = \sum_{n} x[n] \cdot g[2k - n]$$

$$y_{low}[k] = \sum_{n} x[n] \cdot h[2k - n]$$

Donde $y_{high}[k]$ e $y_{low}[k]$ son las salidas de los filtros paso alto y paso bajo, respectivamente, después del sub-muestreo por 2.

Esta descomposición reduce a la mitad la resolución en el tiempo, como consecuencia de la reducción a la mitad del número de muestras originales que caracterizan a la señal. Sin embargo, esta misma operación duplica la resolución en frecuencia, ya que ahora la banda de frecuencia de la señal abarca solamente la mitad de la banda de frecuencias anteriores.

Este procedimiento se denomina codificación de sub-bandas y puede repetirse para conseguir una mayor descomposición. En cada ocasión, el filtrado y el sub-muestreo darán como resultado una disminución a la mitad del número de muestras (resolución en el tiempo dividida) y de la banda de frecuencias abarcada (resolución en la frecuencia duplicada).

En la figura se muestra un ejemplo de este procedimiento, donde x[n] es la señal original que se va a descomponer y h[n] y g[n] son los filtros paso bajo y paso alto, respectivamente. En cada nivel de descomposición, el ancho de banda de la señal aparece señalado en la figura como "f".

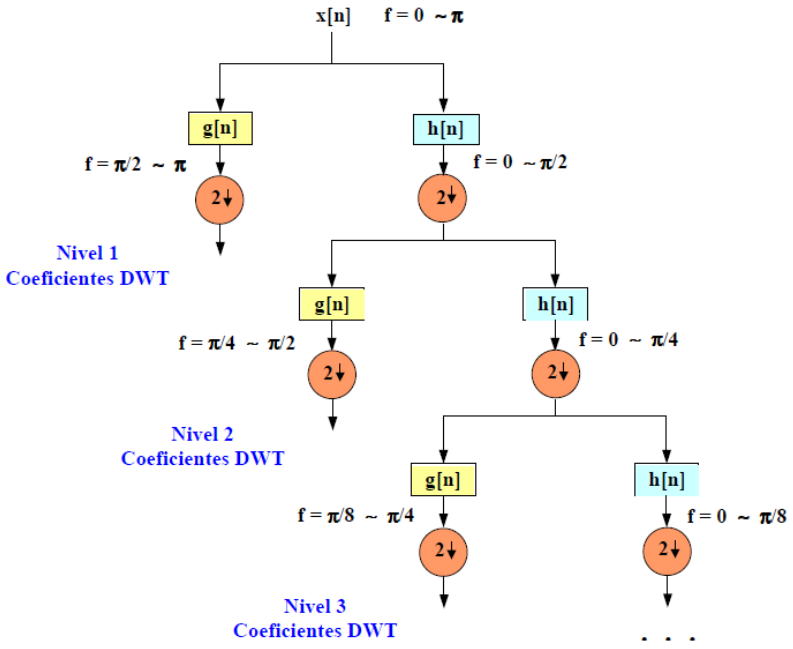

Ilustración 10: Descomposición en sub bandas con el algoritmo aplicado de forma recursiva.

En el ejemplo anterior se parte del supuesto de que se analiza una señal que tiene 512 muestras y una frecuencia en el rango de [0, π] rad/s. En el primer nivel de descomposición, la señal x[n] se pasa a través de los filtros paso alto g[n] y paso bajo h[n], continuando con un sub-muestreo por dos. La salida del filtro paso alto tendrá 256 muestras, lo que conlleva que la resolución en el tiempo se divide a la mitad y la frecuencia abarca la banda entre [π/2, π] rad/seg. Es decir, la resolución en frecuencia se ha duplicado. Estas 256 muestras constituyen el primer nivel de los coeficientes de la DWT.

Por su parte, la salida del filtro-paso bajo también tendrá 256 muestras, pero con una frecuencia que abarca el rango entre [0, π/2] rad/s.

Esta señal de salida se sigue descomponiendo pasándola nuevamente por filtros paso alto y paso bajo, así la salida del segundo filtro paso bajo seguida del sub-muestreo por dos tendrá ahora 128 muestras que abarcan un rango de frecuencias entre [0, π/4] y la salida del segundo filtro paso alto tendrá también 128 muestras, pero con una banda de frecuencias en el rango entre [π/4, π/2]. La segunda señal pasada por el filtro paso alto constituye el segundo nivel de los coeficientes de la DWT. Esta señal tiene la mitad de resolución en el tiempo, pero el doble de la resolución en frecuencia que la señal del primer nivel. En otras palabras, la resolución en el tiempo ha disminuido por un factor de cuatro, mientras que la resolución en frecuencia se ha incrementado por cuatro en comparación con la señal original.

El proceso continúa hasta que queden solamente dos muestras, haciendo que las salidas de los filtros paso bajo sean nuevamente filtradas para una mayor descomposición. Para este ejemplo en particular, podrían existir hasta 8 niveles de descomposición, cada uno con la mitad de muestras que el anterior. La DWT de la señal original se obtiene concatenando todos los coeficientes, comenzando desde el último nivel de descomposición. Así, la DWT tendrá el mismo número de coeficientes que la señal original.

Las frecuencias que son más dominantes en la señal original aparecerán como altas amplitudes en la región de la DWT que incluye esas frecuencias. La diferencia entre la FT y la DWT es que con la DWT no se pierde la localización en el tiempo de estas frecuencias. Sin embargo, la localización en el tiempo tendrá una resolución que dependerá del nivel en que aparezca. De este modo, si la información principal contenida en la señal está en altas frecuencias, como sucede a menudo, entonces la localización en el tiempo de estas frecuencias será más precisa, puesto que estarán caracterizadas por un mayor número de muestras.

En cambio, si la información principal está a muy bajas frecuencias, entonces su localización en el tiempo no podrá ser muy precisa, dado que existirán muy pocas muestras para caracterizar la señal a estas frecuencias.

En resumen, el procedimiento descrito ofrece una buena resolución en el tiempo para las altas frecuencias y una buena resolución en frecuencia para las bajas frecuencias.
Las bandas de frecuencia que no son muy dominantes en la señal x[n] darán origen a coeficientes de la DWT muy pequeños, que pueden despreciarse sin mayor pérdida de información, pero con una importante reducción de los datos.

Una propiedad importante de la DWT es la relación entre las respuestas impulso de los filtros paso alto y paso bajo. Estos filtros no son independientes entre sí, sino que están relacionados a través de la siguiente ecuación:

$$g[L-1-n] = (-1)^n \cdot h[n]$$

Donde g[n] es el filtro paso alto, h[n] es el filtro paso bajo y L es la longitud del filtro expresada en número de puntos.

La conversión de paso bajo a paso alto se hace a través del factor (-1)n. Los filtros que satisfacen esta característica se conocen como Filtros Espejos en Cuadratura (QMF).

Los dos filtrados y la operación de sub-muestreo pueden expresarse como:

$$(Gf)_k = y_{high}[k] = \sum_n x[n] \cdot g[-n + 2k]$$

$$(Hf)_k = y_{low}[k] = \sum_n x[n] \cdot h[-n + 2k]$$

La forma más compacta de describir este proceso, así como de representar los procesos para determinar los coeficientes Wavelet, es la representación de los filtros en forma de operador G y H. Estas ecuaciones representan el filtrado de la señal mediante los filtros digitales h[n], g[n]. El factor *2k* representa el sub-muestreo. Los operadores H y G corresponden a un paso en la descomposición Wavelet.

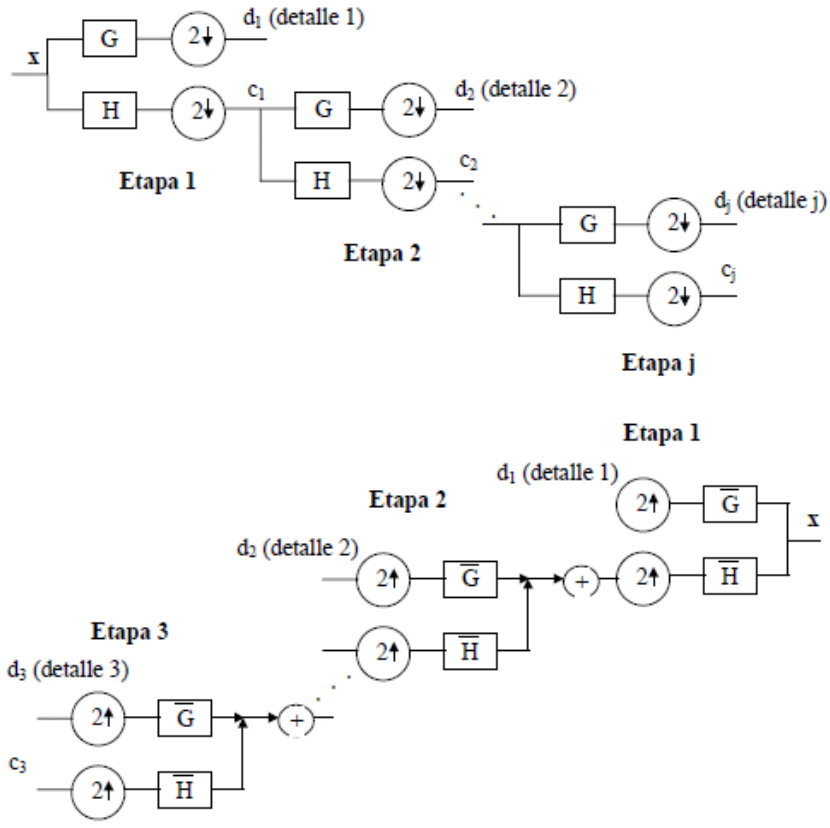

Ilustración 11: Proceso de descomposición y composición de las señales de entrada, al momento de aplicar los filtros de codificación de sub bandas.

Algoritmo piramidal o codificación sub-banda (banco de filtros en octava con *J* etapas).

La parte superior corresponde al análisis y la inferior a la síntesis. H es el filtro paso bajo y G el filtro paso alto.

La DWT anterior puede ser resumida en una única línea como:

$$x \to \left(Gx, GHx, GH^2x, \ldots, GH^{j-1}x, H^jx\right) = \left(d^{(j-1)}, d^{(j-2)}, \ldots, d^{(1)}, d^{(0)}, c^{(0)}\right)$$

Donde d(j-1),d(j-2),...,d(1),d(0) se denominan coeficientes del detalle y c(0) coeficiente de la aproximación. Los detalles y aproximaciones se obtienen de forma iterativa como:

$$c^{(j-1)} = Hc^{(j)}$$
$$d^{(j-1)} = Gd^{(j)}$$

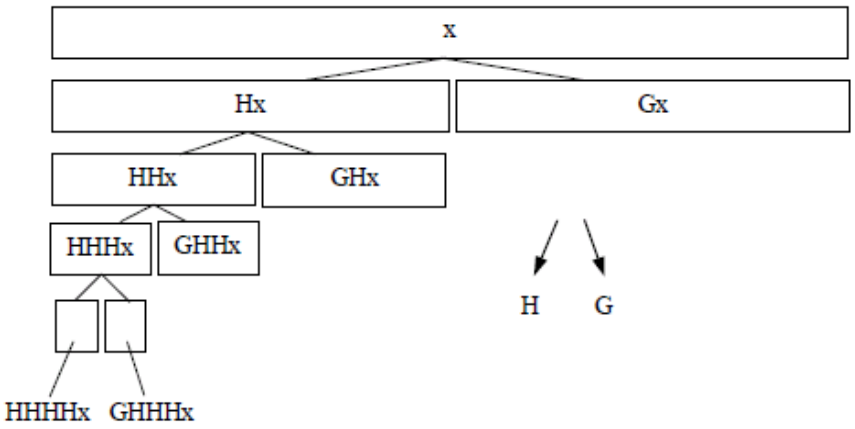

Ilustración 12: Segmentación de los vectores de entrada y su procesamiento, según el nivel de profundidad definido.

La reconstrucción en este caso es muy simple, dado que los filtros de banda media forman una base ortonormal, por lo que el procedimiento anteriormente descrito se sigue en sentido inverso. La señal es interpolada por 2 y pasada a través de los filtros de síntesis g'[n] y h'[n], paso alto y paso bajo respectivamente. Después, ambas salidas se suman.

Los filtros de análisis y síntesis son idénticos. Sin embargo, si los filtros no son de banda media ideal, la reconstrucción perfecta de la señal no puede conseguirse. Los más famosos son los que desarrolló Ingrid Daubechies, conocidos como las wavelets de Daubechies (ver [5]).

4.3 Modelos de redes neuronales y mapas auto-organizativos

En este punto se revisan los mecanismos de procesamiento paralelo distribuido basados en redes de neuronas y se especifican los algoritmos utilizados en este Trabajo de Titulación.

4.3.1 Conceptos y clasificación de las Redes Neuronales

La aparición de las redes neuronales data de la década de los 40, cuando se dieron los primeros intentos por tratar de simular el funcionamiento del cerebro humano. Este esfuerzo permitió el desarrollo de dos ramas de estudio como la Inteligencia Artificial y las Redes Neuronales. Más precisamente se define una Red Neuronal como un gran conjunto de elementos individuales capaces de realizar una tarea muy simple y específica, y que, al ser organizados, pueden realizar el reconocimiento de objetos. Actualmente, también se considera a los sistemas basados en redes neuronales como procesadores paralelos distribuidos.

Las unidades individuales son conocidas como Neuronas y su funcionamiento se inspira en el estudio biológico del proceso del conocimiento. Por eso, su representación es análoga a una célula nerviosa que capta un conjunto de entradas y adquiere o no un estado de excitación, que determina el envío de una salida. Sin embargo, existen un gran número de variables que no han podido ser incluidas en los modelos.

Entre los modelos que actualmente están en uso, se destacan los de la familia basada en Retropropagación (como los algoritmos derivados del Counterpropagation), las redes de Hopfield y, más específicamente, las redes basadas en auto-organización. Estos modelos sobrepasaron en capacidad de operación al Perceptron (descripción específica, demostraciones y detalles de implementación en [10], [28] y [8]).

La utilidad actual de este tipo de sistemas está en aplicaciones que necesiten de reconocimiento de patrones (como el análisis de imágenes y compactación de voz) o cuando existen sistemas complejos que tienen valores de entrada y es necesario obtener las posibles respuestas. Los procesos basados en instrucciones, en cambio, tienen un muy mal desempeño bajo este esquema (ver [6]).

4.3.2 Estructura de procesamiento de las redes neuronales

Las redes neuronales son utilizadas en el tratamiento de altos volúmenes de datos en tareas de clasificación y segmentación.

