

Especificación de requisitos software

Índice de contenidos

1. [Introducción](#)

1.1. [Objetivo](#)

- 1.1.1. Propósito del documento
- 1.1.2. Audiencia a la que va dirigido

1.2. [Alcance](#)

- 1.2.1. Identificación del producto mediante un nombre
- 1.2.2. Qué hace y no hace el producto
- 1.2.3. Aplicaciones del software: beneficios, objetivos y metas

1.3. [Definiciones, acrónimos y abreviaturas](#)

1.4. [Referencias](#)

1.5. [Visión general](#)

- 1.5.1. Descripción del contenido del resto del documento
- 1.5.2 Organización del documento

2. [Descripción general](#)

2.1. [Perspectiva del producto](#)

- 2.1.1. Indicar si es un producto independiente o parte de un sistema mayor
- 2.1.2. Interfaces de sistema
- 2.1.3. Limitaciones de memoria
- 2.1.4. Operaciones
 - 2.1.4.1. Modos de operación de los distintos grupos de usuarios
 - 2.1.4.2. Periodos de operaciones interactivas y automáticas
 - 2.1.4.3. Funciones respaldo del procesamiento de datos
 - 2.1.4.4. Operaciones de backup y recuperación
- 2.1.5. Requerimientos para adaptarse a la ubicación
 - 2.1.5.1. Indicar cualquier dato o secuencia de inicialización específico de cualquier lugar, modo de operación.
 - 2.1.5.2. Características que deben ser modificadas para una instalación en particular.

2.2. [Funciones del producto](#)

2.3. [Características de usuario](#)

2.4. [Restricciones](#)

2.5. [Suposiciones y dependencias](#)

2.6. [Requisitos para futuras versiones del sistema](#)

3. [Requisitos específicos](#)

3.1. [Requisitos de interfaz externo](#)

- 3.1.1. Interfaces de usuario
- 3.1.2. Interfaces hardware
- 3.1.3. Interfaces software
- 3.1.4. Interfaces de comunicaciones

3.2. [Requisitos funcionales](#)

- 3.2.1. [Flujos de información](#)
- 3.2.2. [Descripción de procesos](#)
- 3.2.3. [Diccionario de datos](#)

- 3.3. [Requisitos de rendimiento](#)
- 3.4. [Restricciones de diseño](#)
- 3.5. [Atributos de sistemas software](#)
- 3.6. [Otros requisitos](#)

1. Introducción

1.1. Objetivo

1.1.1. Propósito del documento

Este documento recoge la especificación de requisitos así como la toda la documentación correspondiente del análisis a la aplicación. A partir de él estaremos en condiciones de establecer un diseño que se ajuste a los requerimientos aquí expuestos, después de realizar el estudio de reutilización para determinar componentes de los que ya se dispone, que se ajustan a los requisitos y que se puedan incorporar al diseño.

1.1.2. Audiencia a la que va dirigido

Se dirige en primer término al responsable del diseño de la aplicación, en este caso es la misma persona al tratarse de un proyecto personal. A posteriori, es una justificación de las decisiones tomadas al desarrollar este proyecto fin de carrera, cara al tribunal que lo evalúe.

1.2. Alcance

1.2.1. Identificación del producto mediante un nombre

Se adjudica al proyecto el nombre EACP (Etiquetado Automático de Componentes Prosódicos). A partir de este momento cualquier referencia a EACP corresponde al proyecto fin de carrera que nos ocupa.

1.2.2. Qué hace y no hace el producto

El producto genera etiquetas de manera automática que califican componentes prosódicos dentro de una muestra de voz dada. Por tanto, no se trata de una herramienta asistida de etiquetado; ella misma toma las decisiones de acuerdo a las reglas introducidas por el usuario que definen los componentes prosódicos.

1.2.3. Aplicaciones del software: beneficios, objetivos y metas

El objetivo es disponer de una herramienta que permita determinar reglas, referentes a una serie de parámetros, que permitan definir componentes prosódicos para el castellano. Es una herramienta de ayuda a la investigación, la definición de las reglas y análisis de los resultados corre por cuenta del usuario. El objetivo último es mediante dichas reglas determinar exactamente acentos, y límites prosódicos en el habla.

1.3. Definiciones, acrónimos y abreviaturas

Término	Significado
---------	-------------

EACP	Etiquetado Automático de Componentes Prosódicos
DFD	Diagrama de Flujo de Datos, incluye tanto datos como control. Equivaldría a la unión de DFD y CFD (<i>Control Flow Diagram</i>) en [HAT87]
PSPEC	<i>Process Specification</i> [HAT87]
CSPEC	<i>Control Specification</i> [HAT87]
VOD	<i>Vowel Onset Detection</i>

1.4. Referencias

- [IEEE93] IEEE Std 830-1993 (Revision of IEEE Std 830-1984). Software Engineering Standards Committee of the IEEE Computer Society.

The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394 USA.

- [HAT87] Strategies for real-time system specification. Derek J. Hatley, Imtiaz A. Pirbhai, 1987.

Dorset House Publishing Co., Inc., 353 West 12th Street, New York, NY 10014 USA

1.5. Visión general

1.5.1. Descripción del contenido del resto del documento

Se divide en dos partes. Por una parte la sección 2 describe los factores generales que afectan al producto y sus requerimientos. No se indican los requerimientos específicos si no que se establece un marco de trabajo para dichos requerimientos que son establecidos con toda precisión en la sección 3. En esta se realiza un análisis de la aplicación siguiendo la metodología basada en DFD's [HAT87]. Incluye diagramas de flujos, especificación de procesos y diccionario de datos.

1.5.2 Organización del documento

Esta especificaciones de requisitos software siguen las recomendaciones del estándar [IEEE93] en cuanto a la organización de los distintos apartados así como el contenido de cada uno de ellos.

2. Descripción general

2.1. Perspectiva del producto

2.1.1. Indicar si es un producto independiente o parte de un sistema mayor

El sistema EACP es un producto independiente, aunque para disponer de una de sus entradas es posible el uso de herramientas externas. Estos datos de entrada se refieren a la segmentación en fonemas de la muestra de voz a analizar (ver el diagrama de contexto en la sección 3.2.1).

2.1.2. Interfaces de sistema

1. El sistema debe interactuar correctamente con el sistema operativo UNIX sobre el que se desarrollo e implanta
2. Debe generar una salida válida para la herramienta de generación de gráficas GNUPLOT.

2.1.3. Limitaciones de memoria

No se establecen limitaciones en cuanto a la cantidad de memoria, tanto secundaria como primaria que el sistema deba utilizar.

2.1.4. Operaciones

2.1.4.1. Modos de operación de los distintos grupos de usuarios

El tipo de usuarios es único. Se establece dos modos de operación:

1. De tipo interactivo y gráfico para la herramienta de configuración.
2. El resto de operaciones se realizan automáticamente con la invocación del programa.

2.1.4.2. Funciones respaldo del procesamiento de datos

No se utilizan, cualquier tratamiento ulterior de los datos generados corre por cuenta del usuario.

2.1.4.3. Operaciones de backup y recuperación

No se utilizan

2.1.5. Requerimientos para adaptarse a la ubicación

2.1.5.1. Datos o secuencias de inicialización específicos de cualquier lugar, modo de operación.

El sistema de ficheros debe soportar nombre de al menos siete caracteres y al menos una extensión.

Los requerimientos de memoria y disco del programa se estiman relativamente modesto a fecha actual 1998. En plataformas antiguas se debe verificar.

