



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



MODELADO LINGÜÍSTICO-PROSÓDICO PARA UN SISTEMA CONVERSOR TEXTO A VOZ MEDIANTE CONCATENACIÓN DE DEMISÍLABAS

TESIS PARA OPTAR EL TÍTULO DE INGENIERO INFORMÁTICO

PRESENTADA POR

ELÍ RÓNAL SEGURA SALAS

Lima – Perú
2004

Resumen

El objeto principal de esta Tesis ha sido el estudio del procesamiento lingüístico y prosódico en un sistema conversor de texto a voz basado en concatenación de unidades para un hablante limeño del castellano.

Basado en la información proporcionada por el estudio antes mencionado, se realizó el análisis, diseño e implementación de los módulos del procesamiento lingüístico-prosódico de un sistema conversor texto a voz, así como también, se hicieron los cambios necesarios en los módulos del sintetizador de voz, para aprovechar la información prosódica (duración segmental y contorno de frecuencia fundamental) proveniente de esos módulos.

Para cumplir con la meta de producir voz sintetizada lo más natural posible, el proceso de conversión de texto a voz a sido dividido en tres fases principales: la primera que consiste en obtener un conjunto de palabras que sean lo más independientemente posibles en cuanto a la prosodia (entonación y ritmo) que será la unidad de trabajo del sistema, a este conjunto de palabras la hemos definido como "frase", la cual está dividida del resto del texto por algunos signos de puntuación llamados separadores de frase (punto, punto y coma, dos puntos, etc.).

La segunda fase, es la encargada de generar la información prosódica que se usará en la producción de la voz de alta calidad, para la cual se divide en varios bloques, empezando por separar la frase en unidades más pequeñas que en el sistema es tratado como una lista doblemente enlazada de palabras, se sigue quitando la variabilidad de palabras que hubiera en la frase, pasando los elementos numéricos, fechas, horas, monedas, abreviaturas y siglas a sus equivalencias en texto; la frase es luego silabeada y acentuada fonéticamente, usando un conjunto de reglas. En un tercer paso de esta fase, se lleva a cabo la categorización gramatical de las palabras, para lo cual se hace uso de un conjunto de diccionarios de palabras consideradas como excepciones gramaticales (preposiciones, conjunciones, etc.), y de terminaciones o desinencias, y finalmente, un conjunto de reglas de

desambiguación gramatical. Luego, se pasa a insertar, si es posible, pausas que no están marcadas ortográficamente pero que se producen en la lectura natural debido a la necesidad del hablante de recuperar aliento ante una secuencia larga de habla que no tiene signos ortográficos entre ellos. El siguiente paso consiste en llevar los grafemas del texto a su equivalente en unidades básicas en nuestro caso Demisílabas, para lo cual se tienen en cuenta un conjunto de reglas de conversión, primero de grafemas a alófonos y luego de estos a unidades básicas. El último paso de esta fase, es la generación de los parámetros prosódicos, la cual se compone de dos tareas, una por cada parámetro: duración y contorno de frecuencia fundamental (entonación).

La tercera y última fase, es el llamado sintetizador de voz, que es la encargada de obtener los parámetros de predicción lineal, correspondientes a la secuencia de unidades básicas que se quiere sintetizar, con las cuales se reconstruye la señal de voz correspondiente a cada unidad (Demisílaba), y luego de ajustar el número de sus tramas a la duración calculada para cada una, pasa a generar la voz imponiendo los valores de frecuencia fundamental calculados en la fase anterior.

El sistema consta de dos programas ejecutables principales y una librería (DLL). El primer programa ejecutable fue desarrollado usando Borland Delphi 5.0 y permite hacer consultas y mantenimiento a la base de datos del sistema, así como hacer consultas sobre el resultado del procesamiento de una frase (Contorno de frecuencia fundamental, duración asignada a sus demisílabas, categorización gramatical, etc.). El segundo es un programa que se ejecuta mediante la línea de comandos (antes D.O.S), y sólo se puede mandar leer un texto. La librería fue desarrollada usando Visual C++ 6.0, y se encarga de realizar todo el trabajo del procesamiento lingüístico-prosódico, y el análisis y sintetización de voz.

Dedicatoria

A mis padres Cresencio y Bertha, por su cariño, comprensión, apoyo incondicional y por ser una fuente de inspiración para ser una mejor persona cada día.

A mi esposa Rita por su comprensión y compañía.

Al pequeño Ronald por ser la motivación más importante para seguir adelante.



Agradecimientos

Al profesor Andrés Flores por su colaboración y aportes en el desarrollo de la tesis.

A los amigos y compañeros que me apoyaron en la ejecución de las pruebas de audición del sistema.