En la mayoría de los modelos existen fases de operación diferenciadas:

- **Fase de aprendizaje:**
  Fase en que la red ajusta sus parámetros para obtener los datos necesarios en el proceso de reconocimiento. El proceso de aprendizaje permite clasificar los algoritmos en supervisados y no supervisados. Las redes supervisadas requieren presentar la salida esperada al momento

del aprendizaje. En los mecanismos no supervisados, la red ajusta sus parámetros para poder registrar la entrada a un nodo en particular o cluster.

- **Fase de reconocimiento:**
  Fase en que la red realiza la clasificación de los patrones de entrada. Los parámetros de operación permanecen fijos.

Sin importar el paradigma de aprendizaje, los mecanismo basados en redes neuronales ajustan sus parámetros de operación utilizando básicamente la regla de Hebb (ver detalles en [10], [28] y [11]).
En la actualidad, las redes neuronales son utilizadas ampliamente en sistemas comerciales de alta complejidad (sistemas financieros, predicciones del comportamiento del mercado, etc.) y las nuevas técnicas de recolección de información en bases de datos masivas. Estas técnicas se desarrollan en el contexto de Data Mining (ver detalles [1] y [16]).

Los mapas auto organizativos son de principal interés en este trabajo, dado que los procesos de reconocimiento y clasificación que se esperan implementar consideran la independencia del usuario al momento de la fase de aprendizaje.

En este sentido, los mapas auto organizativos son mecanismos de clasificación y reconocimiento paralelo distribuido basados en técnicas no supervisadas.

Los usos más comunes de este tipo de red son:

Clustering:
Los datos de entrada son agrupados en clases D_j, que se identifican con etiquetas discretas, obtenidas a partir de la dinámica del sistema.

Cuantización de Vectores:
Los datos de entrada corresponden a un espacio continuo R^n que será discretizado. La entrada al sistema es un vector x, y el sistema tiene que encontrar la discretización óptima del espacio de entrada. Se define la función de distribución de probabilidad $p(x)$,que es desconocida, y está implícita en los datos de entrada y la estructura final de los valores sinápticos de la red.

Reducción dimensional:
Los vectores de entrada son agrupados en sub-espacios de menor dimensión. El sistema debe aprender un mapeo óptimo de tal manera que la varianza de los datos de entrada se preserve en los datos de salida, considerando el análisis previo de componentes principales del problema a ser estudiado.

4.3.3 Mapas auto-organizativos

Los modelos de Mapas Auto-Organizativos (SOM) fueron introducidos por T. Kohonen (ver [11]) e identifican a un tipo especial de Redes Neuronales artificiales de aprendizaje no supervisado.

Las ventajas de los mapas auto-organizativos radican en que son capaces de preservar la topología del espacio de los datos, proyectan datos altamente dimensionales a un esquema de representación de baja dimensión y tienen la habilidad de encontrar similitudes en los datos (ver [12]).

El algoritmo de las SOM consiste en un procedimiento iterativo capaz de representar la estructura topológica del espacio de entrada (discreto o continuo) por medio de un conjunto discreto de vectores de peso que se asocian a neuronas de la red.

Las SOM mapean los patrones de entradas a neuronas vecinas, permitiendo asociar conjuntos de vectores de entrada similares a las neuronas e identificando un nodo como el representante.

El mapa se genera al establer una correspondencia entre las señales de entrada y las neuronas se localizan en una cuadrícula discreta. La correspondencia se obtiene a través de un algoritmo de aprendizaje competitivo consistente en una secuencia de pasos de entrenamiento que modifica iterativamente el vector de pesos de los nodos cercanos. La cercanía se establece a través de una definición de distancia. Para este Trabajo de Titulación será considerada la distancia euclidiana.

Cuando una nueva señal llega, en este caso una trama, todas las neuronas compiten para representarla. La unidad que mejor se ajusta (*best matching unit*) es la neurona que gana la competencia. Posteriormente, la red ajusta sus pesos en conjunto con sus vecinos. Gradualmente, las neuronas vecinas se especializarán para lograr representar señales de entradas similares.

A grandes rasgos, lo que se hace es que, en aquellas zonas en las que la red tiene nodos con pesos que coinciden con vectores de entrenamiento, el resto de nodos de su entorno tienden a aproximarse también a ese mismo vector. De esta forma, partiendo de una distribución de pesos inicial (normalmente aleatorios), SOM tiende a aproximarse a una distribución de pesos estable. Cada una de estas zonas que se estabiliza se convierte en un clasificador de propiedades, permitiendo que la red se transforme en una salida que representa una aplicación de clasificación.

Una vez estabilizada la red, cualquier vector nuevo estimulará la zona de la red que tiene pesos similares.

Considerando la siguiente figura:

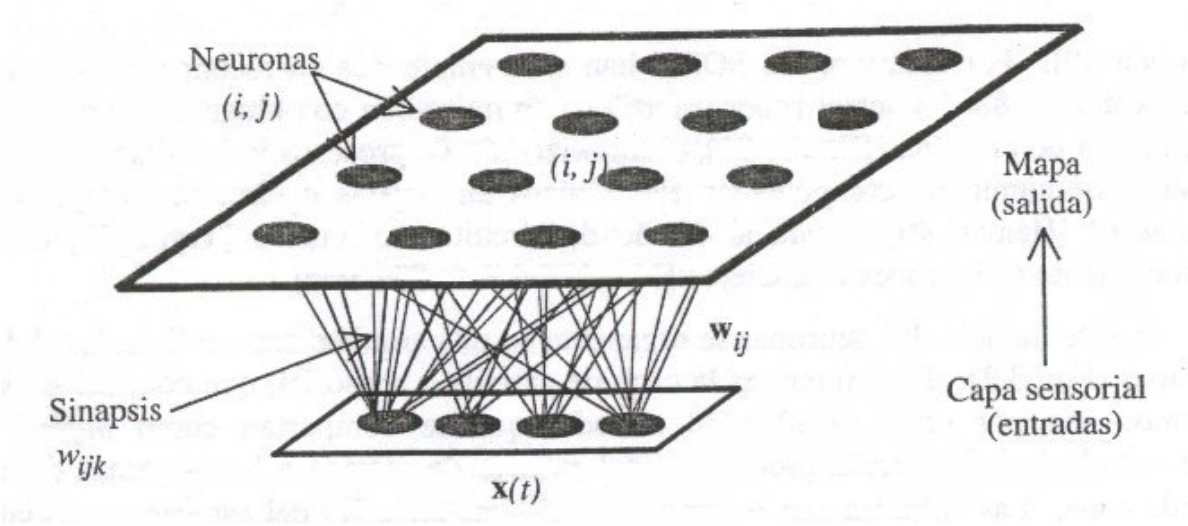

Ilustración 13: Componentes de una red SOM.

Se pueden establecer los parámetros relevantes:

Vector x(t): es el vector de entrada. En este caso corresponde a las tramas obtenidas como resultados de la transformación Wavelet de las señales de voz.

Vector de pesos w(i, j): identifica a los vectores de pesos de cada nodo (i, j). Estos vectores son modificados por las funciones de ajuste de la red SOM.

Neuronas o nodo (i, j): cuadrícula o malla de nodos en una estructura cuadrada, cuya dimensión es definida como parte de los resultados obtenidos.

Sinapsis w(i,j,k): representa la relación existente entre cada vector de entrada x(t) y cada nodo (i,j).

Capa sensorial: conjunto de vectores de entrada utilizados para el entrenamiento y posterior proceso de reconocimiento.

Mapa: estructura que es obtenida al finalizar un proceso de entrenamiento. En este caso, el proceso de entrenamiento se realizará utilizando transformaciones Wavelet de fonemas seleccionados. Cada fonema tendrá su proceso de entrenamiento, permitiendo obtener nodos representativos de cada uno de ellos.

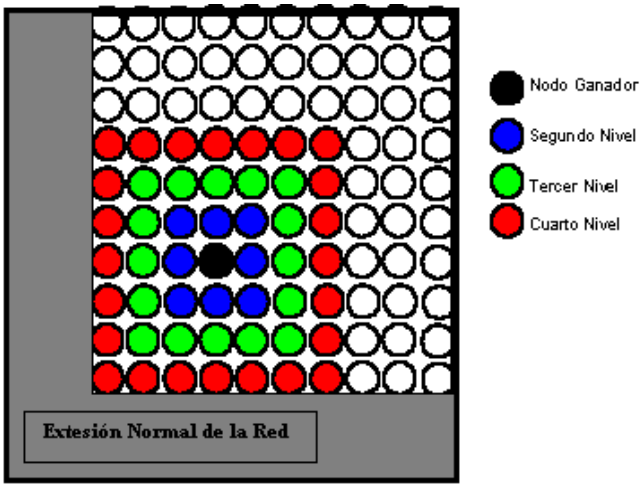

Ilustración 14: Representación de una vecindad alrededor del nodo ganador.

Los pasos que se siguen para el proceso de entrenamiento son:

1. Cada nodo se inicializa con un vector de pesos (aleatorio).

2. Se selecciona un vector del conjunto de entrenamiento. En este caso, una trama de entrada.

3. Se calcula el nodo de la red que tiene el peso más similar al vector anterior. Para ello, simplemente se calculan la distancia euclidiana entre los vectores de cada nodo y el vector de entrenamiento.
   La función de distancia euclidiana es la siguiente:

$$d^2(\mathbf{w}_{ij}, \mathbf{x}) = \sum_k (w_{ijk} - x_k)^2$$

4. Se calcula el radio del entorno. Este radio comenzará siendo grande (como para cubrir la red completa) y se va reduciendo en cada iteración, para un mismo vector de entrada.

5. Cada nodo del entorno ajusta su peso para parecerse al vector de entrenamiento seleccionado en el paso 2, de forma que los nodos más cercanos en la vecindad se vean más modificados.
   La función de ajuste es la siguiente:

$$\delta w_{ijk}(t) = \alpha(t) \cdot h(|\mathbf{i} - \mathbf{g}|, t) \cdot (x_k(t) - w_{ijk}(t))$$

Donde las variables son:

$\alpha(t)$: factor de ritmo de aprendizaje. Existen varias posibilidades para esta función, desde una constante hasta algún tipo de función monótona decreciente con el tiempo.

$h(|\mathbf{i} - \mathbf{g}|, t)$: es una función de vecindad que nos indica en qué medida se modifican los pesos de las neuronas vecinas. Es decir, cuando la neurona ganadora modifica sus pesos, la vecindad de esta neurona lo hace también, en mayor o menor medida, según sea la forma funcional de *h*. En general, las funciones empleadas para *h* tienen un máximo en |i-j|=0 y decrecen a medida que esta distancia aumenta.

$(x_k(t) - w_{ijk}(t))$: diferencia entre el vector de entrada y el vector de peso del nodo actualmente en ajuste. Todos los vectores utilizados están normalizados.

6. Repetir desde el paso 2 el número de iteraciones que se considere necesario.

Los pasos que se siguen para el proceso de reconocimiento son:

1. Se presenta a la red el vector que debe ser reconocido o clasificado.

2. La red busca el nodo de menor distancia euclidiana entre el vector de pesos y el vector de entrada.

3. Con este nodo seleccionado, es posible asignar valores a la red, como por ejemplo, cadenas de caracteres con algún valor. En el caso de la implementación de este Trabajo de Título, se almacenará un string con la identificación de un token.

Para la implementación de SOM, se considerarán un conjunto de parámetros que se explican en la sección 5.4.2 y 5.5.3. Por su parte, los valores de los parámetros utilizados se presentan en la sección 7.4.

4.4 Conceptos de fonética y fonemas

La fonética es la rama de la lingüística que estudia la producción y percepción de los sonidos de una lengua con respecto a sus manifestaciones físicas. La fonética tiene como objetivos determinar el modo en que los sonidos del habla se emplean con fines comunicativos en las lenguas naturales y explicar los mecanismos que condicionan tanto su producción como su percepción en el proceso del habla.

Un fonema es la representación abstracta y generalizada de un sonido con capacidad para diferenciar significados. Se escribe entre / /.

Los fonemas de un idioma pueden ser clasificados en subgrupos basados en sus apariencias visuales. Estos subgrupos son llamados visemas y pueden ser considerados como la equivalencia visual de los fonemas (ver [21] y [26]). Los visemas pueden ser descritos como formas clave de la boca donde cada forma vocal corresponde a uno o más fonemas.

Un fonema es una unidad fonológica diferenciadora, indivisible y abstracta. Es diferenciadora porque cada fonema se delimita dentro del sistema por las cualidades que lo distingue del resto y además es portador de una intención significativa especial. Por ejemplo, /p-e-s-o/ y /b-e-s-o/ son dos palabras que se distinguen semánticamente debido a que /b/ se opone a /p/ por la sonoridad.

Un fonema es indivisible dado que no se puede descomponer en unidades menores. En cambio, la sílaba o el grupo fónico sí pueden fraccionarse. Un fonema está compuesto por un conjunto de diversos elementos fónicos llamados rasgos distintivos, cuya combinación forma el inventario de fonemas. El inventario de rasgos distintivos es asimismo limitado y viene a constituir una tercera articulación del lenguaje (ver [21]).
Un fonema es abstracto, dado que no son sonidos, sino modelos o tipos ideales de sonidos.

Los fonemas se clasifican en dos grupos:

4.4.1 Fonemas Vocálicos

Los Fonemas Vocálicos (o Vocales) son aquellos fonemas que al ser articulados en la boca, el aire no encuentra obstáculos en su salida.

Los Fonemas Vocálicos se clasifican según lo siguientes criterios:

- **Localización (punto de articulación),** según la parte de la boca desde donde se articulan. Pueden ser:
  - Vocales Anteriores → /e/, /i/
  - Vocales Medias o centrales → /a/
  - Vocales Posteriores → /o/, /u/

- **Abertura (modo de articulación),** según la abertura que tiene la boca al pronunciar estos sonidos. Pueden ser:
  - Vocales de Abertura Máxima (o Abiertas) → /a/
  - Vocales de Abertura Media (o Semiabiertas) → /e/, /o/
  - Vocales de Abertura Mínima (o Cerradas) → /i/, /u/

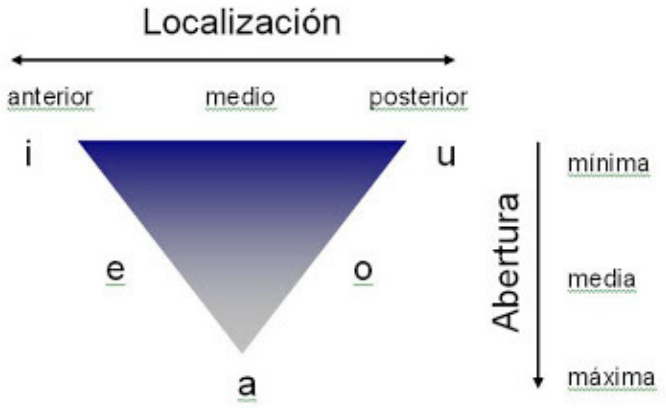

Ilustración 15: Clasificación de fonemas vocálicos. La imagen fue obtenida desde [21]

4.4.2 Fonemas Consonánticos

Los Fonemas Consonánticos son aquellos fonemas que al ser articulados en la boca, el aire encuentra obstáculos en su salida.