2.1.5.2. Características que deben ser modificadas para una instalación en particular.

Por defecto se utiliza la herramienta makefile y el compilador C gcc. Si no se dispone de alguno de ellos, corre a cuenta del instalador la compilación correcta del programa.

2.2. Funciones del producto

Input	Muestra de voz, segmentación en fonemas, reglas de decisión
Output	Etiqueta prosódico
Procesamiento	<ul style="list-style-type: none"> • Análisis de parámetros prosódicos <ul style="list-style-type: none"> ◦ Zscore ◦ Contorno de <i>pitch</i> ◦ Fuerza de las vocales ◦ Registros • Decisión basada en las reglas y una base de hechos disponible

2.3. Características de usuario

La aplicación va dirigida a personal de investigación en análisis de voz y más concretamente en prosodia. Por tanto, se encamina a un usuario con alto nivel de conocimientos y de un grado de experiencia alto para poder extraer toda la utilidad a EACP.

2.4. Restricciones

El sistema debe basarse en la aplicación original del IPO, en su esquema, parámetros tratados, etc. Recordar que estamos ante un proyecto de reutilización.

2.5. Suposiciones y dependencias

Los requerimientos se asumen para un sistema UNIX, con la suficiente potencia (no se supone excesiva) para poder obtener los resultados en un tiempo razonable. No existen requerimientos de tiempo de respuesta, limitaciones de memoria, etc.

2.6. Requisitos para futuras versiones del sistema

Se omite para futuras versiones la construcción de un interfaz gráfico completo que permite realizar trabajo interactivo con la aplicación.

De la misma manera, se sugiere la utilización de un sistema experto u otras técnicas apropiadas de inteligencia artificial para la toma de decisiones.

3. Requisitos específicos

3.1. Requerisitos de interfaz externo

3.1.1. Intefaces de usuario

3.1.2. Interfaces hardware

3.1.3. Interfaces software

3.1.4. Interfaces de comunicaciones

3.2. Requisitos funcionales

Para el desarrollo de esta sección se utiliza básicamente el método indicado por Hatley [HAT87]. Se realizan ciertas modificaciones que se ajustan a lo estudiado a lo largo de este segundo ciclo de Ingeniería en Informática. Son:

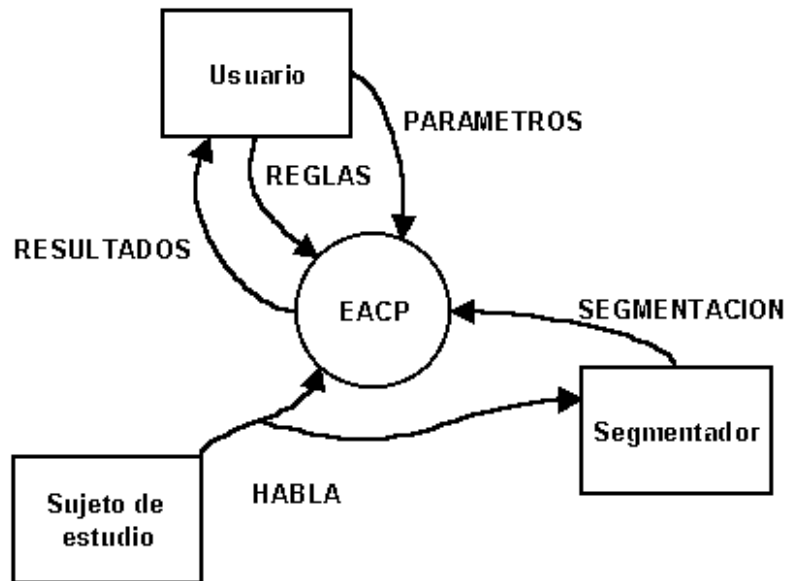
- Dado la escasa incidencia del control en la aplicación los DFD's (*Data Flow Diagram*) y los CFD's (*Control Flow Diagram*) se funden en un único diagrama que a partir de este momento se denota como DFD. Consecuentemente PSPEC (*Process SPECifications*) y CSPEC (*Control SPECifications*) se integran en un único apartado al que se nombrará como PSPEC.
- Las especificaciones de procesos (PSPEC) no se realizaran en Inglés Estructurado, tal como aparece en la referecia dada. Se utilizara pseudocódigo en castellano intentando en todo momento que sea lo más estructurado posible pero sin definir explícitamente una sintaxis.
- En caso de posible ambigüedad se utiliza la notación punteada para cualificar datos que formen parte de otros de nivel superior.

En ciertas partes de este análisis se puede argumentar que el grado de detalle es excesivo. No

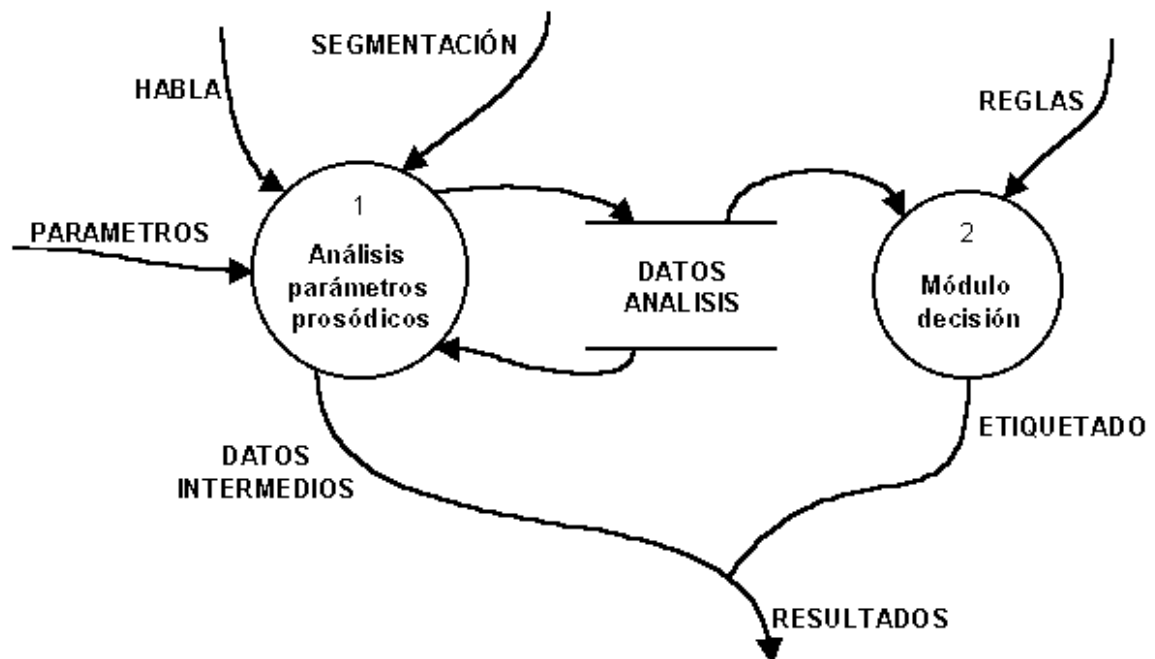
obstante, se consideró adecuado, ya que buena parte se trata de cuestiones de teoría de la señal, que por complejas deben estar correctamente especificadas. Además, en un aspecto exclusivamente de ingeniería se puede argumentar que un programador no debe conocer estos detalles, si no que deben ser concretados exactamente por el responsable de especificar los requisitos funcionales, ya que su implementación es crítica y no ha lugar a ningún tipo de creatividad. La correcta implementación de estos requisitos requiere de una definición exhaustiva de como realizar ciertas tareas como puede ser el filtro sobre una señal.