Índice

1	Introducción	1
1.1	Definición del Dominio del Problema	1
1.2	Aplicaciones de los Conversores Texto a Voz	1
2	Descripción del Desarrollo del Sistema Conversor Texto-Voz (CTVlector)	3
2.1	Generalidades	3
2.1.1	Proceso de Lectura de una Persona	4
2.2	Generación Sintética de la Voz	6
2.2.1	Procesado Lingüístico-Prosódico	7
2.2.1.1	Análisis de Estructura:	8
2.2.1.2	Normalizado	9
2.2.1.3	Preprocesado	12
2.2.1.4	Categorización Gramatical	17
2.2.1.5	Inserción de Pausas	24
2.2.1.6	Conversión de Grafemas a Unidades Básicas	28
2.2.1.7	Procesado Prosódico	33
2.2.2	Procesado Acústico	47
2.2.2.1	Codificación del Inventario de Voces (Demisílabas)	48
2.2.2.2	Síntesis de Voz	50
3	Análisis y Diseño Orientado a Objetos (UML)	53
3.1	Metodología de Desarrollo	53
3.1.1	Definición de Requerimientos	53
3.1.2	Análisis de Requerimientos	53
3.1.3	Diseño Preliminar	53
3.1.4	Diseño Detallado	54
3.1.5	Implementación y Pruebas	54
3.2	Diseño de Sistemas	54
3.2.1	Arquitectura del Sistema	54
3.2.1.1	Diagrama de Paquetes	54
3.2.1.2	Diagrama de Componentes	56
3.2.2	Diseño de la Implantación	59
3.2.2.1	Herramientas de Desarrollo	59
3.2.2.2	Administración de los Datos	60
3.2.3	Estándares	61
3.3	Diseño de Objetos	61
3.3.1	Diseño Funcional	61
3.3.1.1	Diagrama de Casos de Uso	61
3.3.1.2	Diagrama de Clases	65
3.3.1.3	Diagrama de Estados	99
3.3.2	Diseño Técnico	100
3.3.2.1	Diagramas de Secuencia	100
4	Conclusiones, Observaciones y Recomendaciones	105
4.1	Conclusiones y Observaciones	105
4.2	Recomendaciones	106

Anexos

Anexo A Diagrama de Transiciones del Automata Finito Usado en la Normalización de la Frase	109
Anexo B Algoritmo de Reglas para la Separación de Sílabas de una Palabra	113
B-1 Algoritmo para el caso de inicio de sílaba: Vocal.....	114
B-2 Algoritmo para el caso de inicio de sílaba: Consonante + Vocal	115
B-3 Algoritmo para el caso de inicio de sílaba: Consonante + Consonante.....	115
Anexo C Listado del Diccionario Léxico y la Tabla de Terminaciones	116
C-1 Palabras Átonas.....	116
C-2 Palabras Ambiguas en su Acentuación	117
C-3 Excepciones Gramaticales	117
C-4 Terminaciones y Desinencias.....	122
Anexo D Reglas de Transcripción Grafema-Alófono y Fonetización Sintáctica	125
D-1 Reglas de Transcripción Grafema-Alófono	125
D-2 Reglas Para la Fonetización Sintáctica.....	127
Anexo E Duración Consonántica	128
Anexo F Estructura de Archivos de Datos y la Configuración del Sistema	129
Anexo G Estándares de Programación	132
G-1 Estándares de Pantallas.....	132
G-2 Estándares de Reportes.....	133
G-3 Estándares de Programación.....	134
Anexo H Diseño y Funcionalidad de Menús, Programas y Reportes	135
H-1 Diseño de Menús	135
H-2 Diseño de Programas	136
H-3 Diseño de Reportes	143
Anexo I Resultados Experimentales	144
I-1 Números.....	144
I-2 Fechas y Horas.....	144
I-3 Palabras Complicadas.....	145
I-4 Oraciones.....	145
I-5 Párrafos.....	146
I-6 Observaciones de las personas que escucharon el sistema.....	148