Los Fonemas consonánticos se clasifican de acuerdo con los siguientes criterios:

a. Según el Punto de Articulación (o zona de articulación). Indica el lugar donde se articula o produce el sonido:

- **Bilabial:** la articulación del sonido se produce entre ambos labios. Fonemas /p/, /b/, /m/.
- **Labiodental:** la articulación del sonido se produce entre el labio inferior y los dientes superiores. Fonema /f/.
- **Interdental:** la articulación del sonido se produce situando la lengua entre los dientes. Fonema /z/.
- **Dental:** la articulación del sonido se produce situando la lengua detrás de los dientes superiores. Fonemas /t/, /d/.
- **Alveolar:** la articulación del sonido se produce situando la lengua sobre la raíz de los dientes superiores. Fonemas /s/, /l/, /r/, /rr/, /n/.
- **Palatal:** la articulación del sonido se produce situando la lengua en el paladar. Fonemas /ch/, /y/, /ll/, /ñ/.
- **Velar:** la articulación del sonido se produce situando la lengua en el velo del paladar. Fonemas /k/, /g/, /j/.

b. Según el Modo de Articulación. Indica la postura que toman los órganos que intervienen en la articulación del sonido y que genera el obstáculo a la salida del aire:

- **Oclusivo:** se produce momentáneamente el cierre total del paso del aire. Fonemas /p/, /b/, /t/, /d/, /k/, /g/, /n/, /m.
- **Fricativo:** se produce un estrechamiento que provoca que el aire roce al exterior. Fonemas /f/, /z/, /j/, /s/.
- **Africado:** se produce una oclusión seguida de una fricación. Fonemas /ch/, /ñ/.
- **Lateral:** se produce un rozamiento del aire contra los dos lados de la cavidad bucal. Fonemas /l/, /ll/.
- **Vibrante:** ser produce un rozamiento del aire contra la punta de la lengua. Fonemas /r/, /rr/.

c. Según la Actividad de las Cuerdas Vocales.

- **Sordos:** son aquellos fonemas que al ser pronunciados no vibran las cuerdas vocales. Fonemas /p/, /t/, /k/, /ch/, /z/, /s/, /j/, /f/.
- **Sonoros:** son aquellos fonemas que al ser pronunciados, vibran las cuerdas vocales. Fonemas /b/, /z/, /d/, /l/, /r/, /rr/, /m/, /n/, /ll/, /y/, /g/.

d. Según la Actividad de la Cavidad Nasal.

- **Nasales:** aquellos fonemas que al ser pronunciados parte del aire pasa por la cavidad nasal. Fonemas /m/, /n/, /ñ/.
- **Orales:** aquellos fonemas que al ser pronunciados todo el aire pasa por la cavidad bucal. Son el resto de los fonemas.

5. DISEÑO DEL MECANISMO

En este capítulo se describe el diseño del mecanismo.
Las condiciones funcionales están definidas en el Capítulo 2, donde se establen los casos de uso y los requerimientos no funcionales.

5.1 Arquitectura

La arquitectura del mecanismo considera la implementación de cinco componentes que contendrán los programas para la implementación del mecanismo. Los componentes son los siguientes:

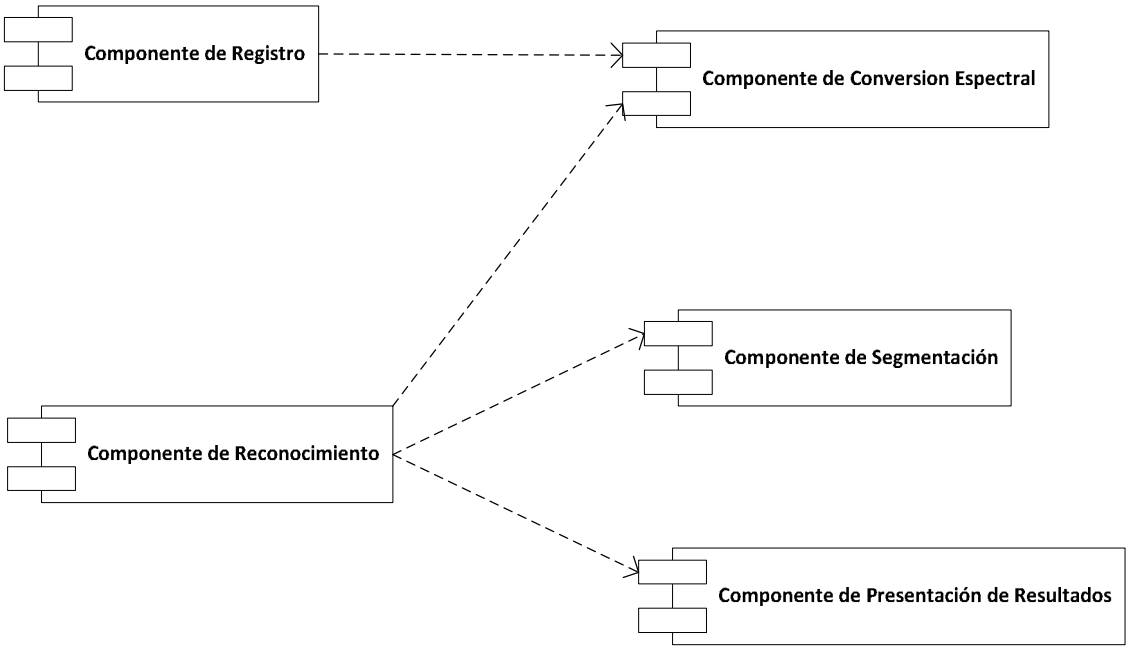

Ilustración 16: Componentes del mecanismo.

Componente de Registro
El componente de Registro tiene por objetivo empaquetar los programas necesarios para la creación del catálogo de fonemas y considerará las funciones de captura de datos para el análisis operacional del proceso.

Este componente implementará las funciones requeridas por el sistema descritas en el los casos de uso CUS02 y CUS05.

Además, utilizará los programas contenidos en el Componente de Transformación Espectral descritos en los puntos siguientes.

Componente de Reconocimiento
El componente de Reconocimiento tiene por objetivo empaquetar los programas necesarios para las funciones de identificación de fonemas descritos en el caso de uso CUS03, y considerará las funciones de captura de datos para el análisis operacional del proceso.

Utilizará los programas contenidos en el Componente de Transformación Espectral, Componente de Segmentación y Componente de Presentación de Resultados.

Componente de Transformación espectral
Este componente contendrá los programas necesarios para la conversión de los archivos de voz a espectros de frecuencia descritos en el caso de uso CUS04 y considerará las funciones de captura de datos para el análisis operacional del proceso.

Componente de Segmentación
Este componente contendrá los programas necesarios para la segmentación de frases compuestas, como es descrito en el caso de uso CUS06 y considerará las funciones de captura de datos para el análisis operacional del proceso.

Componente de Presentación de Resultados
Esta componente contendrá los programas necesarios para la presentación de los resultados al usuario del sistema, tal como es descrito en el caso de uso CUS07.

Las características comunes en cada uno de los componentes son:

- Se implementarán considerando los valores mínimos operacionales. Es decir, tiempo de proceso, CPU, etc.

- Serán encapsulados en módulos independientes, que utilizarán parámetros para el acceso a los recursos necesario en su operación.

- Todos los componentes se desarrollarán de acuerdo con una herramienta de software matemático, de tal forma que no sea necesaria la construcción de piezas de software adicionales o externas.

- La captura de la voz se realizará utilizando micrófonos estándares y software que permita generar archivos en formato WAV. El proceso de captura de los datos será offline en relación a los programas contenidos en los componentes.

- La captura de datos de voz no está considerada como parte del alcance funcional del mecanismo. Sin embargo es muy importante considerar este proceso, dado que dependiendo de la calidad de la grabación, el mecanismo observará variados comportamientos. Para efectos de esta implementación se considerará una frecuencia de muestro de 8 kHz, tal como se describe en los requerimientos no funcionales incluidos en la Sección 2.2.

Se generarán dos conjuntos de datos:

- Registros de voz con fonemas orientados al proceso de aprendizaje. Dichos fonemas serán utilizados como patrones.

- Registros de voz con palabras y frases, que serán utilizados por el mecanismo de reconocimiento. Los fonemas se ocuparán para obtener un análisis de los resultados.

5.2 Procedimiento de uso

Desde el punto de vista operacional, el mecanismo será utilizado en función de dos actividades sucesivas que el usuario debe completar:

Procedimiento de Registro de Patrones
Es la primera actividad, y permitirá al usuario registrar en el mecanismo los patrones a ser utilizados. En este caso, fonemas vocálicos y consonánticos.

El proceso implica registrar en una primera etapa todos los patrones que desea utilizar en su evaluación. Una vez terminado este proceso, queda disponible un archivo con los registros aprendidos que será utilizado en la actividad siguiente.

Procedimiento de Reconocimiento
Esta actividad permite al usuario verificar múltiples archivos con muestras de frases compuestas en relación con un archivo de patrón. De este modo, puede comparar los distintos procesos de reconocimiento que sean aplicados.

5.3 Mecanismo de Captura de Voz

El mecanismo de captura de voz considera la grabación de los fonemas que serán utilizados en el proceso de reconocimiento y las frases compuestas que serán utilizadas en las pruebas.

El proceso de captura de voz es el siguiente:

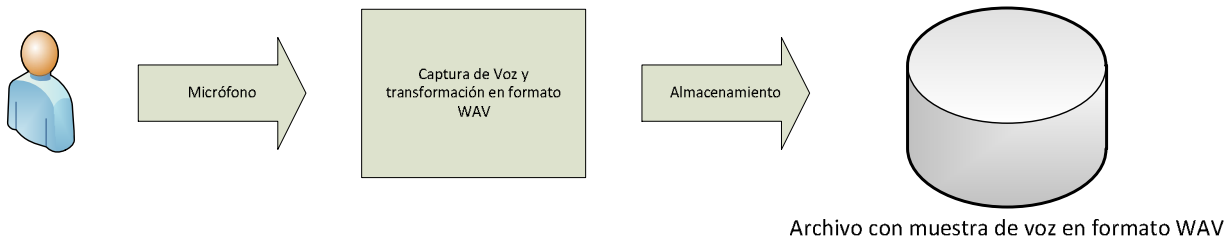

Ilustración 17: Procedimiento de captura de voz.

Los archivos de voz con los fonemas que servirán como patrones, serán organizados utilizando los siguientes criterios:

- Se usará un conjunto controlado de fonemas vocálicos y consonánticos.
- Se usará un conjunto controlado de grabaciones con frases que serán requeridas en el procedimiento de reconocimiento.

El catálogo de grabaciones será la base para la implementación y el estudio del comportamiento del mecanismo.

La herramienta utilizada para la generación de los archivos de voz es Audacity 2.0.5. Esta herramienta fue seleccionada por la amplia variedad de formatos de archivos de sonido que puede crear y reproducir, así como la posibilidad de utilizar múltiples frecuencias de muestreo en formato WAV.

Otra característica muy importante es la disponibilidad de licencias, dado que se trata de un producto de software de código libre.

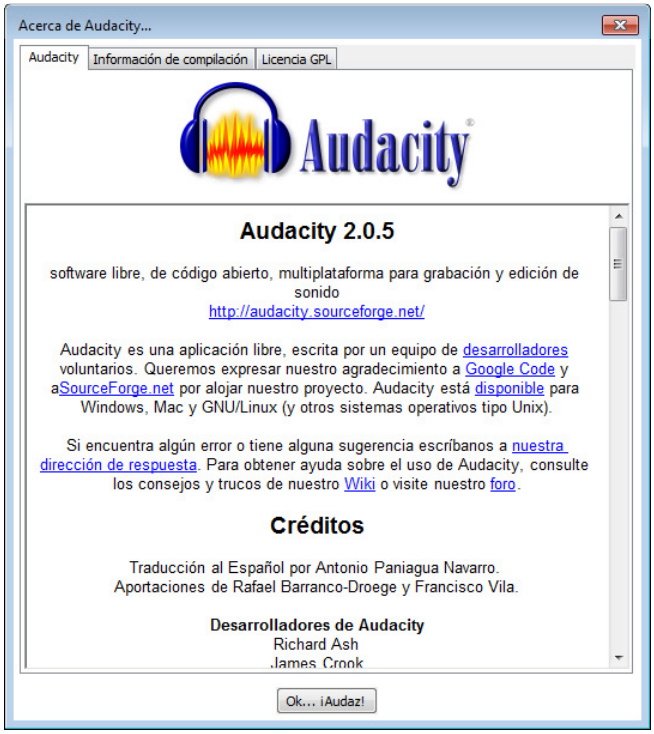

Ilustración 18: Herramienta utilizada para la grabación de fonemas y frases.

El programa de registro, considera la grabación de las muestras de voz.

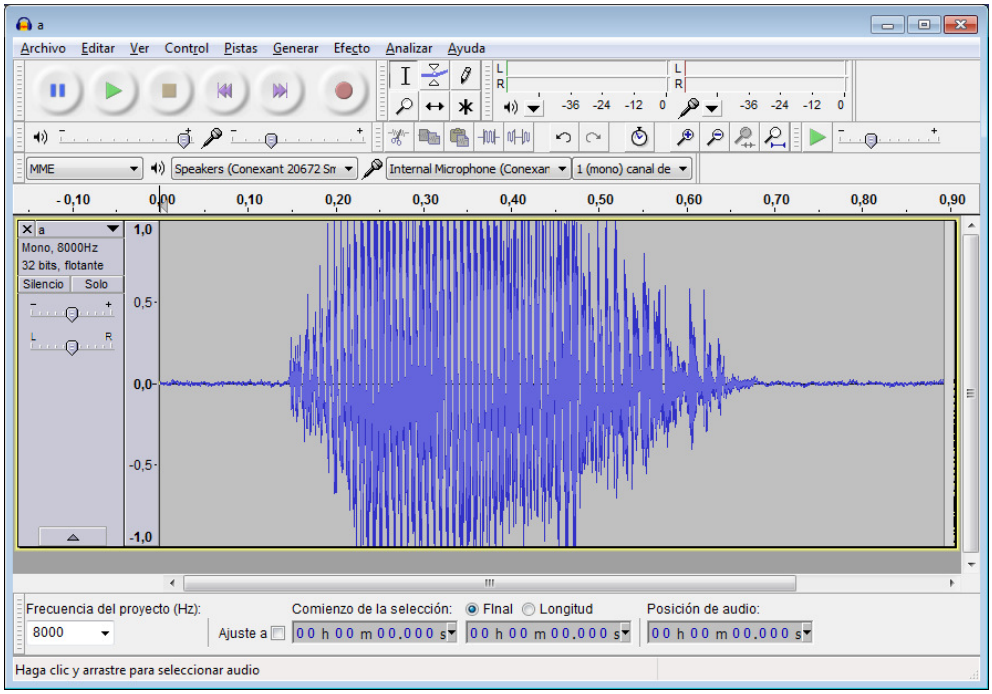

Ilustración 19: Interfaz gráfica de la herramienta Audaciity en el procedimiento de registro de las muestras de voz.