3.2.1. Flujos de información

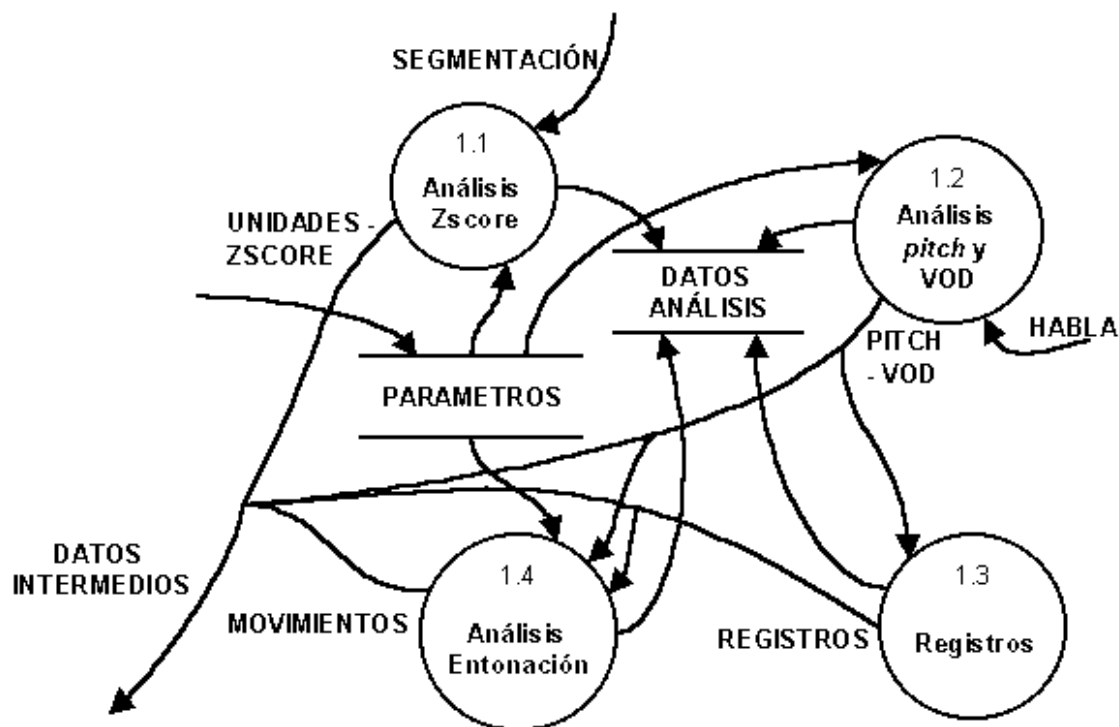
Diagrama de contexto



DFD 0

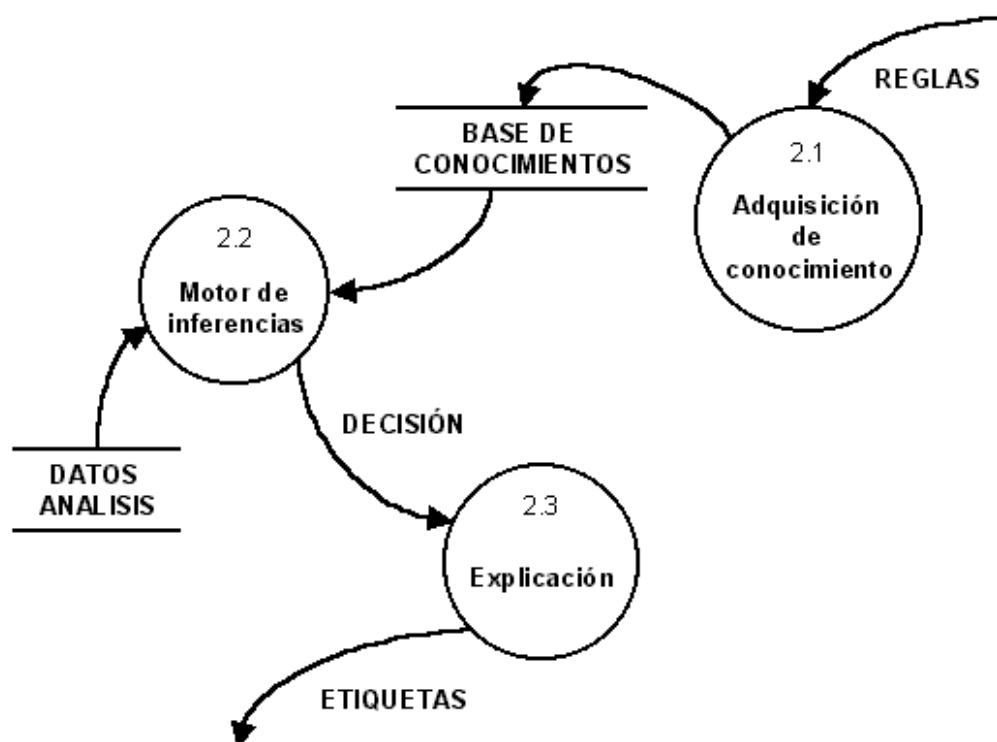


DFD 1

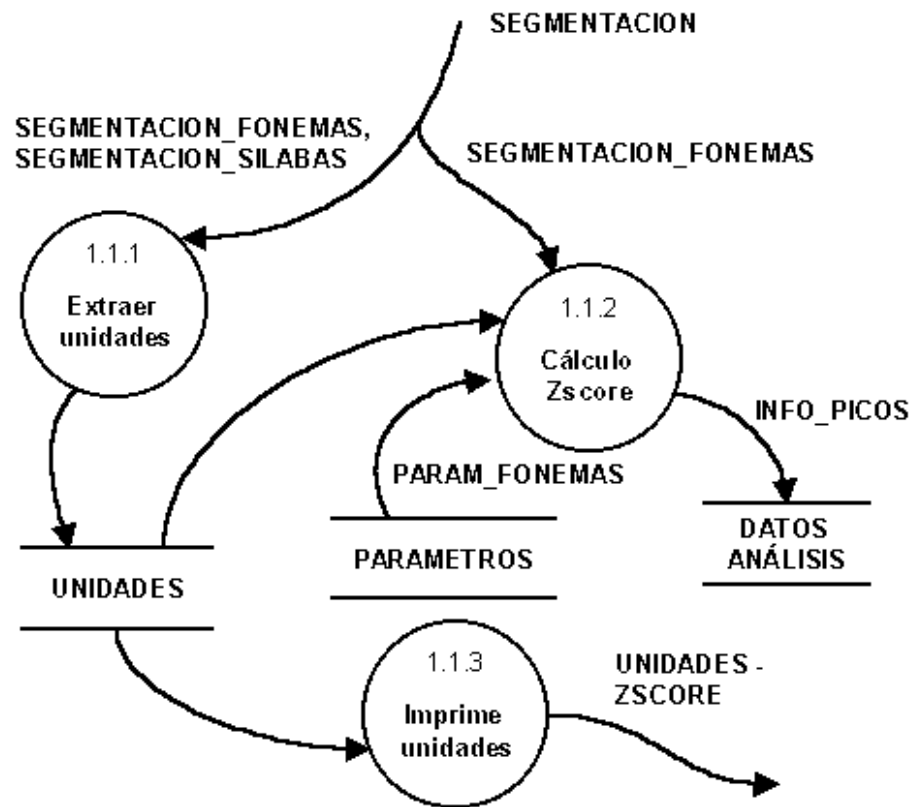


Destacar la presencia del almacén parámetros. Dada la enorme cantidad de ellos que deberán existir en la aplicación y la necesidad de hacer la herramienta flexible (dado su carácter de investigación) mediante la posible variación de los valores ligados a cada uno de los datos presentes en dicho almacén se considera oportuno que figure como tal en este análisis.