Lista de Ilustraciones

Figura 2.3.1. Diagrama de Bloques General del Sistema Conversor de Texto a Voz.....	6
Figura 2.3.1.1. Bloques del Procesado Lingüístico-Prosódico	8
Figura 2.3.1.5.1. Distribución del número de sílabas en los grupos entonativos.....	26
Figura 3.1.1.1.1 Diagrama de Paquetes del sistema CTVLECTOR.....	55
Figura 3.1.1.2.1 Diagrama de componentes	56
Figura 3.2.1.1.1 Diagrama de CasoS de Uso del sistema CTVLECTOR	61
Figura 3.2.1.2.1 Clases de Interfase Visual	66
Figura 3.2.1.2.2 clases de Interfase de Usuario No Visual	72
Figura 3.2.1.2.3. Clases de Configuración y Mantenimiento	74
Figura 3.2.1.2.4 Diagrama de clases de Interfase del Lector	80
Figura 3.2.1.2.5 Diagrama de Clases del Proceso Lingüístico-Prosódico	82
Figura 3.2.1.2.8 Diagrama de clases de codificación y síntesis de voz	98
Figura 3.2.1.3.1 Diagrama de Estados de la Frase.....	100
Figura 3.2.2.2.1 Diagrama de secuencia Configurar Sistema	101
Figura 3.2.2.2.2 Diagrama de secuencia Manipular Lectura	101
Figura 3.2.2.2.3 Diagrama de secuencia Interface de Usuario Entorno Windows	102
Figura 3.2.2.2.4 Diagrama de secuencia Interfase de Usuario por Línea de Comandos.....	102
Figura 3.2.2.2.5 Diagrama de secuencia del Proceso Lingüístico-Prosódico	103
Figura 3.2.2.2.6 Diagrama de secuencia de la Sintetización de Voz	104
Figura A.1 Para el reconocimiento de Siglas.....	109
Figura A.2 Para el reconocimiento de Nombres Propios	109
Figura A.3 Para el reconocimiento de Abreviaturas.....	109
Figura A.4 Para el reconocimiento de Palabras Compuestas	110
Figura A.5 Para el reconocimiento de Palabras Simples	110
Figura A.6 Para el reconocimiento de Números Reales con Signo.....	110
Figura A.7 Para el reconocimiento de Números Enteros con Signo	110
Figura A.8 Para el reconocimiento de Fechas.....	110
Figura A.9 Para el reconocimiento de Horas con Formato DD/MM/AAAA	111
Figura A.10 Para el reconocimiento de Horas Formato DD/MM/AA	111
Figura A.11 Para el reconocimiento de Monedas (Soles y Dólares)	111
Figura A.12 Para el reconocimiento de Signos Ortográficos	112
Figura G.1.3.1 Plantilla de mantenimiento de datos.....	133
Figura G.1 Jerarquía de ventanas del programa Lector.exe	137
Figura G.2 Diseño de la ventana EditorWinMdi	137
Figura G.3 Diseño de la ventana TextoFrm.....	138
Figura G.4 Diseño de la ventana ConfigurarFrm.....	140
Figura G.5 Diseño de la ventana AtonaFrm	141
Figura G.6 Diseño de la ventana DetFraseFrm.....	142
Figura G.7 Diseño de la ventana FrecFundFrm	142
Figura G.8 Diseño de reporte de Palabras Átonas.....	143

Lista de Tablas

Tabla 2.3.1.2.1. Tipos de palabra.....	10
Tabla 2.3.1.2.2. Tabla de pausas.....	11
Tabla 2.3.1.4.1. Categorías gramaticales empleadas en el análisis lingüístico.....	18
Tabla 2.3.1.4.2. ejemplo de ASIGNACIÓN de categorías gramaticales a una frase.....	21
Tabla 2.3.1.4.3. ejemplo de DESAMBIGUACIÓN gramatical.....	24
Tabla 2.3.1.7.1 Estructura de sílabas y demisílabas usadas en el sistema.....	35
Tabla 2.3.1.7.2 Distribución de la duración de una vocal entre las demisílabas que la contienen.....	39
Tabla 2.3.1.7.3. Duración base de las consonantes.....	40
Tabla 2.3.1.7.4 Tipos de Frase.....	44
Tabla 3.2.1.1.1 Definición de actores.....	61
Tabla 3.2.1.1.2 caso de uso Configurar Sistema.....	62
Tabla 3.2.1.1.3 caso de uso Manejar Lectura.....	63
Tabla 3.2.1.1.4 caso de uso Leer Texto.....	63
Tabla 3.2.1.1.5 caso de uso Obtener Unidades de Voz.....	64
Tabla 3.2.1.2.1 Diccionario de datos de la clase TEditorWin.....	66
Tabla 3.2.1.2.2 Diccionario de datos de la clase TTexto.....	68
Tabla 3.2.1.2.3 Diccionario de datos de la clase TDetFrase.....	70
Tabla 3.2.1.2.4 Diccionario de datos de la clase TFrecFund.....	71
Tabla 3.2.1.2.5 Diccionario de datos de la clase TCatGramOut.....	71
Tabla 3.2.1.2.6 Diccionario de datos de la clase TDuracion.....	71
Tabla 3.2.1.2.7 Diccionario de datos de la clase cTexto.....	72
Tabla 3.2.1.2.8 Diccionario de datos de la clase TConfigurar.....	74
Tabla 3.2.1.2.9 Diccionario de datos de la clase TMantBase.....	76
Tabla 3.2.1.2.10 Diccionario de datos de la clase TAbrev.....	76
Tabla 3.2.1.2.11 Diccionario de datos de la clase TSigla.....	77
Tabla 3.2.1.2.12 Diccionario de datos de la clase TDurBaseC.....	77
Tabla 3.2.1.2.13 Diccionario de datos de la clase TAtona.....	77
Tabla 3.2.1.2.14 Diccionario de datos de la clase TArcTerm.....	78
Tabla 3.2.1.2.15 Diccionario de datos de la clase TPalGram.....	78
Tabla 3.2.1.2.16 Diccionario de datos de la clase TPausa.....	78
Tabla 3.2.1.2.17 Diccionario de datos de la clase TAmbiguo.....	79
Tabla 3.2.1.2.18 Diccionario de datos de la clase TReglaDurCons.....	79
Tabla 3.2.1.2.19 Diccionario de datos de la clase TDurBaseV.....	79
Tabla 3.2.1.2.20 Diccionario de datos de la clase TCatGram.....	80
Tabla 3.2.1.2.21 Diccionario de datos de la clase cEnlazador.....	80
Tabla 3.2.1.2.22 Diccionario de datos de las Funciones de Interfase.....	82
Tabla 3.2.1.2.23 Diccionario de datos de la clase cFrase.....	82
Tabla 3.2.1.2.24 Diccionario de datos de la clase cPalabra.....	87
Tabla 3.2.1.2.25 Diccionario de datos de la clase cSilabeo.....	91
Tabla 3.2.1.2.26 Diccionario de datos de la clase cProsodia.....	92
Tabla 3.2.1.2.27 Diccionario de datos de la clase cFonema.....	95
Tabla 3.2.1.2.28 Diccionario de datos de la clase cConsonante.....	95
Tabla 3.2.1.2.29 Diccionario de datos de la clase cVocal.....	97
Tabla 3.2.1.2.30 Diccionario de datos de la clase cSintetizador.....	98
Tabla 3.2.1.2.31 Diccionario de datos de la clase cSenial.....	99
Tabla B.1 Notación Utilizada.....	114
Tabla B.1.1 Posibles casos de terminación de una sílaba que inicia con una vocal.....	114
Tabla B.2.1 Posibles casos de terminación de una sílaba que inicia con Consonante + Vocal.....	115
Tabla B.3.1 Posibles casos de terminación de una sílaba que inicia con Consonante + Consonante.....	115
Tabla C.1. palabras consideradas como excepciones gramaticales.....	116
Tabla C.2.1 palabras ambiguas consideradas en el sistema.....	117
Tabla C.3.1 Listado de excepciones gramaticales.....	117