Una vez que las muestras ya están almacenadas en archivos WAV, desde SCILAB es posible observar su composición en el tiempo, utilizando funciones contenidas en el producto (ver sección 11.1).

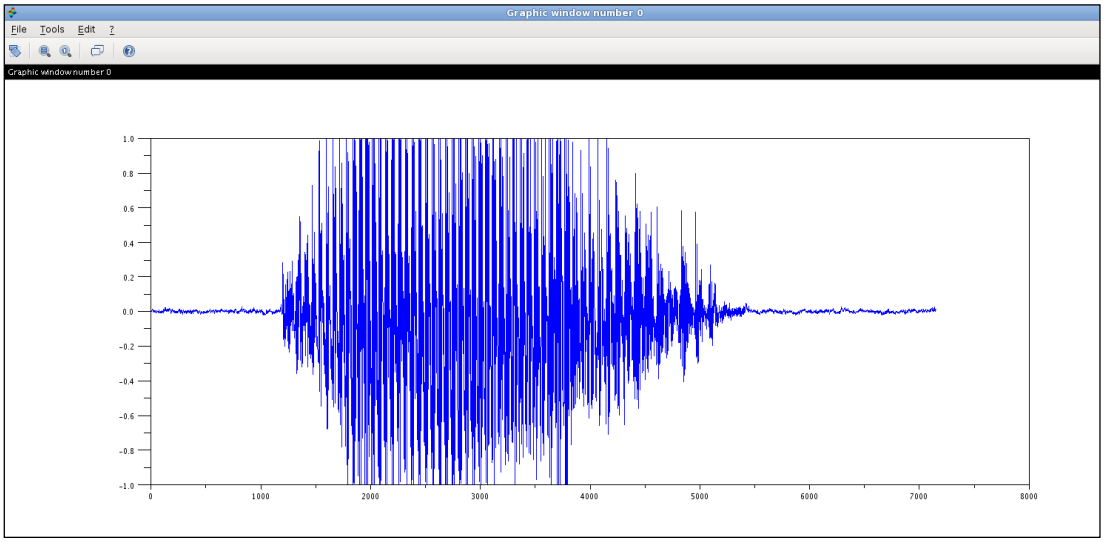

Ilustración 20: Gráfica obtenida en SCILAB de una muestra de voz de un archivo WAV.

5.4 Mecanismo de Registro

El mecanismo de registro utilizará los programas definidos en los componentes descritos en el punto 5.1.

Estos programas consideran la utilización de transformadas Wavelet basadas en banco de filtros para la obtención de espectros de frecuencia QMF (ver capítulo 4.2).

5.4.1 Procedimiento de Registro

Los programas consideran la implementación de un algoritmo basado en redes neuronales, en el que el registro se lleva a cabo a través un procedimiento de aprendizaje no supervisado, utilizando un modelo de redes SOM (ver Capitulo 4.3).

Este mecanismo permite organizar vectores n dimensionales obtenidos desde el programa de generación de espectros. La red resultante se almacena en un archivo que será utilizado posteriormente en el mecanismo de reconocimiento.

Esta red es la que contiene el conocimiento adquirido en el proceso de registro, y por ello se habla indistintamente de proceso de aprendizaje.

El diagrama de operación de los programas es el siguiente:

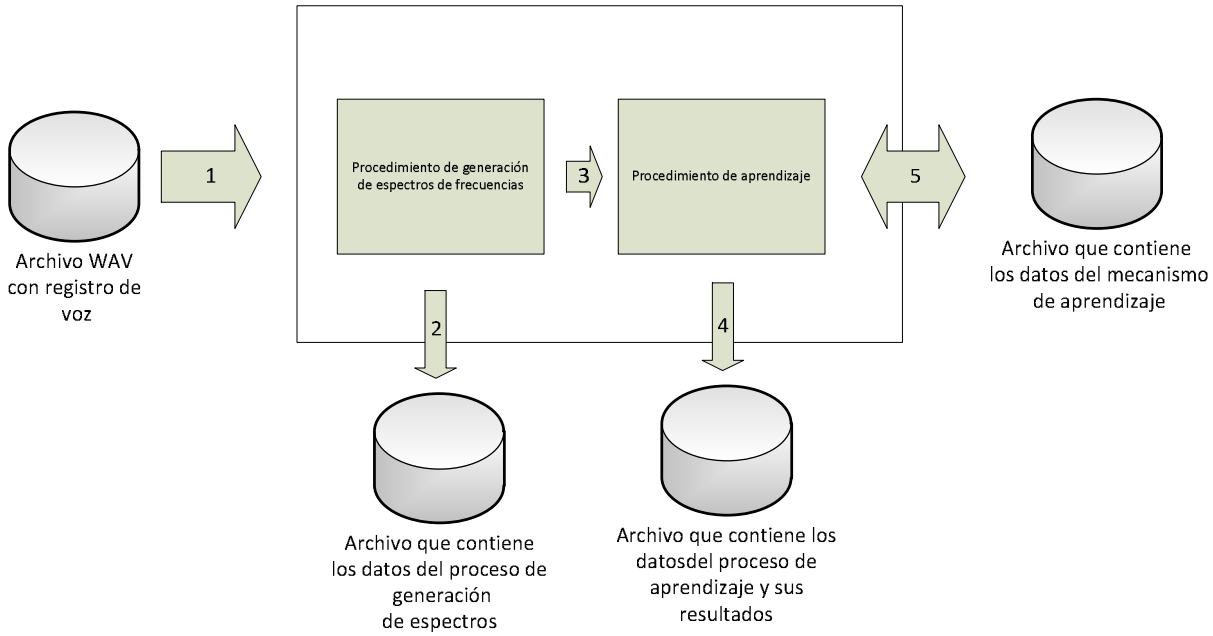

Ilustración 21: Secuencia de operación del mecanismo de aprendizaje. Considera el procedimiento de generación de espectros de frecuencias y posteriormente el procedimiento de aprendizaje.

El procedimiento de registro posee los siguientes pasos:

1. El mecanismo de reconocimiento recibe los parámetros:
   - Referencia a archivo de voz con patrón que se enseñará al mecanismo.
   - Referencia a archivo con los datos del mecanismo de aprendizaje.
   - Token con el símbolo que representará el contenido del archivo y que será utilizado en el despliegue en el procedimiento de reconocimiento.

2. Se registran los datos del comportamiento del proceso de generación de espectros. Estos datos permiten identificar los niveles operacionales del mecanismo.

3. Una vez generado el espectro del patrón, se establecen las tramas utilizadas por el mecanismo de aprendizaje. En el proceso de generación de espectros se aplicarán las transformaciones de pre-énfasis y normalizaciones requeridas (ver capítulo 4.2).

4. Se registran los datos del procedimiento de aprendizaje del programa SOM. Estos datos permiten verificar los niveles operacionales del mecanismo.

5. El mecanismo de aprendizaje utiliza un archivo externo que contiene la representación actual del conocimiento adquirido. En el caso de una red neuronal, este archivo contiene los pesos de las conexiones de todos los nodos de la red y se genera como resultado final del procedimiento de registro.

Los espectros de frecuencias se obtienen con el procesamiento de señales de una dimensión a raíz de la utilización de librerías (ver sección 6.1). Este proceso considera el uso de una ventana de tiempo que se desliza en el vector de entrada, permitiendo la obtención de un espectro de frecuencia.

Para la implementación del mecanismo de registro fue utilizado el paquete SCILAB 5.3

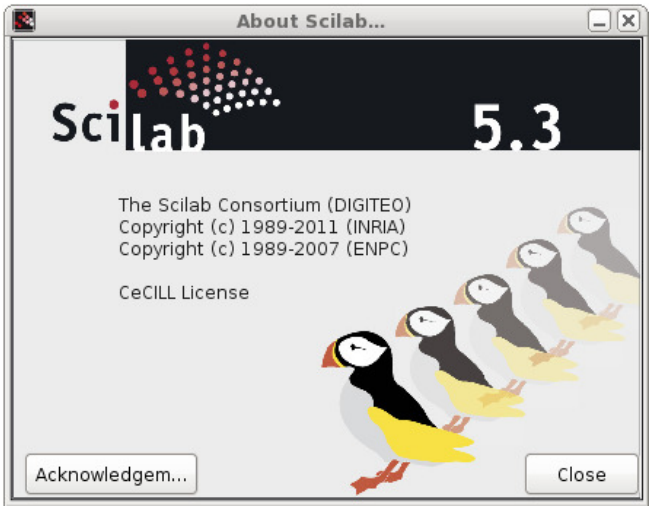

Ilustración 22: Paquete matemático utilizado en la implementación del mecanismo de registro

Este paquete fue seleccionado por las siguientes razones:

- Disponibilidad de paquetes para transformaciones Wavelet.
- Librerías para el manejo de archivos WAV.
- Uso de estructuras matriciales para la implementación de una red neuronal.
- Disponibilidad de licencia y documentación.

5.4.2 Transformación Wavelet

El mecanismo utilizará archivos de voz digitalizados en formato WAV con una frecuencia de muestreo de 8 kHz. Desde el punto de vista del mecanismo de registro, es presentado un conjunto de vectores en el tiempo. Para procesar esta entrada y aplicar los filtros Wavelet, se requiere establecer una ventana en el tiempo que permita transformar los grupos de vectores de entrada y obtener tramas, que serán presentadas a la red SOM.
El algoritmo considera un ciclo principal sobre todos los vectores de entrada, estableciendo ventanas en el tiempo, donde grupos de vectores son utilizados por los filtros Wavelet para obtener una trama.

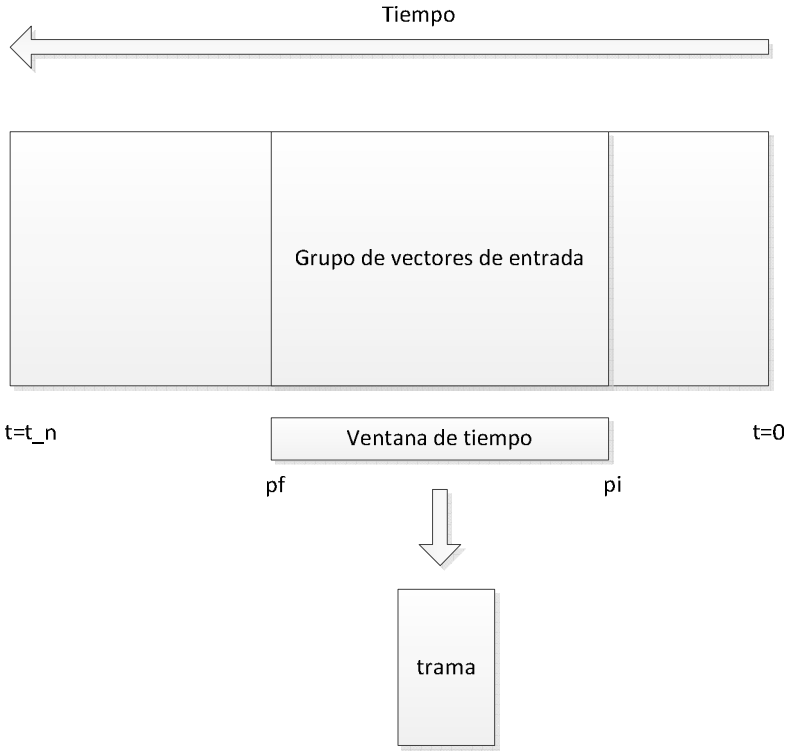

Ilustración 23: Procedimiento de transformación de los vectores originales extraídos desde los archivos WAV a tramas que serán utilizadas por la red SOM.

Los parámetros utilizados son:

- **t:** variable de tiempo, donde los valores límite son 0 y t_n (tamaño total del arreglo de vectores obtenidos desde el archivo WAV).

- **pi, pf:** posición inicial y final de la ventana de tiempo utilizada como entrada a los filtros Wavelet.

Para la obtención de espectros de frecuencia, es necesario establecer un traslape de los vectores de entrada, que se logra considerando dos parámetros adicionales:

- **dt:** cantidad de vectores de entrada a ser considerados en los filtros Wavelet. Corresponde a (pf - pi).

- **t_d:** paso definido en el ciclo principal para recorrer el arreglo de vectores de entrada.

En este caso t_d < dt.

El ciclo principal se estructura de la siguiente forma:

```
variableEntrada = LecturaArchivoWAV(archivo_wav);
t_n     = tamaño de la entrada;
dt      = valor del tamaño de la ventana en el tiempo;
t_d     = valor del paso en el ciclo sobre el arreglo de vectores iniciales;

ciclo desde t=0 hasta t_n, incrementando en t_d
        pi = t;
        pf = pi+dt;
        si excede el tamaño t_n, fin;
        Aplicación de pre-enfasis al vector de entrada;
        res = transformacionWavelet(entrada(pi:pf),dim_niv_des,tipo_wvt);
        tramas = tramas concatenado con res;
fin ciclo
```

Los parámetros considerados son los siguientes:

Tipo de Parámetro	**Nombre del Parámetro**	**Objetivo del Parámetro**
Parámetro Wavelet	dim_niv_des	Nivel de profundidad de los filtros Wavelet (QMF)
Parámetro Wavelet	tipo_wvt	Identificación de función madre Wavelet
Parámetro ventana de tiempo	dt	Tamaño de ventana de tiempo en la entrada
Parámetro ventana de tiempo	t_d	Valor de paso en el muestreo en la entrada

En los filtros Wavelet son necesarios dos parámetros:

- **Tipo de función Madre**
  - Es necesario indicar qué familia de funciones madres van a ser utilizadas (ver Sección 4.2)

- **Profundidad de la aplicación de los filtros**
  - Este parámetro define adicionalmente el tamaño de los vectores resultantes de la transformación Wavelet. Es decir, el tamaño de las tramas (ver Sección 4.2).