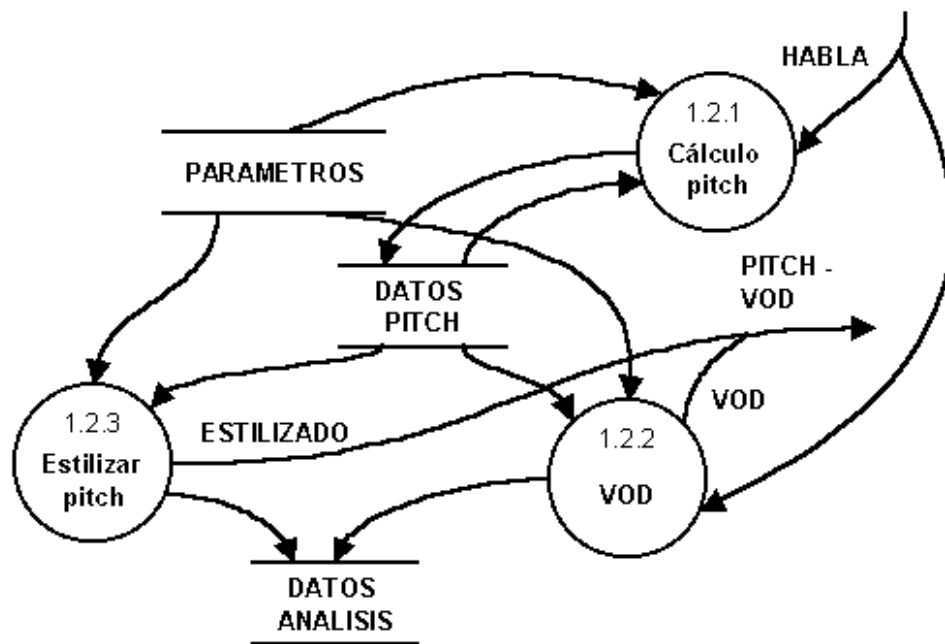
DFD 2



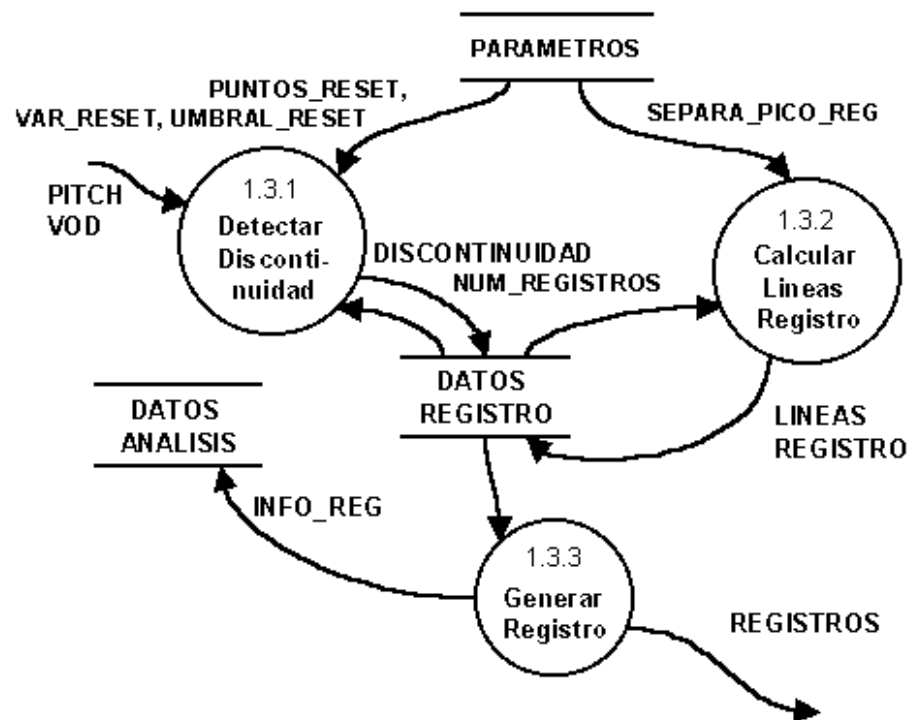
DFD 1.1



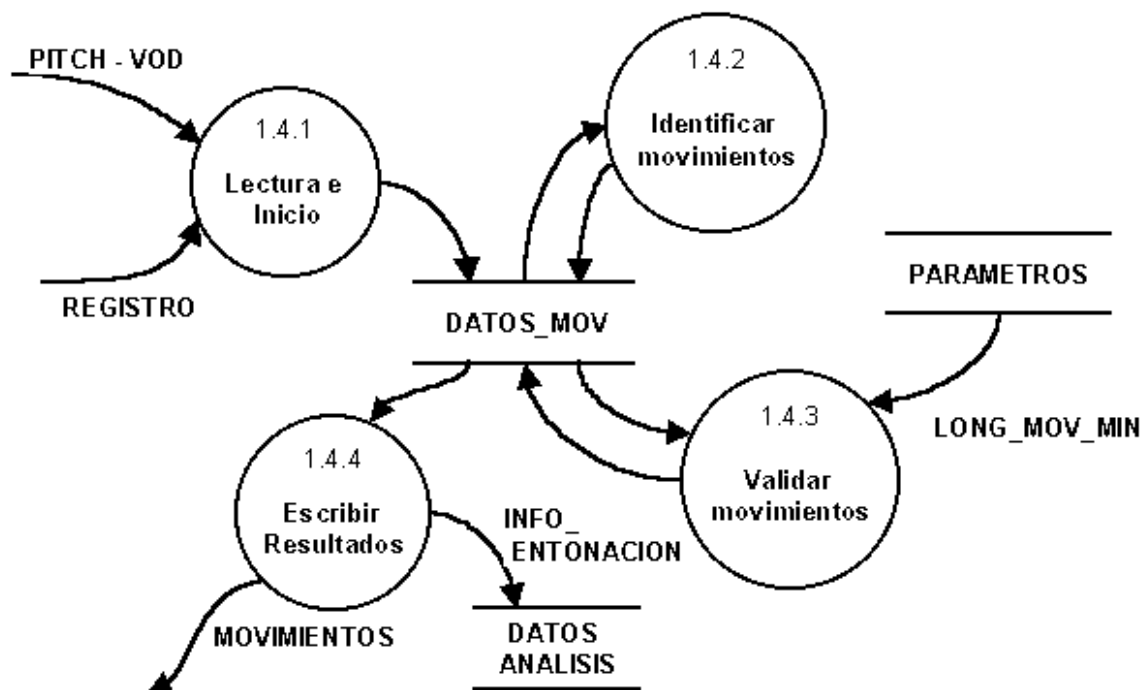
DFD 1.2



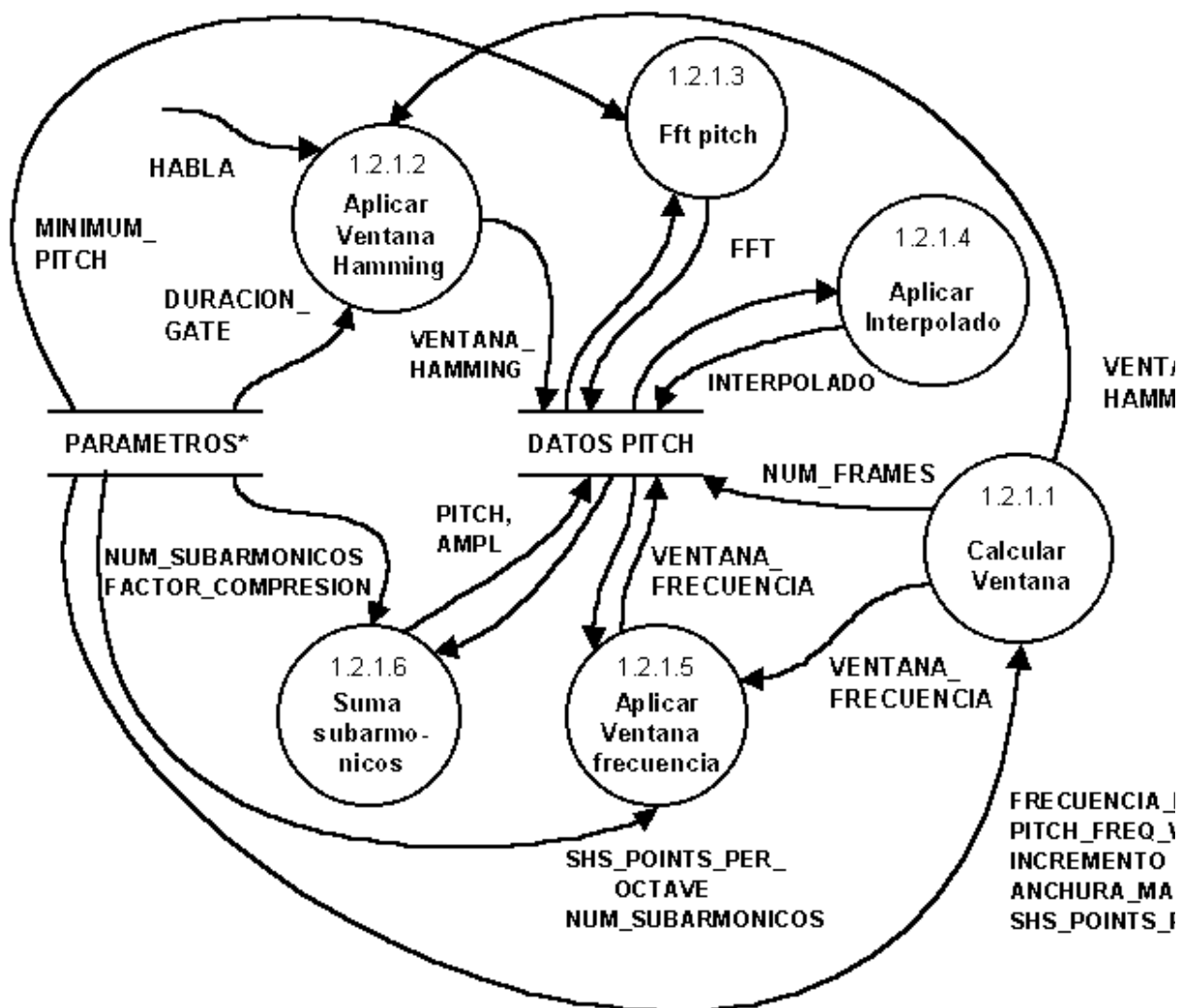
DFD 1.3



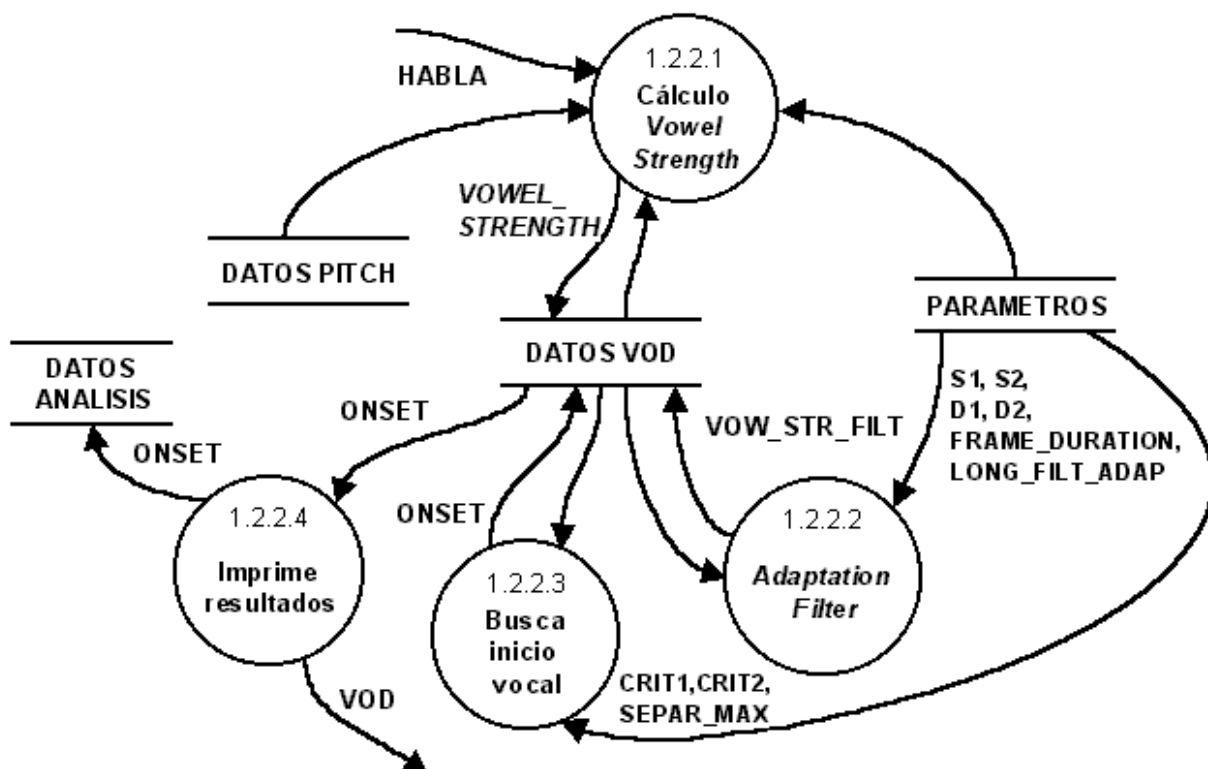
DFD 1.4



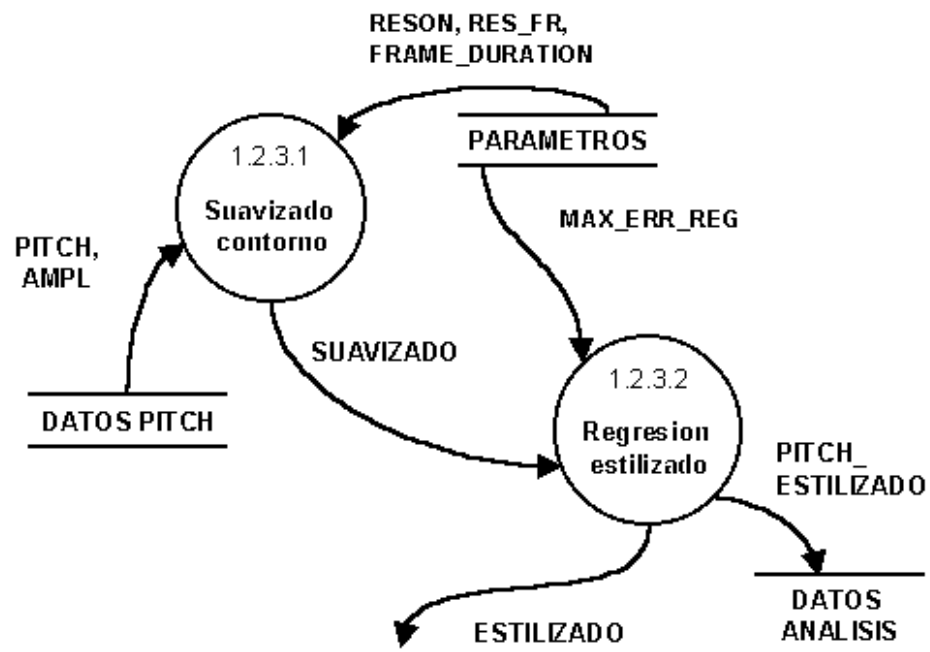
DFD 1.2.1



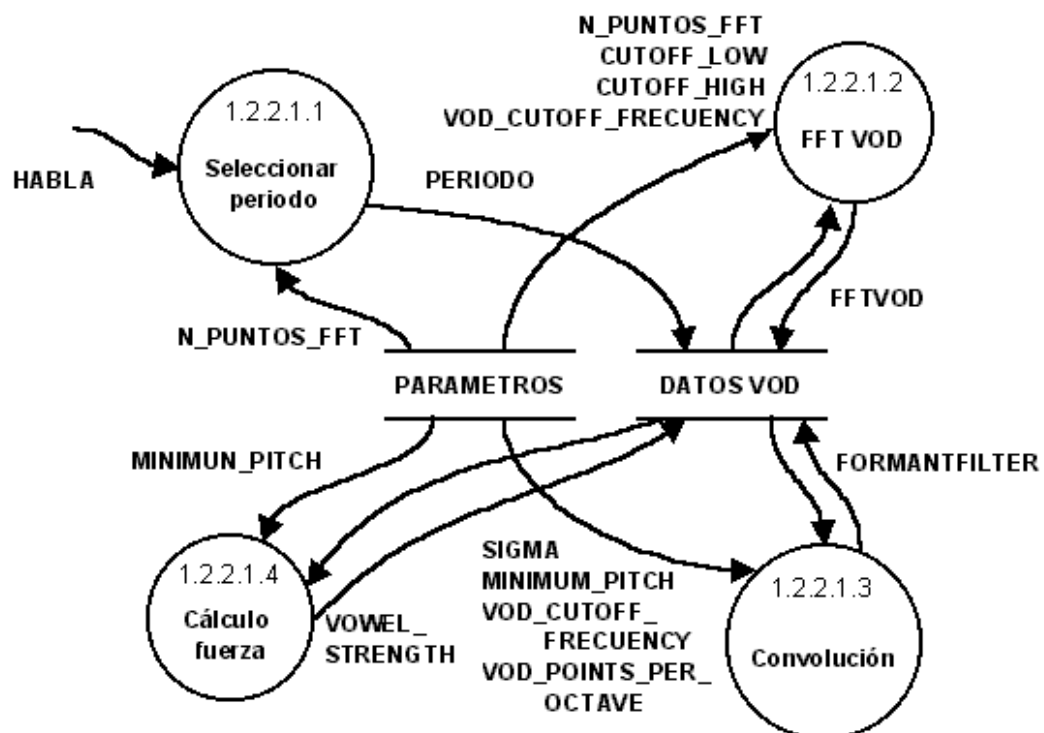
DFD 1.2.2



DFD 1.2.3



DFD 1.2.2.1



3.2.2. Descripción de procesos

PSPEC 1.1; Análisis Zscore.

Orden de ejecución de tareas:

1. Extraer grupos fónicos
2. Cálculo Zscore
3. Imprime unidades

PSPEC 1.1.1; Extraer unidades.

Omitir silencio inicial si es que existe.

Para cada una de las unidades indicadas en SEGMENTACION_SILABAS

UNIDADES.puntoInicio = comienzo primer fonema

UNIDADES.duracion = suma de los fonemas que lo componen y pausas intermedias
(donde la información sobre los fonemas se encuentra en

SEGMENTACION_FONEMAS)

Fin_para

PSPEC 1.1.2; Cálculo Zscore.

Inicializar INFO_PICOS