Tabla C.4.1 Tabla de terminaciones y desinencias.....	122
Tabla D.1.1 Reglas de Transcripción Grafema-Alófono	125
Tabla D.2.1 Reglas para la fonetización sintáctica.....	127
Tabla E.1 Duración Consonántica Según el Acento, la Posición en la Sílabla y la Posición en el Grupo Fónico.....	128
Tabla F.1. Estructura y Definición de los Archivos de Datos del Sistema	129
Tabla F.2. Estructura de la Configuración del Sistema	130
Tabla G.1.1.1 Estándares para nombre de archivos.....	132
Tabla G.1.2.1 Estándares para controles	132
Tabla G.1.3.1 Plantilla de Botones.....	133



1

Introducción

1.1 Definición del Dominio del Problema

La naturaleza ha brindado al hombre la capacidad de comunicarse a través del habla, y esta capacidad de hablar a distinguido al hombre de otras especies permitiéndole constituirse como un ser sociable y organizacional.

Por otro lado, desde un punto de vista tecnológico, la autopista de la información y los sistemas informáticos ofrecen una serie casi ilimitada de posibilidades de acceso a cualquier tipo de documento (información). De esta conjunción entre posibilidades tecnológicas y nuestra esencia comunicativa a través del habla nace lo que se llama las Tecnologías del Habla, y que está cobrando cada día mayor importancia en la interacción hombre-máquina.

Dentro de esta tecnología se encuentran, los sistemas de conversión de texto a voz, que si bien es cierto, en estos últimos años han tenido un gran avance en lograr una producción de voz de alta calidad, esta todavía no es lo suficientemente buena como para ser usada en aplicaciones tanto personales como dirigidas al público, especialmente en sistemas para el Castellano.

Es así, que el problema principal que se trata de resolver en esta Tesis es, no solo que el texto leído por el sistema sea inteligible, sino que también suene lo más natural posible, emulando la forma como lee un ser humano.

1.2 Aplicaciones de los Conversores Texto a Voz

Los sistemas de conversión de texto a voz se están introduciendo en un amplio rango de aplicaciones. Estos son algunos ejemplos de las múltiples posibilidades que ofrece la conversión texto-voz: Sistemas de lectura de textos para ciegos, sistemas de habla para personas mudas, verificación de textos, ayuda mediante guía telefónica, lectura de correo electrónico, fax, alarmas habladas,

servicios bancarios, sistemas interactivos de voz (acceso a bases de datos), lectura de información variable: pronósticos meteorológicos, estado del tráfico, noticias.

En cualquiera de las aplicaciones el mensaje a reproducir se especifica de forma simbólica (texto) y luego se convierte en voz.



2

Descripción del Desarrollo del Sistema Conversor Texto-Voz (CTVlector)

2.1 Generalidades

Como se mencionó anteriormente, para que un sistema conversor de texto-voz genere voz de alta calidad debe ser desarrollado tomando en cuenta diferentes áreas del saber humano, a esta agrupación se conoce como “Tecnología del Habla”. Una de sus principales áreas viene a ser la Lingüística Computacional, que se encarga de estudiar o formular las teorías y algoritmos por medio de los cuales los procedimientos lingüísticos se puedan implementar en una computadora.

El principal objetivo de la teoría del lenguaje es mostrar como entendemos unidades largas de texto combinando nuestro entendimiento de las unidades más pequeñas. Para hacer esto se subdivide esta tarea en dos partes: la *sintaxis* que describe como los diferentes elementos formales de una unidad de texto, frecuentemente la sentencia (en nuestro caso será llamado frase), pueden ser combinados; y la *semántica* que describe como la interpretación de esta sentencia puede ser calculada.