El siguiente ejemplo representa la aplicación del algoritmo a un archivo WAV:

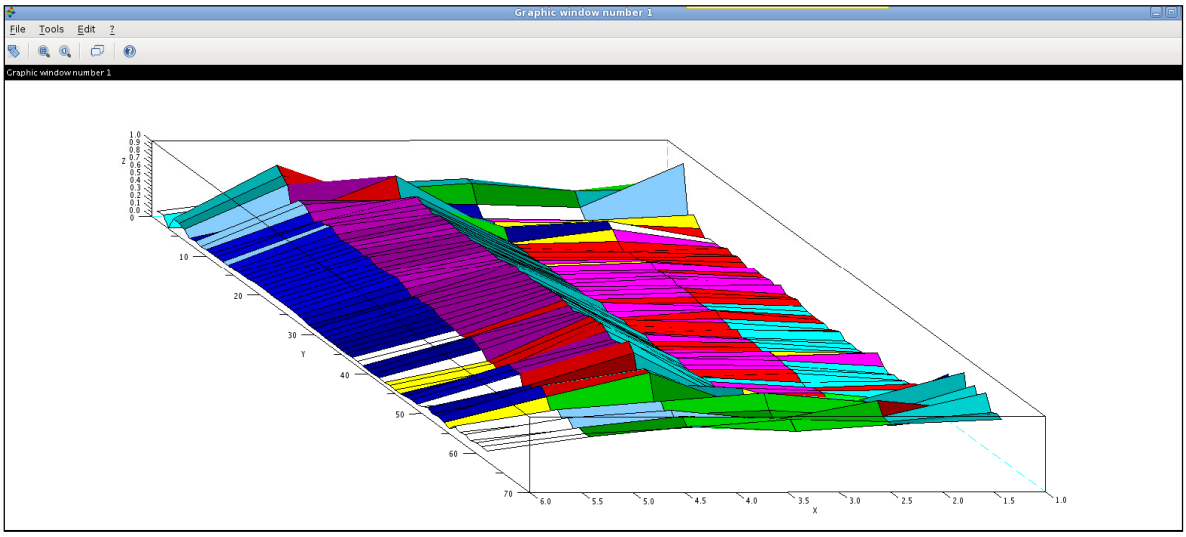

Ilustración 24: Espectrograma obtenido como resultado de la aplicación de filtros Wavelet. Este diagrama se obtiene al generar un gráfico desde la herramienta SCILAB.

Donde los ejes representan los siguientes datos:

- **Eje x:** Este eje representa los 6 canales o coeficientes obtenidos al aplicar la transformación Wavelet, considerando que se han utilizado Daubechies 2 (variable tipo_wvt), con un nivel de profundidad 6 (variable dim_niv_des).

- **Eje y:** Este eje representa el tiempo, en el que se produce una reducción temporal, debido a la transformación.

- **Eje z:** Este eje representa el valor de la transformada, considerando la normalización de sus valores (amplitud).

5.4.3 Algoritmo de aprendizaje SOM

Para la implementación del mecanismo de aprendizaje, se considera el desarrollo de una red neuronal basada en SOM (ver Sección 4.3).

El algoritmo a implementarse utiliza un arreglo de vectores que contienen los pesos y funciones para el cálculo de la distancia y la obtención de los mínimos en cada iteración.

Por su parte, el mecanismo de registro utiliza dos modos de trabajo, en los que el parámetro definido como **topera** identifica una de las dos opciones siguientes:

- **topera="c" :** modo creación. Se crea un nuevo archivo de red, en el que se inicializan los pesos de todos los nodos.
- **topera="r" :** modo lectura. Se carga un archivo de red ya existente, recuperando los pesos registrados y procediendo a almacenar el resultado obtenido con el nuevo token.

Para mayor claridad, se presentan los algoritmos según el modo de operación del mecanismo.

A. Modo creación

Pesos = inicialización y normalización de vectores de pesos de la red;

Ciclo de iteraciones por cada trama de entrada

    Ciclo de iteraciones sobre la Red,

        Cálculo de distancia entre trama y pesos de cada nodo;

        Selección de nodo con distancia mínima;

        Ajuste de pesos del nodo ganador y de sus nodos vecinos;

    fin ciclo

fin ciclo

B. Modo lectura

Pesos = lectura de archivo de red con los pesos de todos los nodos y los datos de los tokens registrados;

Ciclo de iteraciones por cada trama de entrada

    Ciclo de iteraciones sobre la Red,

Cálculo de distancia entre trama y pesos de cada nodo;

Selección de nodo con distancia mínima;

Ajuste de pesos del nodo ganador y de sus nodos vecinos;

fin ciclo

fin ciclo

Los criterios de ajuste están descritos en la Sección 4.3.

El archivo de red contiene los siguientes datos de cada token registrado:

- Símbolo de los tokens.
- Nodos representantes de cada token registrado en la red.

Cada vez que el mecanismo actúa en modo lectura, se recuperan todos los datos, el nuevo token es procesado y finalmente es almacenado el archivo con los nuevos datos.

5.5 Mecanismo de Reconocimiento

El mecanismo de reconocimiento utilizará los programas organizados en los componentes descritos en el punto 5.1.

5.5.1 Procedimiento de Reconocimiento

Al igual que el mecanismo de registro, se utilizan las mismas transformaciones espectrales y la implementación de un mecanismo de reconocimiento basado en la red SOM.

El mecanismo de reconocimiento actúa sobre frases compuestas, por ello requiere el uso de un algoritmo de segmentación que permita identificar palabras y posteriormente contrastar las palabras con el registro creado en el mecanismo anterior.

El diagrama de operación de los programas es el siguiente:

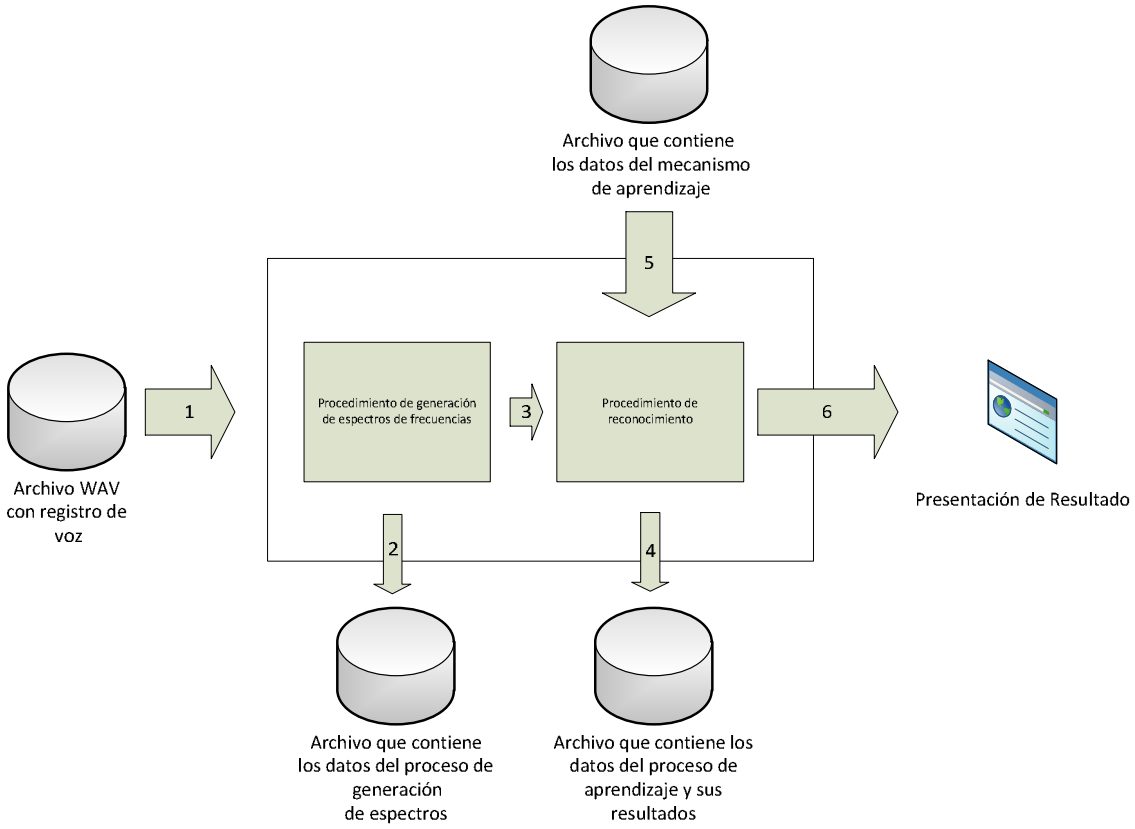

Ilustración 25: Procedimiento de reconocimiento con los componentes que lo conforman.

El proceso de reconocimiento posee los siguientes pasos:

1. El mecanismo de reconocimiento recibirá los siguientes parámetros:
   - Referencia a archivo con registro de voz que contiene las frases compuestas.
   - Referencia a archivo con datos del mecanismo de reconocimiento. Es decir, la red neuronal.

2. Se registran los datos del comportamiento del proceso de generación de espectros. Estos datos permiten identificar los niveles operacionales del mecanismo.

3. Una vez generado el espectro con la entrada a procesar, se generan las tramas utilizadas por el mecanismo de reconocimiento. En el proceso de generación de espectros están consideradas las transformaciones de pre-énfasis y las normalizaciones requeridas. Además en este punto es donde son utilizadas las funciones de segmentación. Es decir, una entrada compleja de varias palabras se segmentará en grupos de tramas organizadas temporalmente. El mecanismo de reconocimiento analiza los grupos de tramas con los patrones almacenados.

4. Se registran los datos del procedimiento de reconocimiento, que permiten verificar los niveles operacionales del mecanismo.

5. El mecanismo de reconocimiento utiliza un archivo externo que contiene la representación actual del conocimiento adquirido. En el caso de una red neuronal, este archivo contiene los pesos de las conexiones de todos los nodos de la red y es utilizado solo como lectura.

6. El mecanismo presenta al usuario la secuencia de patrones reconocidos en el tiempo, utilizando los tokens registrados en el procedimiento de aprendizaje.

5.5.2 Algoritmo de segmentación de frases

En el mecanismo de reconocimiento es necesario considerar la implementación de un procedimiento de segmentación de las frases, con el fin de identificar unidades fonéticas que puedan estar presentes.

Como parte de este Trabajo de Título, se exploró la alternativa de identificar fonemas a través de patrones existentes en los espectros de frecuencias.

Al observar la composición espectral de una frase, podemos identificar un patrón en las bajas frecuencias, en el que es posible observar cómo al momento de producir un sonido, aparecen máximos locales.

Por ejemplo, para la palabra “CASERÍO”, es posible obtener los siguientes espectrogramas:

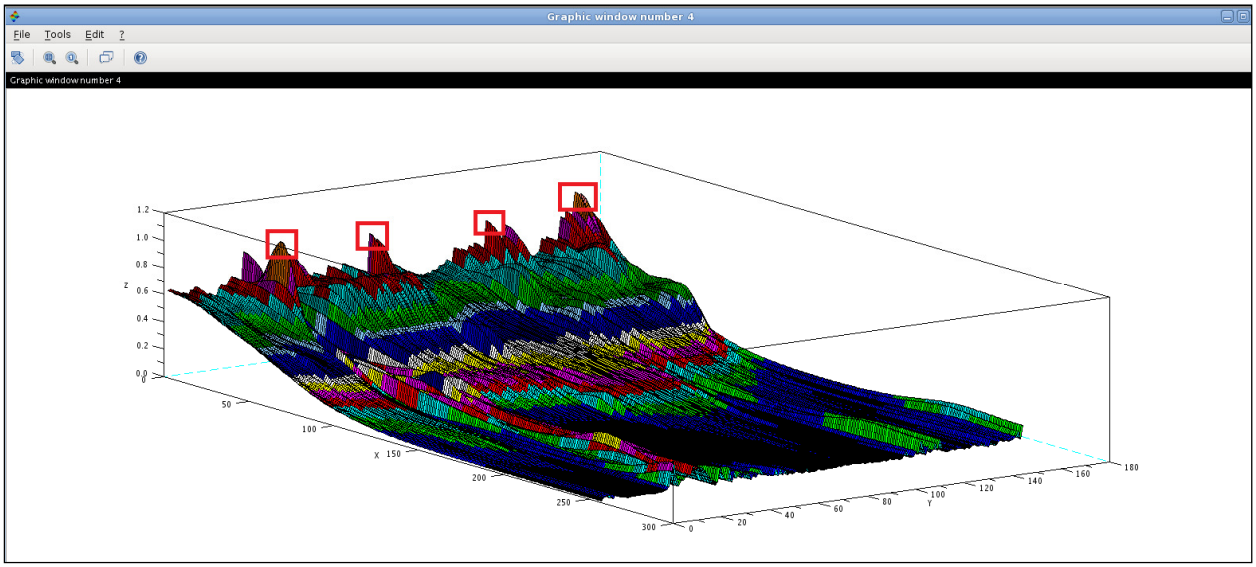

Ilustración 26: Espectrograma de la palabra CASERÍO, obtenido al realizar la transformación Wavelet en el mecanismo de reconocimiento. El gráfico fue generado en SCILAB.

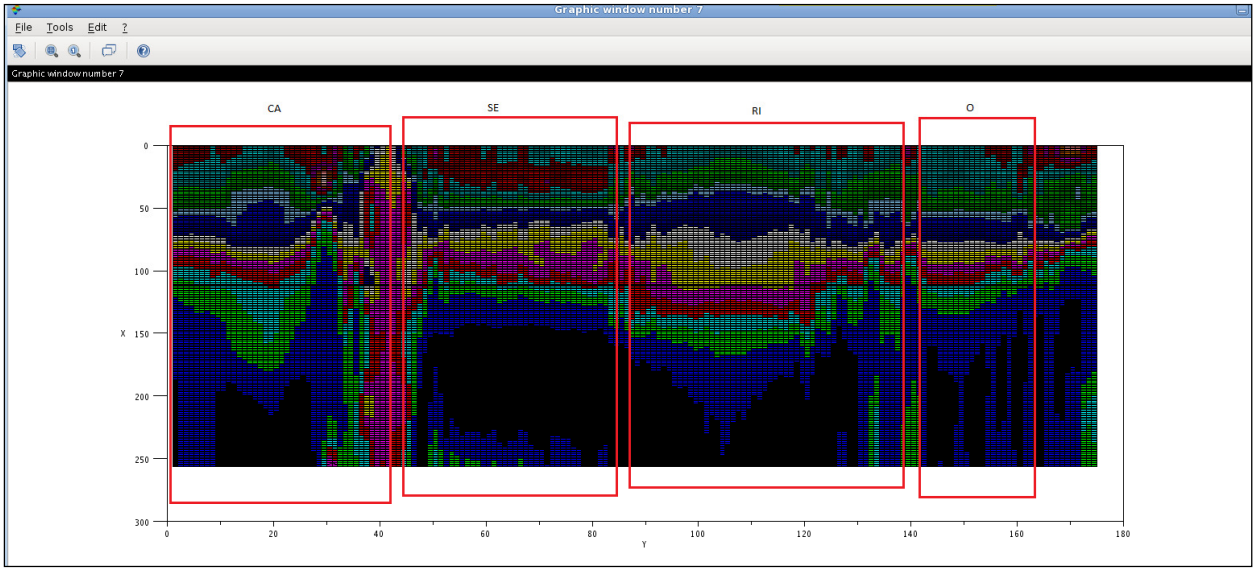

Ilustración 27: Espectrograma de la palabra CASERÍO, en el que se registra la representación de los tokens componentes de la palabra. Este gráfico fue obtenido utilizando SCILAB.