Calcular *Global speech rate* según la siguiente fórmula:

$$\frac{1}{n} \sum_{i=1}^n \frac{\log(dur[i]) - \mu[i]}{\sigma[i]}$$

$dur[i]$: duración del fonema i

$\mu[i]$: duración media para el fonema i.

n : número de fonemas del discurso

$\sigma[i]$: desviación típica para el fonema i.

Para cada unidad j en UNIDADES

Para cada fonema i en unidad j

Obtener k de:

$$D[j] = \mu[i] + k[j] \sigma[i]$$

$D[j]$: duración de la unidad

$K[j]$: Zscore

Fin para_cada

Fin para_cada

Buscar picos en k

Guardar pico en INFO_PICOS.PEAK

Guardar valor de k para dicho pico en INFO_PICOS.HEIGHT

Filtrar k

$$Zs[i] = (Z[i-1] + 2 * Z[i] + Z[i+1]) / 4$$

$Zs[i]$: valor filtrado de k en la unidad i

$Z[i]$: valor de k en la unidad i

Buscar picos en Zs.

Buscar correspondencia con k.

Guardar pico en INFO_PICOS.PICO

Guardar longitud en INFO_PICOS.LENGTH

PSPEC 1.1.3.; Imprime unidades.

Guardar en UNIDADES_ZSCORE datos de la unidad calculada:

punto de inicio, duración, valor zscore, valor zscore suavizado.