Se identifican al menos los siguientes tipos de conocimiento que son relevantes para que un sistema de síntesis convierta un texto en su correspondiente cadena hablada:

1. *Conocimiento Fonológico*: Información sobre el sistema de sonidos y la estructura de las palabras y expresiones, los patrones rítmicos de acentuación, entonación, etc.
2. *Conocimiento Morfológico*: Información sobre las estructuras de las palabras; por ejemplo, que fonemas se añaden a los sustantivos para formar el plural.

En nuestro caso, se tiene un diccionario de palabras y terminaciones de estas que nos permiten identificar la posible categoría gramatical a la que pudieran pertenecer.

3. *Conocimiento Sintáctico*: Información sobre la organización de las palabras en frases y oraciones; por ejemplo, cuando en el contexto izquierdo de una categoría gramatical desconocida se encuentra una preposición y un artículo o pronombre preverbal, y en el contexto derecho una preposición, entonces se puede decir, que la categoría desconocida es un sustantivo. En el sistema se maneja un conjunto de reglas gramaticales para realizar esta tarea.
4. *Conocimiento Semántico*: Información sobre el significado de las palabras y de cómo esos significados se combinan para formar el significado de las oraciones; por ejemplo, que el concepto denotado por *pegar* implica dos entidades: un agente (el que pega) y un paciente (la persona o cosa a la que el agente pega), y que esas entidades estarán representadas, correspondientemente, en el sujeto y el objeto del verbo. En el presente trabajo no se implementa este conocimiento debido a que es muy complicada su representación y no se dispone de información necesaria para su desarrollo.
5. *Conocimiento Pragmático*: Información que puede ser considerada central en muchas tareas específicas que los sistemas de Procesamiento de Lenguaje Natural han de realizar, por ejemplo, el análisis de las presuposiciones del hablante, o de las intenciones comunicativas que subyacen en una frase en particular. Casi todas estas tareas implican información sobre la intención y la disposición de los interlocutores, sobre el contexto extralingüístico, etc. Este punto tampoco es implementado debido a que corresponde a un contexto específico y viene a ser un tema extralingüístico.

2.1.1 Proceso de Lectura de una Persona

Debido a que un Sistema Conversor Texto a Voz(CTV) trata de emular el modo en que una persona lee en voz alta, analizaremos los procesos que un ser humano realiza al leer para ayudar a clarificar las tareas que debe realizar el CTV.

Al leer en voz alta, el objetivo generalmente es establecer una comunicación con el oyente. Para que esa comunicación sea efectiva es necesario que emisor y receptor compartan el mismo código, que en este caso vendría a ser el mismo lenguaje. Otro factor importante en la comunicación es el contexto, junto a otros factores no lingüísticos, que aunque son independientes del lenguaje, en muchos casos son imprescindibles para interpretar correctamente los mensajes.

Otra característica de la comunicación hablada es que cada uno interpreta los mensajes de una manera particular. De esta forma, al leer no sólo se utilizan códigos y normativas comunes a los hablantes, sino que también aparecen factores individuales que dan personalidad y "calidad" a una lectura que de otra forma sólo sería correcta en un único contexto.

Cuando se lee, el lector reconoce los caracteres y los relaciona con las abstracciones de los sonidos, los fonemas que comparten los hablantes de una lengua. Sin embargo, esta conversión no se hace de manera directa. Salvo en el primer momento del aprendizaje, el lector no identifica y pronuncia el sonido correspondiente a cada letra, sino que reconoce estructuras más complejas, como palabras y sentencias (en nuestro caso será la frase). De esta manera, habitualmente se puede interpretar y leer correctamente un texto con errores tipográficos, sin que el lector se dé cuenta siquiera de los mismos. Lo mismo ocurre con otros tipos de caracteres impresos, que son tan usuales que se manejan como si estuvieran escritos de manera literal, ejemplos de estos son: los números, fechas, las abreviaturas más comunes, etc.

Además de identificar los sonidos que debe pronunciar para leer un texto, el lector es capaz de entenderlo, haciendo un análisis sintáctico y semántico de las palabras que conforman la sentencia. Estas tareas se realizan simultáneamente y entremezcladas. A partir de esta comprensión, junto con otros factores (como el estado de ánimo, o la intención que se quiera dar al mensaje) se genera la prosodia. La información de los fonemas (información segmental) se refleja en el contenido espectral de la señal de voz, y en

instantes concretos, mientras que la prosodia (información suprasegmental) modifica principalmente otros parámetros como son: la frecuencia fundamental, la duración de cada unidad y la energía, a lo largo del discurso.

2.2 Generación Sintética de la Voz

La generación de voz es el proceso que permite la transformación de una cadena fonética y símbolos prosódicos en una señal sintética de voz. La calidad del resultado esta en función de la calidad de la cadena, así como también de la calidad de generación de la señal de voz en si misma.