Con este resultado se procede a diseñar una función de segmentación que permite entregar al mecanismo de reconocimiento las posiciones en el tiempo donde se identifica un token.

El algoritmo es el siguiente:

Obtención del máximo valor espectral de todas las tramas;

Ciclo sobre cada trama

Obtención de un valor espectral desde la trama;

Si el valor de la trama es menor en un rango definido en relación al máximo:

Puntos = Puntos concatenados con el nuevo punto; (posición en el tiempo)

fin si

fin ciclo

Para la implementación del mecanismo de segmentación es necesario definir un valor escalar en función de los valores vectoriales de cada trama. En este caso, será utilizada la función de estimación de máxima entropía, función existente en las librerías de SCILAB. Este estimador ha sido aplicada en otros trabajos (ver [29] y [30]).

5.5.3 Algoritmo de reconocimiento SOM

El algoritmo de reconocimiento se basa en el procedimiento de reconocimiento existente en SOM (ver Sección 4.3).

La finalidad de este algoritmo consiste en encontrar el nodo con mínima distancia en relación a las tramas de entrada. El mecanismo utilizará la función de segmentación para seleccionar cada posible token.

El algoritmo es el siguiente:

Puntos = puntos en el tiempo obtenidos por la función de segmentación;

Ciclo sobre cada segmento obtenido

    Obtención del nodo con mínima distancia;

    Si existe registro del token en la red:

        Presentación del resultado;

    Fin si;

Fin ciclo

Cuando existe un nodo ganador, en relación a la trama de entrada, se presentan los textos registrados en el archivo de red. Si el nodo ganador no existe en el registro de tokens de la red, no es presentado ningún resultado.

5.6 Definición de fonemas a ser utilizados

Como parte del diseño, se han considerado todos los fonemas vocálicos y un subconjunto de fonemas consonánticos que son representativos del habla en español para las pruebas.

a(ha)	ba(va)	ca(ka)	cha	da	fa	ga	ja	la	ma	na	pa	ra	sa(za)	ta	ya
e(he)	be(ve)		che	de	fe		je	le	me	ne	pe	re	se(ce)	te	ye
i(hi)	bi(vi)		chi	di	fi		ji	li	mi	ni	pi	ri	si (ci)	ti	yi
o(ho)	bo(vo)	co (ko)	cho	do	fo	go	jo	lo	mo	no	po	ro	so(zo)	to	yo
u(hu)	bu(vu)	cu(ku)	chu	du	fu	gu	ju	lu	mu	nu	pu	ru	su(zu)	tu	yu

Siguiendo la misma lógica, se han construido frases (ver Sección 7.2) que contienen estos fonemas, de tal forma que se pudieran controlar los datos necesarios para el análisis de resultados.

5.7 Definiciones de costo computacional y observaciones al mecanismo

Para efecto del diseño del mecanismo, tal como se menciona en la Sección 5.1, se considerarán para su implementación los valores mínimos operacionales. Esto es, tiempo de proceso, CPU, etc.

Se define como costo computacional el esfuerzo requerido para el procesamiento de las señales de entrada. Factores como frecuencia de muestreo y tamaño de la red neuronal inciden en los tiempos de respuesta del mecanismo, tanto en la fase de registro como en la fase reconocimiento.

En el caso de la frecuencia de muestreo, a mayor frecuencia mayor es la cantidad de datos requeridos en la representación WAV.

En el caso de la cantidad de nodos en una red neuronal, a mayor tamaño de la red mayor es el esfuerzo requerido para las comparaciones y evaluaciones.

Para este Trabajo de Título, los valores se presentan en Capítulo 7. Estos valores son el resultado de experimentos realizados con el mecanismo.

6. CONSTRUCIÓN DEL MECANISMO

El objetivo de este capítulo es describir la construcción del mecanismo, considerando las distintas componentes de software desarrolladas y la presentación de los parámetros definidos en los programas.

6.1 Herramientas de Software utilizadas

Para la implementación del mecanismo fueron utilizadas las siguientes herramientas de software:

- **SCILAB 5**
  Herramienta de análisis matemático que permite la implementación de algoritmos numéricos. Esta herramienta fue seleccionada por su potencia, su disponibilidad a nivel de licencias y el conjunto de librerías matemáticas disponibles para la construcción del mecanismo.
  Los programas que conforman los distintos componentes se desarrollaron utilizando el lenguaje base de SCILAB, organizando su estructura en librerías de funciones.

- **SWT – Scilab Wavelet Toolbox**
  Librería nativa de MathLab, implementada en SCILAB, que permite el uso de un conjunto de algoritmos Wavelet. Implementa el algoritmo QMF para la generación de los espectros de salida.

- **AUDACITY**
  Editor digital de audio, de licencia libre, que permite la captura de la voz en diversos formatos, además de poseer herramientas para configurar los parámetros de muestreo, ruido y la calidad de las grabaciones.

6.2 Mecanismo de Registro

El objetivo de este punto es describir el programa de registro que se implementó en el Trabajo de Título.

6.2.1 Consideraciones generales

El mecanismo de registro se implementó considerando las siguientes premisas:

- El programa opera en dos modos:
  - **Modo nuevo aprendizaje**
    - En este modo, el programa genera un nuevo archivo de red neuronal, a raíz del primer aprendizaje.

- **Modo agregar**
  - En este modo, el programa utiliza un archivo de red neuronal ya existente y permite agregar nuevos patrones.

- Las muestras de voz a ser presentadas son tokens. Es decir, son fonemas específicos. En este mecanismo no existe la función de segmentación, dado que la muestra es una unidad de aprendizaje.

- Cada muestra es mapeada en conjunto con una etiqueta (Sting), que se utiliza en el procedimiento de reconocimiento para la presentación de su identificación y que está registrada en el archivo de red.

- El mecanismo de aprendizaje fue diseñado considerando autonomía en las distintas actividades que debe realizar, entre ellas, la obtención y transformación del archivo de voz, transformación a espectros de frecuencia basados en Wavelets y la generación de archivo con el conocimiento adquirido.

- El mecanismo fue desarrollado considerando la generación de un conjunto de archivos con datos para realizar la tarea de verificación de los límites operacionales del modelo, tal como es descrito en el diseño.

6.2.2 Estructura del programa

El programa está desarrollado según el siguiente diagrama:

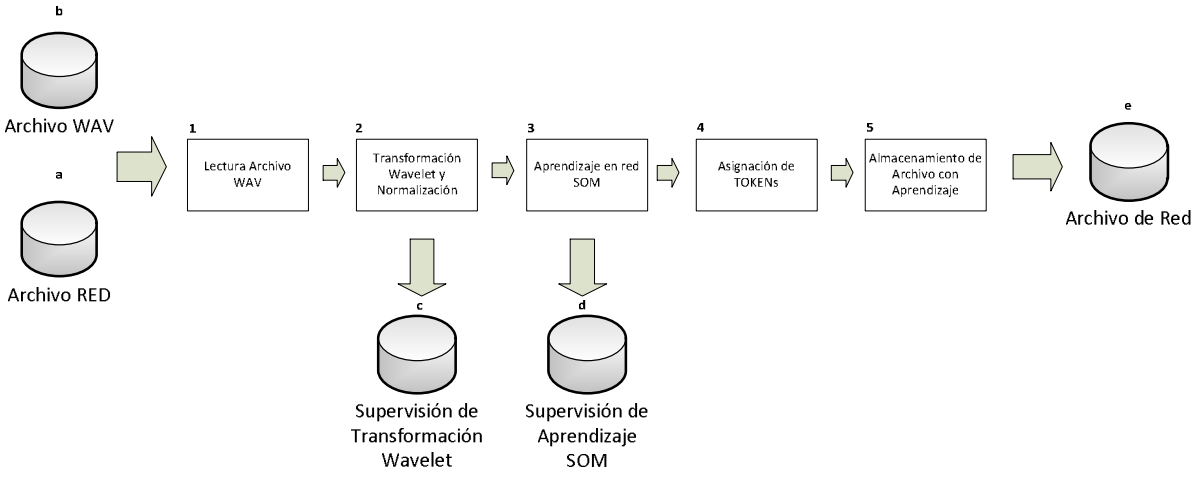

Ilustración 28: Secuencia de tareas que son ejecutas en el procedimiento de registro y aprendizaje de fonemas (tokens).

Los archivos utilizados son los siguientes:

a. **Archivo WAV con las muestras voz**. Muestras obtenidas a un frecuencia de 8 kHz en formato WAV.

b. **Archivo de red**. Este archivo se utiliza en la entrada. Cuando el programa actúa en modo crear, se genera uno nuevo. Cuando el programa actúa en modo leer, utiliza el archivo entregado en el parámetro.

c. **Archivo de medición y control** utilizado para supervisar el comportamiento de la transformación Wavelet. Este archivo permite ajustar los parámetros de funcionamiento de los algoritmos seleccionados.

d. **Archivo de medición y control** utilizado por proceso de aprendizaje en la red SOM. Este archivo permite ajustar los parámetros utilizados en la red neuronal y verificar los resultados.

e. **Archivo de red** que contiene el registro del aprendizaje. Este archivo es utilizado en nuevas secuencias de aprendizajes y en el proceso de reconocimiento.

El programa de registro implementa los algoritmos descritos en las secciones 5.3 y 5.4.

En el caso particular de la red neuronal, la implementación se ha realizado mediante un arreglo de vectores de pesos, en los que fueron utilizadas funciones de SCILAB para los cálculos de distancia y la obtención de los valores mínimos requeridos (ver Sección 11.1).

6.2.3 Parámetros de ajuste

El algoritmo implementado y los parámetros de operación se describen en la Sección 5.4.

El lenguaje utilizado es SCILAB.

En el mecanismo de aprendizaje, los parámetros operacionales se resumen en la siguiente tabla:

Tipo de Parámetro	Nombre del Parámetro	Objetivo del Parámetro
Transformación de espectros	dim_niv_des	Nivel de profundidad de QMF
Transformación de espectros	tipo_wvt	Identificación de función madre Wavelet
Operación red neuronal	dt	Tamaño de ventana de tiempo en la entrada
Operación red neuronal	t_d	Valor de paso en el muestreo en la entrada
Operación red neuronal	L	Cantidad de nodos en red neuronal
Operación red neuronal	Ciclos	Cantidad de ciclos de aprendizaje en la red
Operación red neuronal	Tau	Factor de ajuste en proceso de aprendizaje en la red
Operación red neuronal	tau_rate	Razón de decaimiento en cada ciclo de aprendizaje en la red

Los valores de los parámetros son presentados en el Capítulo 7

6.3 Mecanismo de reconocimiento

El objetivo de este punto consiste en describir el programa de reconocimiento implementado en el contexto de este Trabajo de Título.

6.3.1 Consideraciones generales

El mecanismo de reconocimiento fue implementado considerando las siguientes premisas:

- Las muestras de voz que son procesadas deben contener al menos un fonema. El mecanismo de reconocimiento fue diseñado para procesar frases compuestas, en las que será posible identificar los patrones aprendidos y registrados en la red neuronal, que es utilizada como parámetro de entrada.

- El mecanismo considera un algoritmo de segmentación que permite identificar palabras o fonemas en un primer nivel, dada la estructura espectral utilizada como entrada.

- El mecanismo fue desarrollado considerando la generación de un conjunto de archivos con datos para realizar la tarea de verificación de los límites operacionales del modelo.

- El mecanismo es capaz de reconocer las entradas que están registradas en el archivo de conocimiento (red SOM) y presentar en la salida las etiquetas que fueron utilizadas al momento de completar el aprendizaje.

6.3.2 Estructura del programa

El programa desarrollado está diseñado según el siguiente diagrama:

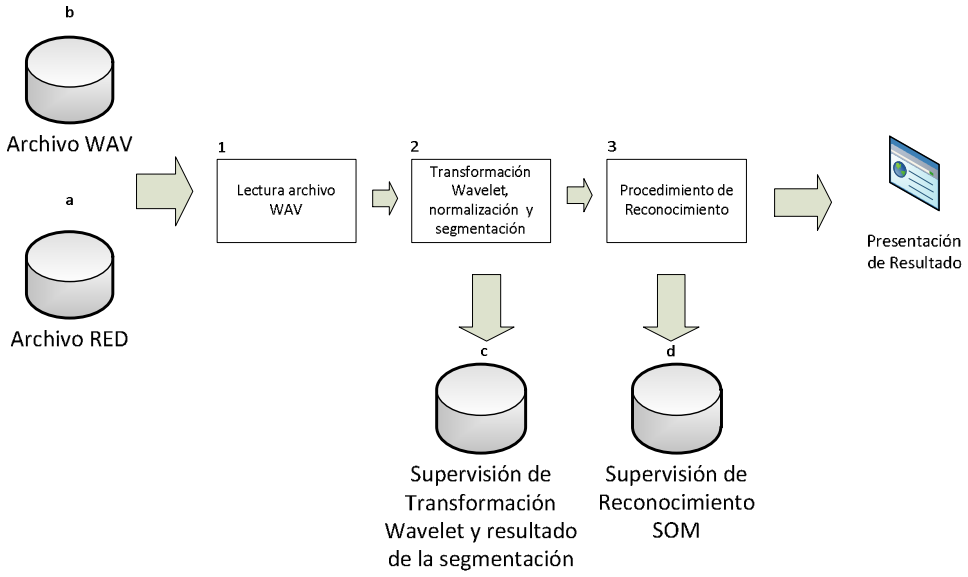

Ilustración 29: Secuencia de tareas que son ejecutadas en el procedimiento de reconocimiento. Es importante destacar el procedimiento de segmentación que permite identificar grupos de tramas de entrada que forman un token en el espacio espectral.

Los archivos utilizados son los siguientes:

a. **Archivo WAV con las muestras voz**. Inicialmente, muestras con frases grabadas a una frecuencia de 8 kHz en formato WAV.

b. **Archivo de red**. Este archivo contiene los resultados del proceso de aprendizaje.

c. **Archivo de medición y control** utilizado para supervisar el comportamiento de la transformación Wavelet y el proceso de segmentación implementado. Este archivo permite ajustar los parámetros de funcionamiento del algoritmo seleccionado.

d. **Archivo de red** utilizado por el proceso de reconocimiento en la red SOM. Este archivo permite ajustar los parámetros utilizados en la red neuronal y verificar los resultados.

La implementación de la red neuronal es la misma descrita en la Sección 6.2.2 y la implementación de la función de segmentación, descrita en la Sección 5.2.2, es construida considerando una función escalar que permite la obtención de un estimador de maxima entropía.

Es posible revisar la implementación en la Sección 11.3.