PSPEC 1.2; Análisis *pitch* y VOD.

Orden de ejecución de tareas:

1. Calcular pitch
2. VOD
3. Estilizar pitch

PSPEC 1.2.1; Cálculo pitch.

Calcular ventana

Desde FRAME=1 hasta NUM_FRAMES

Aplicar Ventana Hamming

Fft pitch

Aplicar interpolado

Aplicar Ventana Frecuencia

Suma subarmónicos

Zona vocálica

Fin_desde

PSPEC 1.2.1.1; Calcular ventana.

Calcular anchura en muestras (N) a partir de DURATION_GATE y FRECUENCIA_MUESTREO.

Desde i=1 hasta ANCHURA_GATE

$$VENTANA_HAMMING[i] = 0.54 - 0.46 \cos\left(\frac{2\pi i}{N-1}\right)$$

Fin_desde

s = SHS_POINTS_PER_OCTAVE * log2(PITCH_FREQ_WINDOW/MINIMUM_PITCH)

Desde i=1 hasta ANCHURA_MAX

$$VENTANA_FRECUENCIA[i] = 0.5 + \frac{1}{\pi} \operatorname{atan}\left(\frac{INCREMENTO * (i-1)}{SHS_POINTS_PER_OCTAVE}\right)$$

Fin_desde

$$NUM_FRAMES = 1 + \frac{NUM_MUESTRAS}{DURACION_FRAME * FRECUENCIA_MUESTREO + 0.5}$$

PSPEC 1.2.1.2; Aplicar Ventana Hamming.

k = número de muestras correspondientes a DURATION_GATE segundos.