Para obtener un sistema conversor texto a voz de gran calidad, este debe ir más allá de la simple inteligibilidad, sino que además debe de tratar de conseguir la naturalidad de un hablante humano.

Un lector humano introduce ante un texto una información que se conoce como prosodia. La prosodia relaciona los diferentes sonidos del mensaje hablado, y permite reflejar tanto elemento lingüísticos, imprescindibles para el sentido de la oración (modalidad, énfasis, etc.), como elementos no lingüísticos (características del locutor, estado de ánimo, etc.). Por consiguiente, la habilidad del lector para reflejar acústicamente el sentido del mensaje mide la calidad del proceso de lectura que realiza.

El conversor texto-voz se compone de dos partes o módulos principales claramente diferenciados: un bloque de procesado lingüístico-prosódico y un bloque de procesado acústico o de síntesis de voz, como se muestra en el siguiente gráfico:

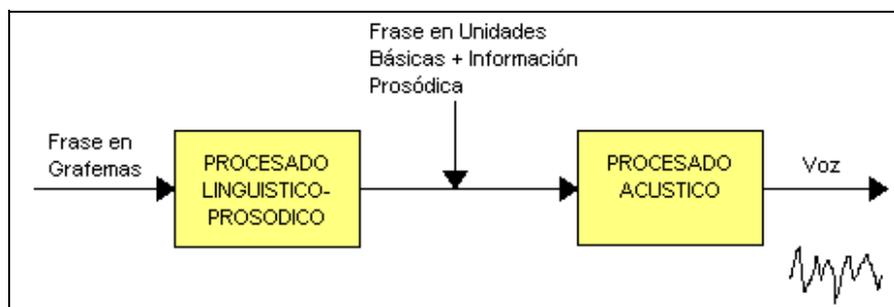


FIGURA 2.3.1. DIAGRAMA DE BLOQUES GENERAL DEL SISTEMA CONVERSOR DE TEXTO A VOZ

Comparándolo con un lector humano se puede decir que el bloque de procesamiento lingüístico-prosódico hace las veces de la vista y cerebro humano, debido a que es la encargada de capturar los caracteres que componen la frase (la vista), hacer la expansión de los textos que no están escritos de manera literal (números, abreviaturas, etc.), hacer el análisis sintáctico de la frase, y basado en la información anterior generar los segmentos de sonido y la forma (prosodia) en que se van a pronunciar.

El procesamiento acústico es la que hace la labor del aparato fonador humano, encargado de generar la forma de onda correspondiente a los segmentos de sonido y la prosodia generada en la etapa anterior.

2.2.1 Procesado Lingüístico-Prosódico

El objetivo general de este bloque es generar a partir de un texto dos tipos de información que se conocen como información segmental y suprasegmental:

- La información *segmental*, es la asociada a la cadena de sonidos que componen el mensaje. Los sonidos que se pueden producir con los órganos de articulación son muy variados, aún si se considera un único locutor. Pero cada idioma tiene seleccionado una serie limitada de “sonidos ideales” aceptados por todos para su uso en el habla. Estas representaciones se llaman fonemas y su número depende de cada idioma, en el caso del Español existen 24 fonemas [Navarro 1972]. Sin embargo, debido al carácter continuo del habla hay variaciones en el punto, modo y carácter sordo / sonoro de cada sonido ideal, estas condiciones dan lugar a los diferentes alófonos de un fonema. El alófono es usualmente definida como una realización acústica de un fonema. El alfabeto fonético es usualmente dividido en dos principales categorías: vocales y consonantes. Las vocales son siempre sonidos sonoros y son producidos por las cuerdas vocales en vibración, mientras que los consonantes pueden ser cualquiera sonoros o sordos.
- La información suprasegmental, es aquella que esta asociada a la prosodia. En consecuencia, refleja tanto elementos lingüísticos(tipo de frase, pausas, acentos, etc.), como elementos no lingüísticos

(características personales del locutor, estado de ánimo, etc.). Esta información es la clave para conseguir una alta naturalidad en los sistemas de síntesis de voz, y generalmente viene codificada a través de tres parámetros acústicos de la señal de voz: la evolución temporal de la frecuencia fundamental, la duración de los segmentos o sonidos, y la curva de energía de la señal acústica. Siendo las dos primeras las más importantes y las que serán objeto de estudio e implementación en esta Tesis.

A continuación se presentan los aspectos prácticos de los bloques funcionales del Proceso Lingüístico-Prosódico, según se han implementado en el sistema desarrollado como parte de esta Tesis, los bloques considerados se muestran en la Figura 2.3.1.1.