6.3.3 Parámetros de ajuste

El algoritmo implementado y los parámetros de operación se describen en la sección 5.5.

El lenguaje utilizado es SCILAB.

En el mecanismo de aprendizaje, los parámetros operacionales se resumen en la siguiente tabla:

Tipo de Parámetro	**Nombre del Parámetro**	**Objetivo del Parámetro**
Transformación de espectros	dim_niv_des	Nivel de profundidad de QMF
Transformación de espectros	tipo_wvt	Identificación de función madre Wavelet
Operación red neuronal	dt	Tamaño de ventana de tiempo en la entrada
Operación red neuronal	t_d	Valor de paso en el muestreo en la entrada

Los valores de los parámetros se presentan en el Capítulo 7.

7. PRUEBAS Y RESULTADOS

En este punto se describen los resultados de las pruebas y las condiciones de borde del mecanismo.

7.1 Condiciones de borde

Para la evaluación del comportamiento del mecanismo implementado, se establecen las siguientes condiciones de borde en la estructura de los tokes a utilizar:

- Los tokens serán un conjunto reducido de combinaciones de vocales y consonantes.
- Las frases contienen un conjunto controlado de tokens, que al ser procesados por el mecanismo de reconocimiento permiten la obtención de las métricas de análisis al aplicar la métrica f-measure.
- Los parámetros operacionales a nivel de transformación espectral son los mismos para ambos mecanismos, dado que deben ser coherentes.
- Los datos de composición de la red neuronal están auto contenidos en el archivo que representa el conocimiento adquirido y que es utilizado por el mecanismo de reconocimiento.
- Se utilizan muestras de voz a una frecuencia de 8 kHz, de calidad similar a la telefónica y con un formato de muestra de 32 bits flotantes.
- Para efectos del análisis, se consideran un conjunto de valores establecidos para los parámetros operacionales del mecanismo de aprendizaje y de reconocimiento. Los valores de los parámetros son los siguientes:

Mecanismo de Aprendizaje

Tipo de Parámetro	Nombre del Parámetro	Objetivo del Parámetro	Valor
Transformación de espectros	dim_niv_des	Nivel de profundidad de QMF	6
Transformación de espectros	tipo_wvt	Identificación de función madre Wavelet	db2
Operación red neuronal	dt	Tamaño de ventana de tiempo en la entrada	1000
Operación red neuronal	t_d	Valor de paso en el muestreo en la entrada	100
Operación red neuronal	L	Cantidad de nodos en red neuronal	100
Operación red neuronal	Ciclos	Cantidad de ciclos de aprendizaje en la red	3
Operación red neuronal	tau	Factor de ajuste en el aprendizaje en la red	1.0
Operación red neuronal	tau_rate	Razón de decaimiento del aprendizaje en la red	0.99

Mecanismo de Reconocimiento

Tipo de Parámetro	Nombre del Parámetro	Objetivo del Parámetro	Valor
Transformación de espectros	dim_niv_des	Nivel de profundidad de QMF	6
Transformación de espectros	tipo_wvt	Identificación de función madre Wavelet	db2
Operación red neuronal	dt	Tamaño de ventana de tiempo en la entrada	1000
Operación red neuronal	t_d	Valor de paso en el muestreo en la entrada	100

Estos valores fueron definidos considerando el mínimo costo computacional, según las consideraciones especificadas en la Sección 5.7.

7.2 Datos a ser utilizados

Implementación del Diccionario Base
Se define un vocabulario base, considerando tokens de aprendizaje y reconocimiento.

Los tokens son:

a(ha)	ba(va)	ca(ka)	cha	da	fa	ga	ja	la	ma	na	pa	ra	sa(za)	ta	ya
e(he)	be(ve)		che	de	fe		je	le	me	ne	pe	re	se(ce)	te	ye
i(hi)	bi(vi)		chi	di	fi		ji	li	mi	ni	pi	ri	si(ci)	ti	yi
o(ho)	bo(vo)	co (ko)	cho	do	fo	go	jo	lo	mo	no	po	ro	so(zo)	to	yo
u(hu)	bu(vu)	cu(ku)	chu	du	fu	gu	ju	lu	mu	nu	pu	ru	su(zu)	tu	yu

Frases de Reconocimiento
Los criterios de la toma de muestras son los mismos que en la captura del diccionario de tokens.

Las frases en el mecanismo de reconocimiento son:

Frase	Total Tokens
Mi casa está sola e invitaré a mucha gente	17
Me parece agotado	8
El paralelepípedo está en medio de la lista	17
El pájaro está parado en la casa	13
Mi saco está tirado	8
El ropero está solo	8
Tengo todo el monto tomado	10
Reparo la televisión	8
Tengo fe en la vida	7
Leseras solamente vi	8

El comportamiento del mecanismo se evalúa de acuerdo al procedimiento definido en el punto 3.3 Métricas de Calidad y Procedimiento de Análisis Comparativo.
Las frases fueron diseñadas a partir de los fonemas utilizados en la fase de registro.

7.3 Plan de pruebas

El plan de pruebas se inicia con el entrenamiento de la red neuronal con los tokens descritos en el punto anterior. Posteriormente, se aplica el mecanismo de reconocimiento, obteniendo los valores necesarios para aplicar las metricas las métricas descritas (ver sección 3.3).

7.4 Resultados obtenidos

Los resultados obtenidos en la identificación de fonemas vocálicos aislados son:

Vocal/Token	Total Tokes	A	B	C	Recall	Precison	F-measure
a	1	1	0	0	1	1	1
e	1	1	0	0	1	1	1
i	1	1	0	0	1	1	1
o	1	1	0	0	1	1	1
u	1	1	0	0	1	1	1
pa	1	1	0	0	1	1	1
pe	1	1	0	0	1	1	1
pi	1	1	0	0	1	1	1
po	1	1	0	0	1	1	1
pu	1	1	0	0	1	1	1
sa	1	1	0	0	1	1	1
se	1	1	0	0	1	1	1
si	1	1	0	0	1	1	1
so	1	1	0	0	1	1	1
su	1	1	0	0	1	1	1

Dados los resultados presentados, es posible concluir:

- La identificación de fonemas vocálicos fue completamente exitosa, salvo en aquellas situaciones donde la sensibilidad de red confundía fonemas de características espectrales similares (por ejemplo pu y tu). Estos fonemas no fueron considerados en las pruebas realizadas sobre las frases.

- El funcionamiento de la red neuronal fue adecuado al momento de aprender y reconocer fonemas vocálicos.

Los resultados obtenidos en el procesamiento de frases son:

Frase	Total Tokens	A	B	C	Recall	Precision	F-measure
Mi casa está sola e invitaré a mucha gente	17	6	7	4	0,4615	0,6000	0,5217
Me parece agotado	8	4	2	2	0,6667	0,6667	0,6667
El paralelepípedo está en medio de la lista	17	8	6	3	0,5714	0,7273	0,6400
El pájaro está parado en la casa	13	5	4	4	0,5556	0,5556	0,5556
Mi saco está tirado	8	3	4	1	0,4286	0,7500	0,5455
El ropero está solo	8	4	3	1	0,5714	0,8000	0,6667
Tengo todo el monto tomado	10	4	3	3	0,5714	0,5714	0,5714
Reparo la televisión	8	4	2	2	0,6667	0,6667	0,6667
Tengo fe en la vida	7	3	2	2	0,6000	0,6000	0,6000
Leseras solamente vi	8	3	3	2	0,5000	0,6000	0,5455

Dados los resultados presentados, es posible concluir:

- El mecanismo posee una baja precisión al considerar la segmentación de las muestras de voz más complejas. Esta situación se puede observar en los resultados presentados en la tabla.

- La identificación de los puntos de corte en una palabra o frase dependen de un conjunto de parámetros no analizados en este trabajo, como son los tonos utilizados en la grabación de aprendizaje y los tonos utilizados en las grabaciones de reconocimiento (frases).

- En esta implementación se consideran grabaciones hechas a una frecuencia de muestreo de 8 kHz y en formato WAV (ver Sección 2.2 en la que se especifican las condiciones de borde definidas).

7.5 Revisión comparativa de los resultados obtenidos

En los trabajos revisados [29] y [30], en los que se aplican los indicadores precision y recall al reconocimiento de patrones, se pueden observar valores superiores a los obtenidos por el mecanismo desarrollado en este Trabajo de Título.

En el primer caso, el trabajo de análisis se realiza sobre la base de una segmentación de frases utilizando técnicas basadas en lógica difusa. Los valores presentados en este trabajo son precision: 0.813901, recall: 0.697629.

En el segundo caso, el trabajo considera la evaluación de un mecanismo de reconocimiento de voz sobre telefónia IP, en el que los parámetros de precision y recall están indexados a la velocidad de transmisión. La mayoría de los valores de precision están sobre el 90 %.

8. CONCLUSIONES

De acuerdo con los objetivos planteados en este trabajo, es necesario revisar los siguientes puntos:

- Este trabajo ha significado un amplio estudio de los mecanismos para el procesamiento de la voz humana, situando el foco inicialmente en el uso de transformadas de Fourier y posteriormente implementando los procesos con transformadas más adecuadas a la realidad del problema, transformadas Wavelet.

- El estudio del uso de redes neuronales ha permitido diseñar un procedimiento autónomo de identificación de patrones basados en vectores, que han sido generados como resultado de la obtención de espectros de frecuencia. De este modo, el mecanismo podía identificar los patrones en el tiempo.

En la ejecución de las pruebas sobre el mecanismo, ha sido posible obtener resultados en ambas áreas de trabajo, que pueden resumirse de la siguiente forma:

8.1 Proceso de generación de Espectros de Voz

La generación de espectros de voz ha sido un aspecto fundamental, dado que se constituyó en la base para poder identificar patrones únicos en el habla.

La transformación de grabaciones de voz a grupos de vectores representativos ha supuesto un árduo trabajo de investigación, clave para la selección de las herramientas utilizadas. De la misma forma, el tratamiento del ruido, la aplicación de filtros para pre-énfasis, la normalización de los vectores y el uso de herramientas para la detección de niveles de energía en la voz humana ha requerido un proceso largo, con numerosas pruebas, hasta lograr un medio que permitiera aislar patrones en el espacio tiempo/espectral y así poder utilizarlos finalmente como una trama presentable a un mecanismo de aprendizaje y reconocimiento.

8.2 Comportamiento de la Red Neuronal

La implementación de una red neuronal basada en mapas auto organizativos presentó un gran desafío, por la necesidad de identificar los parámetros adecuados en su funcionamiento y las dimensiones necesarias para procesar los datos generados por los espectrogramas.

Como resultado del funcionamiento completo e integrado del mecanismo, es posible concluir que:

- El mecanismo presenta una baja tasa de reconocimiento en palabras y frases. Sin embargo, al utilizar fonemas vocálicos, su comportamiento es correcto (ver sección 7.4)

- El mecanismo posee un alto costo computacional en el procesamiento de la red neuronal (consideraciones en la sección5.7), no siendo factible en las pruebas realizadas una disminución de estos valores.

El Trabajo de Título realizado, por tanto, abre un conjunto de oportunidades para la investigación, principalmente en dos líneas:

- En el mejoramiento del mecanismo de obtención de espectros. En este trabajo se ha logrado llegar a la segmentación de palabras utilizando librerías propias de SCILAB, considerando distribuciones de energía y estimación de entropía máxima. Es posible extender el tema al estudio de otras transformaciones que permitan una segmentación de mejor calidad, permitiendo incluir fonemas consonánticos que con el mecanismo desarrollado producen confusión en el plano espectral.

- En el mejoramiento del mecanismo de aprendizaje y reconocimiento. En este punto se ha desarrollado un programa para lograr el registro de los datos procesados. Sin embargo, es posible revisar otros mecanismos existentes y medir sus resultados.

9. BIBLIOGRAFÍA

[1] Data Minig
Adriaans, P.
Addison-Wesley, Edinburgh Gate, UK 1997.

[2] Multiresolution Signal Descomposition: Transforms, Subbands and Wavelets,
Ali N. Akasu, Richard A. Haddad, 2001.

[3] Speech Enhancement
Jacob Benesty, Shoji Makino, Jingdong Chen
Springer, 29-04-2005.

[4] A Taxonomy of Self-organizing Maps for Temporal Sequence Processing.
Gabriela Guimarães, Victor Sousa Lobo,Fernando Moura-Pires
CENTRIA, Computer Science Department, New **University** of Lisbon, Portugal, 2001.

[5] Ten Lectures on Wavelet
Ingrid Daubechies, 2006.

[6] Redes Neuronales, Algoritmos, aplicaciones y técnicas de programación
Freeman, J.
Addison-Wesley/Diaz Santos, Wilington, Delaware, USA 1993.

[7] Towards Unsupervised Speech Processing MIT
James Glass
Computer Science and Artificial Intelligence Laboratory, 2011.

[8] Neuro-Fuzzy and Soft Computing
Jang, J.-S. R.
Pretince Hall, Upper Saddle River, 1997.

[9] Automatic Speech Recognition – A Brief History of the Technology Development
B.H. Juang, Lawrence R. Rabiner
Georgia Institute of Technology, Atlanta 2004.

[10] Foundation of Neural Networks
Khanna, T.
Addison-Wesley,Massachusetts 1989.

[11] Self-Organization and Associative Memory
Kohonen, T.

Springer-Verlag, Berlin 1984.

[12] The 'Neural' Phonetic Typewriter
Kohonen, T.
IEEE Computer 21(3) 1988, páginas: 11-22.

[13] Neural Networks for Signal Processing
Kosko, B.
Pretince Hall, University of Southern California, 1992.

[14] Neural Networks and Fuzzy Systems
Kosko, B.
Pretince Hall, University of Southern California, 1992.

[15] Reconocimiento de Formas y Visión Artificial
Maravall, D.
Addison-Wesley/Iberoamericana, Wilington, Delaware, USA 1994.

[16] Using Speech Recognition
Marowitz, J.
Pretince Hall PTR, New Jersey 1996.

[17] A Wavelet Tour of Signal Processing: The Sparse Way
Stephane Mallat, 2008.

[18] Sistemas Digitales y Analógicos, Transformadas de Fourier y Estimación Espectral
Papoulis, A.
Boixareu Editores, Barcelona 1986.

[19] Speech Segmentation Algorithm Based on an Analysis of the Normalized Power Spectral Density
Dzmitry Pekar and Siarhei Tsikhanenka
Belarusian State University, 2010.

[20] Wavelet, Time-Frequency, and Multirate Signal Processing
Ilya Pollak.
ECE 648 – Spring 2005

[21] Tratado de Fonología y Fonética Española
Quilis, A.
Editorial Gredos, Madrid.

[22] Noise Reduction of Speech Signal using Wavelet Transform with Modified Universal Threshold,
Rajeev Aggarwal y otros

International Journal of Computer Applications (0975 – 8887)
Volume 20– No.5, April 2011.