Desde i=1 hasta k

$$VENTANA_HAMMING_APLICADA[i] = \begin{matrix} 0 & \text{si} \\ & HABLA[i] \leq 0 \\ VENTANA_HAMMING[i] * HABLA[i] & \text{si} \\ & HABLA[i] > 0 \end{matrix}$$

Fin_desde

PSPEC 1.2.1.3; Fft pitch.

Realizar *Fast Fourier Transform* sobre VENTANA_HAMMING

Realzar picos

Ignorar picos < MINIMUN_PITCH
 Ignorar picos alejados de un máximo local en más de 2 puntos
 Realizar un suavizado de Hamming y guardar en FFT

PSPEC 1.2.1.4; Aplicar interpolado.

Realizar regresión cúbica mediante splines
 Interpolador valores obteniendo el resultado en escala logarítmica.
 Guardar resultado en INTERPOLADO

PSPEC 1.2.1.5; Aplicar ventana en frecuencia.

Desde i=1 hasta SHS_POINTS_PER_OCTAVA * NUM_OCTAVAS

$$\begin{aligned} & \text{VENTANA_FRECUENCIA_APLICADA}[i] = \begin{cases} 0 & \text{si } \text{INTERPOLADO}[i] \leq 0 \\ \text{VENTANA_FRECUENCIA}[i] * \text{INTERPOLADO}[i] & \text{si } \text{INTERPOLADO}[i] > 0 \end{cases} \end{aligned}$$

Fin_desde

PSPEC 1.2.1.6; Suma subarmónicos.

Determinar PITCH a partir de VENTANA_FRECUENCIA_APLICADA según la fórmula:

$$PITCH(f) = \sum_{i=1}^N (h_n(P(nf)))$$

n: FACTOR_COMPRESION

h_n : Secuencia decreciente, donde el armónico más alto contribuye menos que el más bajo

N: SUM_SUBARMONICOS

P: VENTANA_FRECUENCIA_APLICADA

Además guardar en AMPL los valores de amplitud correspondientes.

PSPEC 1.2.2; VOD

- 1°. Cálculo *vowel strength*
- 2°. *Adaptation filter*
- 3°. Busca inicio vocal
- 4°. Imprime resultados

PSPEC 1.2.2.1; Cálculo *vowel strength*

Para FRAME=1 hasta NUM_FRAMES
 Seleccionar periodo
 FFT VOD
 Convolución
 Cálculo fuerza
 Fin_para

PSPEC 1.2.2.2; Adaptation filter

LONG_ADAPT=(D1+D2+2*S1+2*S2) / FRAME_DURATION

t=-D1-2*S1

Para i=1 hasta LONG_ADAPT

$$\text{ADAPTFILTER}[i] = \frac{1}{\sqrt{2\pi}} \left(\frac{1}{S1} \exp \left(-0.5 \left(\frac{t+D1}{S1} \right)^2 \right) - \frac{1}{S2} \exp \left(-0.5 \left(\frac{t-D2}{S2} \right)^2 \right) \right)$$

t=t+FRAME_DURATION

Fin_para

Para frame=1 hasta NUM_FRAMES

del = 0.55 / DURACION_FRAME

Si (frame >= del)

VOW_STR_FILT[frame-del]=0

Para k=0 hasta k<LONG_FILTRO_ADAPT

Si (frame-k)>=0

VOW_STR_FILT[frame-del]+=ADAPT_FILTER[k]*

VOWEL_STRENGTH[frame-k]

Fin_si

Fin_para

Fin_si

Fin_para_cada

PSPEC 1.2.2.3; Busca inicio vocal

NRONSET = 0

Buscar un máximo local en VOW_STR_FILT

Buscar un pico candidato

Comprobar que no hay otro pico CRIT1 segundos antes

Comprobar que no hay otro pico CRIT1 segundos despues

Comprobar que tiene un valor al menos CRIT2 factor respecto al máximo total

Si cumple estas condiciones en el frame(fr) se da una discontinuidad.

NRONSET = NRONSET + 1

Fin_si

Guardar la altura del pico encontrado

A partir de fr buscar el fonema correspondiente

Si está separado más de 0.1

seg ONSET[fr]='?'

Si no asignar ONSET[fr]=fonema

Fin_si

PSPEC 1.2.2.4; Imprime resultados

Trasladar ONSET a DATOS_ANALISIS

Imprimir ONSET en VOD

PSPEC 1.2.2.1.1; Seleccionar periodo

Realizar regresión cúbica mediante splines sobre el fragmento de HABLA correspondiente a FRAME

Interpolar N_PUNTOS_FFT valores .

Guardar resultado en INTERPOLA_VOD

PSPEC 1.2.2.1.2; FFT VOD

Realizar *Fast Fourier Transform* sobre INTERPOLA_VOD, resultado en FV.

Filtrar el resultado

i=1

fr=PITCH[FRAME]

Para fr<2*CUTOOF_LOW

$$\text{FFTVOD}[i] = 0,5 * \left(1 - \cos\left(\frac{\pi}{2}\right) \frac{\text{fr}}{\text{CUTOFF_LOW}} \right) * \text{FV}[i]$$

fr=fr+PITCH[FRAME]

i=i+1

Fin_para

Para fr<CUTOFF_HIGH+CUTOFF_HIGH-VOD_CUTOFF_FREQUENCY

FFTVOD[i]=FV[i]

fr=fr+PITCH[FRAME]

i=i+1

Fin_para

Para fr<VOD_CUTOFF_FREQUENCY

$$\text{FFTVOD}[i] = 0,5 * \left(1 - \cos\left(\frac{\pi}{2}\right) \frac{\text{VOD_CUTOFF_FREQUENCY} - \text{fr}}{\text{VOD_CUTOFF_FREQUENCY} - \text{CUTOFF_HIGH}} \right)$$

fr=fr+PITCH[FRAME]

i=i+1

Fin_para

PSPEC 1.2.2.1.3; Convolución

Calcular una ventana tipo gaussiana

Para i=0 hasta VOD_POINTS_PER_OCTAVE

$$\text{GAUSS}[i] = \exp \left(-0,5 * \left(\frac{i}{\text{SIGMA} * \text{VOD_POINTS_PER_OCTAVE}} \right)^2 \right)$$

Fin_para

Calcular el número de puntos en la escala logarítmica (k)

$$k = \left(\text{VOD_POINTS_PER_OCTAVE} * \log \left(\frac{\text{VOD_CUTOFF_FREQUENCY}}{\log(2) * \text{MINIMUM_PITCH}} \right) \right) + 0.5$$

Aplicar la ventana a cada línea espectral, observando la escala logarítmica .