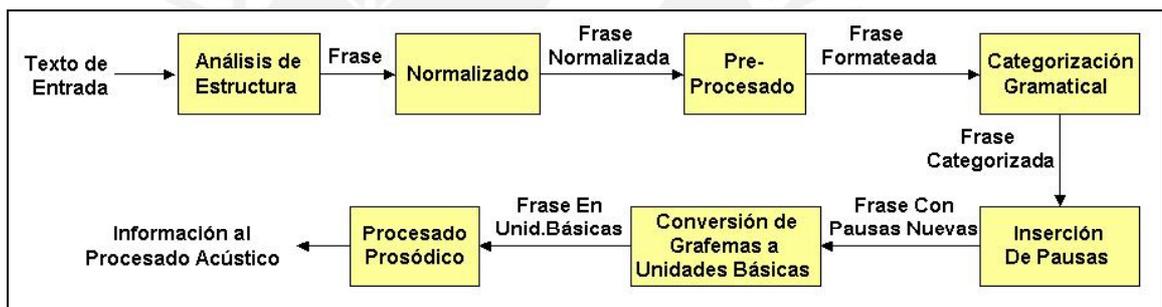


FIGURA 2.3.1.1. BLOQUES DEL PROCESADO LINGÜÍSTICO-PROSÓDICO

2.2.1.1 Análisis de Estructura:

Este proceso realiza la segmentación del texto de entrada que consiste en proporcionar unidades básicas para el análisis lingüístico. Se ha elegido como la unidad de síntesis a la “frase”, debido a que dos frases consecutivas son prácticamente independientes tanto lingüística como prosódicamente.

Para este trabajo se define la frase como un conjunto de caracteres delimitados entre determinados signos ortográficos, llamados de “fin de frase”. Los signos que se han considerado como marcadores de fin de frase son: el punto fin de frase (.), los puntos suspensivos (...), dos puntos (:), punto y coma (;), Inicio de interrogación (¿), cierre de interrogación (?), inicio de exclamación (!) y el cierre de exclamación (!). Es así, que la frase puede contener signos ortográficos que no están definidos como fin de frase como

son las comas, paréntesis, lo que nos permite realizar un mejor análisis lingüístico y prosódico.

La tarea de obtener la frase de un texto no es, en modo alguno, trivial debido a que algunos signos ortográficos se pueden presentar como parte de las palabras, como por ejemplo: el punto, que puede ser punto de una abreviatura, sigla, fecha (dd.mm.aaaa), separador de decimales. Además, hay que distinguir el punto de los puntos suspensivos. Otro caso, es el de los dos puntos, que puede aparecer como separador de los minutos y segundos en las palabras tipo hora (hh:mm:ss).

La implementación del reconocimiento de la frase en un texto se realiza haciendo uso de un conjunto de reglas, que cubren de manera correcta prácticamente todos los casos para un archivo de texto normal ortográficamente bien escrito. El sistema reconoce un signo ortográfico en el texto, si este signo está definido como “fin de frase”, se pasa a aplicar las reglas de contexto para verificar si se encontró el final de una frase o no. En caso que las reglas confirmen que es el final de una frase, se devuelve el texto desde donde se inició el análisis hasta el signo para su procesado en los siguientes módulos del sistema. Caso, contrario se continua con el análisis del siguiente signo. Este proceso se repite hasta leer todas las frases del texto o hasta que se detenga la lectura del texto.

Este proceso es la interfaz entre el sistema y la fuente del texto razón por la cual es muy dependiente de la aplicación para la que se use el conversor, por ejemplo, si la fuente del texto es un scanner de libros la forma de obtener las frases será diferente a la forma como se obtiene de un editor de textos o del texto almacenado en una base de datos. En nuestro caso, este es el único módulo que esta fuera de la librería del lector(Lector.dll), y se implementa en los programas ejecutables (Lector.exe y LectorLC.exe).

2.2.1.2 Normalizado

El normalizador se ocupa de realizar una serie de tareas previas sobre la frase, de modo que los módulos siguientes encuentren unos datos que en su estructura presenten menos variabilidad para su análisis. Al realizar estas

tareas se fija y uniformiza la representación de los caracteres del texto. Para cumplir con este objetivo el sistema desarrollado realiza lo siguiente:

1. Separa e identifica todas las palabras de la frase, usando una lista doblemente enlazada para almacenarlas. La razón por la cual se usa esta estructura es para facilitar el acceso a las palabras anterior y posterior de la que se está analizando, y de esta manera facilitar la aplicación de las reglas de reconocimiento y categorización gramatical que dependen del contexto en que se encuentra una palabra.

TABLA 2.3.1.2.1. TIPOS DE PALABRA

Código	Descripción
PALSIMPLE	Palabra simple o común
PALCOMP	Palabra compuesta separada por un guión
NOMPROPIO	Nombre propio
ABREV	Abreviatura
SIGLA	Sigla o acrónimo
NUMERAL	Número real
NUMENTERO	Número entero
FECHA1	Fecha con formato DD/MM/AAAA
FECHA2	Fecha con formato DD/MM/AA
HORA1	Hora con formato HH:MI:SS
HORA2	Hora con formato HH:MI
SOLES	Moneda tipo Nuevos Soles (S/.)
DOLARES	Moneda tipo Dólares Americanos (\$)
PUNTO	Punto (.)
PUNTOSUSP	Puntos suspensivos (...)
DOSPUNTOS	Dos puntos (:)
COMA	Coma (,)
PUNTOYCOMA	Punto y coma (;)
INICOMILLA	Comilla de inicio (“
FINCOMILLA	Comilla de cierre (“
GUION	Guión(-)
INIPREGUNTA	Inicio de interrogación (¿)
FINPREGUNTA	Cierre de interrogación (?)
INIEXCLAMACION	Inicio de exclamación (¡)
FINEXCLAMACION	Cierre de exclamación (!)
INIPARENTESIS	Inicio de paréntesis (“(”)
FINPARENTESIS	Cierre de paréntesis (“)”)
FINFRASE	Signo ortográfico de fin de frase
DESCONOCIDO	Palabra no reconocida