[23] Robinson T., “Speech Analysis”
Mphil in Computer Speech and Language Processing, Cambridge University, 1996.

[24] Wavelet Theory: An Elementary Approach with Applications
David K. Ruch,Patrick J. Van Fleet, 2011.

[25] Hybrid Speech Recognition System based on Wavelet 9/7 and Mel-Frequency Cepstral Coefficient
Sozan Mahmood, Mihran Abdulrahim
International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE'2012).

[26] Elementos de Lingüística
Martín Vide, C.
Editorial Octaedro Universal, Barcelona 1996.

[27] Memoria para optar al grado de Magister en Ciencias, mención Computación
Velásquez, J.
Departamento de Ciencias de la Computación-Universidad de Chile, 1998.

[28] Wasserman P., “Neural Computing, Theory and Practice”,
Van Nostrand Reinhold,
New York 1989.

[29] Wavelet Theory and Its Applications to Pattern Recognition
Yuan Y. Tang,
World Scientific,2008.

[30] Fuzzy Recall and Precision for Speech Segmentation Evaluation
Bartosz Zi´ołko, Suresh Manandhar, Richard C. Wilson
Department of Computer Science, University of York
Heslington, YO10 5DD, York, UK, 2007

[31] Reconocimiento de Voz Codificada sobre Redes IP
Jose Luis Carmona Maqueda
Dpto. de Teoría de la Señal Telemática y Comunicaciones
Universidad de Granada, 2009

10. REFERENCIAS WEB

[W1] Automated Source Classification Using Kohonen Net [en línea] <http://adsabs.harvard.edu/full/1995ApJ...452L..77M> [Consulta: 07/2014]

[W2] Speaker Independent Vowel Recognition [en línea] <http://elearn.sce.ac.il/users/www/15469/papers/Vowels.htm> [Consulta: 07/2014]

[W4] SOM Training [en línea]<http://www.cs.hmc.edu/~kpang/nn/som.html> [Consulta: 07/2014]

[W5] Tuning the Neural Networks Parameters [en línea] <http://www.sciencedirect.com/science/article/pii/S1026309811002136> [Consulta: 07/2014]

[W6] The evolution of speech recognition [en línea] <http://blog.spoken.com/2012/07/speech-recognition-evolution.html> [Consulta: 07/2014]

[W7] Wiener Filtering [en línea] <http://www.owlnet.rice.edu/~elec539/ Projects99/BACH/proj2/wiener.html> [Consulta: 07/2014]

11. ANEXOS

11.1 PROGRAMA PARA EL REGISTRO DE TOKENS

```
function [Patron,respuesta]=app(topera,archivo_wav,muestreo,archivo_net,simbolo)
        dim_niv_des             =  6;
        tipo_wvt                = 'db2';
        [ entrada,frecuencia]=wavread(archivo_wav);
         entrada= intdec(entrada,muestreo);
        t_n = size(entrada,2);
        dt = 1000;dp = 100 ;t_d = 100;tramas=0;
        for t=1:t_d:t_n
                pi=t ;
                pf=pi+dt;
                 if pf > t_n
                        break ;
                 end ,
                [c,l]=wavedec(entrada(pi:pf),dim_niv_des,tipo_wvt); // Transformacion a Espectros Wavelet
                [a,d]=wenergy(c,l);                                 //  Obtencion de vector de energia del espectro
                norma = norm(d);
                if norma <> 0
                        d = d/norma;
                end,
                 if t == 1
                   posEspectro = d';
                 elseif t>1
                   posEspectro = [posEspectro,d']
                end,
                tramas = tramas +1;
        end,
        M=posEspectro;L = 100;K = L^2;F = zeros(L,L);D = zeros(L,L);T = size(M,2);N = size(M,1);
        Ciclos = 1;x = M';delta = linspace(1,L,L)';
        if topera == 'c'
              W = abs(rand(K,N,'normal'));
              for j = 1:K
                      W(j,:) = W(j,:)/norm(W(j,:));
              end;
              for t = 1 : T;
                      tau = 1.0;
                      tau_rate =0.99999;
                      for d= 1 : 1;
                         XY = (ones(K,1) * x(t,:)) - W;
                         dist = sum(XY .* XY, 'c');
                         [winner_dist,k] = min(dist);
                         k_x = modulo(k-1,L) + 1;
                         k_y = (k - k_x)/L + 1;
                         dd_x = delta - k_x * ones(L,1);
                         d_x = dd_x .* dd_x / d;
                         dd_y = delta - k_y * ones(L,1);
                         d_y = dd_y .* dd_y / d;
                         psi =  (exp(- d_y) .*. exp(- d_x));
                         W = W + (tau*psi * ones(1,N)) .* (ones(K,1) * x(t,:) - W);
                         tau = tau * tau_rate;
                      end;    //fin ciclo por trama
              end;
      end,
```

```
if topera == 'r'
        load(archivo_net,'W','Patron');
        Lx=msscanf(size(Patron(:,2),1),Patron(:,2),'%i')'
        Ly=msscanf(size(Patron(:,3),1),Patron(:,3),'%i')'
        Bx = msscanf(Patron(1,2),'%i');
        By = msscanf(Patron(1,3),'%i')
        P=size(Patron,1);
        for t = 1 : T;
                 tau = 1.0;
                 tau_rate = 0.99999;
                for d=1 : 1;
                   XY = (ones(K,1) * x(t,:)) - W;
                   dist = sum(XY .* XY, 'c');
                   [winner_dist,k] = min(dist);
                   k_x = modulo(k-1,L) + 1;
                   k_y = (k - k_x)/L + 1;
                   dd_x = delta - k_x * ones(L,1);
                   d_x = dd_x .* dd_x / d;
                   dd_y = delta - k_y * ones(L,1);
                   d_y = dd_y .* dd_y / d;
                   psi =  (exp(- d_y) .*. exp(- d_x));
                   W = W + (tau*psi * ones(1,N)) .* (ones(K,1) * x(t,:) - W);
                   tau = tau * tau_rate;
                end;
        end;

end,
  F = zeros(L,L);
  D = zeros(L,L);
  for t = 1 : T;
          XY = (ones(K,1) * x(t,:)) - W;
          dist = sum(XY .* XY, 'c');
          [winner_dist,k] = min(dist);
          k_x = modulo(k-1,L) + 1;
          k_y = (k - k_x)/L + 1;
          F(k_x,k_y) = F(k_x,k_y)+1;
          if  t == 1
                 A(simbolo)= [ k_x k_y norm(x(t,:)) x(t,:)] ;
                 respuesta=[k_x,k_y,t];
          else
                 A(simbolo)= [A(simbolo);k_x k_y norm(x(t,:)) x(t,:)] ;
                 respuesta = [respuesta;k_x,k_y,t];
          end
 end;
 [valor,pos]=max(F);
  posV = pos ;
  Datosb=[simbolo,msprintf('%d',posV(1)),msprintf('%d',posV(2)),'1' ];
 if topera == 'c'
          Patron = [Datosb];
 end,

 if topera == 'r'
         Patron = [Patron;Datosb];
 end,


       if topera == 'f'
              if(p(find(p(:,:)=='i'),:)==[])
                   disp('patron no registrado....');
                   abort ;
              else
                 Patron(1,:)=['b',msprintf('%d',posV(1)),msprintf('%d',posV(2)),'1' ];
                 Patron(find(Patron(:,:)==simbolo),:)=[simbolo,msprintf('%d',pos(1)),msprintf('%d',pos(2)),'1' ];
              end;
       end,
endfunction
```

Descripción del programa

El programa implementa los siguientes pasos:

1. Definición de parámetros para los filtros Wavelet.
2. Lectura del archivo WAV.
3. Ciclo de transformación de datos de voz a espectros de frecuencia:
    a. Transformación Wavelet.
    b. Obtención de espectros de energía.
4. Procesamiento de la red neuronal:
    a. Si la opción de operación es creación:
        i. Inicialización de pesos con valores aleatorios..
        ii. Clustering de la primera entrada.
    b. Si la opción es agregación:
        i. Lectura del archivo de red existente de los datos registrados.
        ii. Clustering sobre una red ya existente.
5. Asignación de la etiqueta (símbolo u token).
6. Almacenamiento de los resultados en el archivo de red.

Observaciones

- En este proceso se utilizan tokens para el proceso de aprendizaje.

- El problema principal a resolver ha sido la estabilidad de la red neuronal, en términos de su sensibilidad a los aprendizajes sucesivos.

11.2 PROGRAMA PARA EL RECONOCIMIENTO DE TOKENS

```
function [respuesta,Patron]=rec(archivo_net,archivo_wav)
        exec sgm;
        [entrada,frecuencia]=wavread(archivo_wav);
        load(archivo_net,'W','Patron');
        puntos = sgm(archivo_wav);
        dim_niv_des = 6 ;
        tipo_wvt    = 'db2';
        t_n = size(entrada,2);
        dt = 1000;dp = 100;t_d = 100;tramas=0;
        for t=1:t_d:t_n
                  pi=t ;
                  pf=pi+dt;

                  if pf > t_n
                    break ;
                  end ,

                  [c,l]=wavedec(entrada(pi:pf),dim_niv_des,tipo_wvt);
                  [a,d]=wenergy(c,l);

                  norma = norm(d);
                  if norma <> 0
                        d = d/norma;
                  end,
                  if t == 1
                     posEspectro = d';
                  elseif t>1
                     posEspectro = [posEspectro,d']
                  end,
                      tramas = tramas +1;
     end,
     mwt=posEspectro;
     segmentos = size(puntos,2);
     for si = 1:1 : (segmentos-1);
             M= mwt(:,puntos(si):puntos(si+1));
             L = sqrt(size(W,1)) ;
             K = L^2;
             T = size(M,2);N = size(M,1);
             x = M';
             F = zeros(L,L);D = zeros(L,L);
             t_d= 0 ;t_p=1;
             for pi = 1:t_p : T;
                       TT =pi+t_d;
                       if TT > T
                          TT=T;
                       end ,

                      for t = pi:TT ;
                             XY = (ones(K,1) * x(t,:)) - W;
                             dist = sum(XY .* XY, 'c');
                             [winner_dist,k] = min(dist);
                             k_x = modulo(k-1,L) + 1;
                             k_y = (k - k_x)/L + 1;
                             F(k_x,k_y) = F(k_x,k_y)+1;
                      end;

                      [valor,pos]=max(F);
```

```
                        F = zeros(L,L);
                        P=size(Patron,1);
                        d_min = 100.0 ;
                        for i = 1 : P;
                              px=msscanf(Patron(i,2),"%i");
                              py=msscanf(Patron(i,3),"%i");
                              ld = sqrt( (pos(1)-px)^2+(pos(2)-py)^2);
                              if ld < d_min
                                      d_min = ld ;
                                      simbolo = Patron(i,1);
                              end
                         end;
                         if  pi == 1
                              respuesta= [ simbolo] ;
                         else
                              respuesta = [respuesta,simbolo];
                         end
                end;
                rsb=tabul(respuesta);
                [psim qsim]= max(rsb(2));
                if  si == 1
                        fres = [(rsb(1)(qsim))] ;
                else
                        fres = [fres,rsb(1)(qsim)];
                end
        end;
endfunction
```

Descripción del programa

El programa implementa los siguientes pasos:

1. Carga de la función de segmentación.
2. Lectura del archivo WAV.
3. Obtención de los puntos de segmentación en el tiempo de la señal de entrada.
4. Definición de parámetros para los filtros Wavelet.
5. Ciclo de transformación de datos de voz a espectros de frecuencia:
    a. Transformación Wavelet.
    b. Obtención de espectros de energía.
6. Reconocimiento de cada segmento en la red neuronal.
7. Obtención de la etiqueta (símbolo u token) registrado para el nodo ganador.

Observaciones

- En este proceso se transforman los archivos WAV en espectros de frecuencia que son segmentados para la obtención de los patrones.

- El problema principal a resolver ha sidol la estabilidad de la red neuronal, en términos de su sensibilidad en el reconocimiento de tramas.

11.3 PROGRAMA PARA LA SEGMENTACIÓN

```
function [puntos]=sgm(archivo_wav)
        dim_niv_des = 6;
        tipo_wvt    = 'db2';
        [entrada,frecuencia]=wavread(archivo_wav);
        t_n = size(entrada,2);
        dt = 1000;dp = 100 ;t_d = 100;tramas=0;
        for t=1:t_d:t_n
                 pi=t ;
                 pf=pi+dt;
                 if pf > t_n
                        break ;
                 end ,
                [c,l]=wavedec(entrada(pi:pf),dim_niv_des,tipo_wvt);
                [a,d]=wenergy(c,l);
                norma = norm(d);
                if norma <> 0
                        d = d/norma;
                end,
                [Ea,Eb]=mese(d,256); //Calculo de vector de entropia
                 if t == 1
                         posEspectro = d';
                         posEntropy = Ea';
                         p = max(Ea);
                         posPos = p;
                 elseif t>1
                         posEntropy = [posEntropy Ea'];
                         posEspectro = [posEspectro,d']
                         p = max(Ea);
                         posPos = [posPos p];
               end,
               tramas = tramas +1;
       end,
       mm=max(posPos)*0.75
       tt=0;puntos = [];
       for t=1:size(posPos,2)
               if posPos(t)< mm  // elimina el dato solo para visualizacion
                       posPos(t) = 0;
               end,
               if posPos(t)> 0         // se compone el vectos de posiciones
                              puntos = [puntos t];
               end,
       end,
       puntosPaso = [];
       i=0;
       for t=1:size(puntos,2)-1
               if puntos(t)< puntos(t+1)-40
                       puntosPaso = [puntosPaso puntos(t)];
                       i = i+1;
               end,
       end,
       puntos = puntosPaso;
       puntos =[puntos size(posPos,2)];
endfunction
```

Descripción del programa

El programa implementa los siguientes pasos:

1. Definición de parámetros para los filtros Wavelet.
2. Ciclo de transformación de datos de voz a espectros de frecuencia:
    a. Transformación Wavelet.
    b. Obtención de espectros de energía.
    c. Obtención de medidas de los estimadores de la entropía máxima.
3. Proceso de identificación de máximos relativos de la entropía máxima.
4. Obtención de los puntos que segmentan al archivo de voz ingresado.

Observaciones

- En este proceso se transforman los archivos WAV en espectros de frecuencia que son segmentados para la obtención de los posibles tokens.

- El proceso de segmentación está basado en la composición de tokens, en el que se observa la obtención de valores máximos con las bajas frecuencias, al momento de obtener los vectores de entropía máxima (ver sección 5.5.2). Esto permite considerar este criterio como la base de la segmentación de un registro de voz.

- El problema principal a resolver ha sido la identificación del criterio de segmentación, el cual requiere la observación exhaustiva de los espectros de frecuencias generados por las distintas combinaciones de vocales y consonantes que forman el diccionario de tokens utilizados.