PSPEC 1.2.2.1.4; Cálculo fuerza

fr = MINIMUM_PITCH

$$\text{dfr} = 2^{\frac{1}{\text{VOD_POINTS_PER_OCTAVE}}}$$

t = FORMANT_FILTER[0]

i=1

Repetir

s = FORMANT_FILTER[i] - FORMANT_FILTER[i-1]

Si (s>0) t=t+s

Fin_si

i=i+1


```

fr=fr*dfr
Hasta (fr<=VOD_CUTOFF_FRECUENCY)
VOWEL_STRENGTH[FRAME]=t/25000
VOWEL_STRENGTH[FRAME]=VOWEL_STRENGTH[FRAME]*
(AMPL[FRAME]/40000)

```

PSPEC 1.2.3.1; Suavizado contorno

```

A = RESON2
B = -2*RESON*cos(2*PI*FRAME_DURATION*RES_FR)
C = 1+A+B
D = -(A+A+B) / C
INTD = D+0,5
Para FRAME=1 hasta NUM_FRAME
    AMPSUAV[FRAME] = C*log(AMPL[FRAME]) - B*AMPSUAV[FRAME-1] -
                    A*AMPSUAV[FRAME-2]
    PITCHSUAV[FRAME] = C*log(AMPL[FRAME])*PITCH[FRAME] -
                    B*AMPLSUAV[FRAME-1]*PITCHSUAV[FRAME-1] -
                    A * AMPSUAV [FRAME - 2] * PITCHSUAV [FRAME - 2]
                    AMPSUAV [FRAME]
Fin_para
Para FRAME=1 hasta NUM_FRAME
    SUAVIZADO.PITCH[FRAME] = PITCHSUAV[FRAME+INTD]
    SUAVIZADO.AMPL[FRAME] = exp(AMPLSUAV[FRAME+INTD])
Fin_para

```

PSPEC 1.2.3.2; Regresión estilizado

```

Para FRAME con SUAVIZADO.PITCH[FRAME]=0
    PITCH_ESTILIZADO[FRAME]=0
Fin_para
Para FRAME con SUAVIZADO.PITCH[FRAME]! =0
    Tomar 10 puntos y calcular la línea de regresión y el error de regresión
    (ERR_REGR)
    Si ERR_REGR<MAX_ERR_REGR
        Para esos 10 puntos asignar a PITCH_ESTILIZADO los puntos
        correspondientes de la línea de regresión
    Si no
        Elegir de los 5 primeros puntos aquel con el que se obtenga menor
        ERR_REGR, sea P
        Asignar para los P primeros puntos los puntos correspondientes de la línea de
        regresión a PITCH_ESTILIZADO
        Para los 10-P puntos restantes calcular la línea de regresión y asignarla a
        PITCH_ESTILIZADO
    Fin_si
Fin_para
Escribir PITCH_ESTILIZADO en ESTILIZADO

```

PSPEC 1.3; Registros

Orden de ejecución de tareas:

1. Detectar Discontinuidad

2. Calcular líneas Registro
3. Generar Registro

PSPEC 1.3.1; Detectar Discontinuidad

Calcular en número total de muestras de PITCH_VOD (N)

Para $i = \text{PUNTOS_RESET}$ hasta $\text{PUNTOS_RESET} + \text{VAR_RESET}$

$M = N - i$

Para $j = 0$ hasta M

Calcular la línea de regresión para i valores de ESTILIZADO a partir de j :
del tipo $\text{ESTILIZADO} = Ax + B$

Guardar A , B

Fin_para

Calcular el final de las discontinuidades (DISCONTINUIDAD.FINAL) para los M valores disponibles¹

Fin_para

Calcular el comienzo de las discontinuidades (DISCONTINUIDAD.INICIO)²

Calcular en NUM_REGISTROS el número total de registros obtenidos

¹ Se toman como inicio del registro el final de las zonas de pendiente positiva de duración mayor de UMBRAL_RESET milisegundos

² Se toma como final del registro el comienzo de una zona de pendiente positiva

PSPEC 1.3.2; Calcular líneas Registro

Para $i = 1$ hasta NUM_REGISTROS

Calcular la línea de regresión general ($A_GEN \cdot x + B_GEN$) entre DISCONTINUIDAD.INICIO y DISCONTINUIDAD.FINAL

Seleccionar picos por encima de línea de regresión general:

- Separados al menos en SEPARA_PICO_REG
- Eliminar aquellos demasiado extremos

Calcular línea de regresión positiva ($A_POS \cdot x + B_POS$)

Seleccionar picos por debajo de línea de regresión general:

- Separados al menos en SEPARA_PICO_REG
- Eliminar aquellos demasiado extremos

Calcular línea de regresión negativa ($A_NEG \cdot x + B_NEG$)

Guardar en LINEAS_REGISTRO: A_GEN , B_GEN , A_POS , B_POS , A_NEG , B_NEG)

Fin_para

PSPEC 1.3.3; Generar Registro

Imprimir la información de LINEAS_REGISTROS en REGISTROS, junto con la escala de tiempo

Para cada unidad j en UNIDADES

Si existe una discontinuidad en j

Indicarlo en INFO_REG.RESET

Guardar en INFO_REG.INCLINACION la inclinación del siguiente registro

Guardar en INFO_REG.ALTURA la altura de la discontinuidad

si no

Indicarlo en INFO_REG.RESET

Fin_si

Fin_para

PSPEC 1.4; Análisis entonación

Orden de ejecución de tareas:

1. Lectura e inicio
2. Identificar movimientos
3. Validar movimientos
4. Escribir resultados

PSPEC 1.4.1; Lectura e inicio

Determinar todos los movimientos según PITCH_VOD:

Tener en cuenta que un movimiento está formado por dos puntos consecutivos sobre el contorno estilizado

Agrupar movimientos consecutivos similares o planos según los valores de los registros en REGISTRO

Guardar en DATOS_MOV.ORIGIN

PSPEC 1.4.2; Identificar movimientos

Clasificar los movimientos en:

- Subidas
- Bajadas
- Planos

Guardar en DATOS_MOV.INTERMEDIO

PSPEC 1.4.3; Validar movimientos

Validar movimientos eliminando aquellos excesivamente cortos ($< \text{LONG_MOV_MIN}$)

Guardar en DATOS_MOV.FINAL

PSPEC 1.4.4; Escribir resultados

Para cada unidad j en UNIDADES

Indicar en INFO_ENTONACION.PROMINENCIA si es comienzo de movimiento

Indicar en INFO_ENTONACION.TIPO la categoría a la que pertenece el movimiento tratado

Fin_para

Escribir DATOS_MOV en MOVIMIENTOS

PSPEC 2.1; Adquisición de conocimiento

Leer REGLAS

Análisis sintáctico de REGLAS

Si error

Notificar y finalizar la decisión

Si no

Análisis léxico, extraer ETIQUETA_REGLA, MAGNITUD_REGLA, VALOR_REGLA

Introducir resultado en BASE DE CONOCIMIENTOS

Fin_si

PSPEC 2.2; Motor de inferencias

Buscar DATOS-ANALISIS que satisfagan las reglas de BASE DE CONOCIMIENTOS *

Para cada correspondencia encontrada

Introducir ETIQUETA, MAGNITUD y POSICION en DECISION

Fin_para

* Es en este punto donde se pueden aplicar distintas técnicas, desde un simple reconocimiento de patrones hasta técnicas de IA: sistemas expertos, etc.

PSPEC 2.3; Explicación

Imprimir DECISIÓN en ETIQUETAS con el formato oportuno.

3.2.3. Diccionario de datos

(En parámetros)

PUNTOS_RESET: número de puntos en un grupo para la detección de discontinuidades en el cálculo de los registros

VAR_RESET: variación en PUNTOS_RESET

ETIQUETA_REGLA, MAGNITUD_REGLA, VALOR_REGLA, una o varias por cada elemento de BASE DE CONOCIMIENTOS.

3.3. Requisitos de rendimiento

Únicamente, recalcar que la aplicación debe funcionar sobre un sistema UNIX convencional.

3.4. Restricciones de diseño

Toda las salidas del producto deben soportar dos tipos

1. Modo texto, con los datos convenientemente organizados para poder ser utilizado con herramientas genéricas
2. Script para la herramienta GNUPLOT, es decir, se debe generar un salida texto que puede interpretar el producto mencionado para la construcción de gráficas.

3.5. Atributos de sistemas software

3.6. Otros requisitos