Al emplear el término palabra nos estamos refiriendo a la agrupación de caracteres separados por una sucesión de blancos, tabulaciones, saltos de línea, retornos de carro. Es decir, no siempre son palabras propiamente dichas o como comúnmente se entiende, sino que se

incluyen también a las abreviaturas, números, fechas, monedas, signos ortográficos, etc. En la Tabla 2.3.1.2.1, se presenta el listado de los tipos de palabra que reconoce el sistema. El reconocimiento del tipo de palabra, se hace mediante la implementación de un autómata finito determinístico que reconoce las secuencias regulares correspondientes a cada tipo de palabra definido, los diagramas del autómata finito se muestran en el Anexo A.

2. Se normaliza la escritura de las palabras para reducir su variabilidad tipográfica. En nuestro caso, pasamos las palabras a letras minúsculas, pero guardando la información del modo como originalmente estaban escritas, estas pueden ser: todo en mayúsculas, todo en minúsculas, inicial en mayúscula, o variable (minúsculas y mayúsculas combinadas).
3. Se reconocen los signos ortográficos, y se les asigna el código del tipo de pausa que les corresponde. En la Tabla 2.3.1.2.2 se muestran los tipos de pausa que usa el sistema. A las palabras comunes se les asigna temporalmente un código de ausencia de pausa, más adelante el proceso de “Inserción de pausas no marcadas ortográficamente” puede modificar esta información.

TABLA 2.3.1.2.2. TABLA DE PAUSAS

Código	Descripción	Duración(ms)
255	No pausa	0
1	Punto	921
2	Puntos suspensivos	669
3	Dos puntos	727
4	Coma	396
5	Punto y coma	799
6	Comilla de Inicio	577
7	Comilla de cierre	421
8	Guión	387
9	Interrogación de cierre	897
10	Exclamación de cierre	817
11	Paréntesis de inicio	412
12	Paréntesis de cierre	498
13	Pausa antes de nexo de coordinación	400
14	Pausa antes de nexo de subordinación	390
15	Pausa antes de un pronombre relativo	380
16	Pausa antes de un verbo	350
17	Pausa antes de una preposición	300
18	Pausa antes de palabra función	280

4. Se reconoce el signo ortográfico que hace de marcador de fin de frase, esta información servirá más adelante para determinar el tipo de frase que se está procesando (enunciación, interrogación, exclamación).

2.2.1.3 Preprocesado

Este bloque realiza tres tareas:

1. Expansión de Palabras Especiales: Un texto normal, sin restricciones, puede contener expresiones que no estén formadas por letras y palabras, y que es necesario interpretarlas para la correcta y completa categorización de la frase y en definitiva para su correcta lectura. Luego de la normalización donde se delimitó la frase y las diferentes palabras que la componen, el siguiente paso es expandir aquellas expresiones que requieren un procesado especial como los números, fechas, monedas. El preprocesado se encarga de sustituir estas expresiones o palabras especiales por la secuencia de caracteres alfabéticos que equivale a la lectura que haría una persona. A continuación se definen las palabras especiales con las que trabaja el sistema:
 - *Abreviatura*: Formada de una sola letra o palabra y que no se puede pronunciar en la mayoría de los casos: Sr., señor. Las abreviaturas llevan punto y cuando van seguidas van separadas por espacio, después del signo admiten cualquier signo de puntuación excepto el punto. Para la detección de abreviaturas, el sistema comprueba que se trata de palabras terminadas en punto que se encuentran en el diccionario de abreviaturas.
 - *Siglas*: Al igual que las abreviaturas se definen vía una expresión regular. Son siglas palabras de entre dos y seis letras escritas en mayúsculas (ETA), o palabras con letras mayúsculas seguidas de números(U2), o palabras que tengan la misma definición que las abreviaturas pero con letras en mayúsculas(S.A). Para su reconocimiento el sistema verifica que la palabra candidata este en la tabla de siglas.

- *Números*: Conjunto de dígitos que tienen una de las dos siguientes estructuras: Uno o más dígitos del 0 al 9 (245). Uno o más dígitos seguidos por un punto y a continuación, uno o más dígitos del 0 al 9 (1536.04). Los dígitos de un mismo número no pueden estar separadas por una coma o espacios, el único signo admitido es el punto separador de decimales. La expansión de los números se realiza mediante unas reglas de conversión contenidas en funciones del programa. El sistema puede leer números desde el cero hasta cuatrillón, entre enteros y reales, incluyendo sus negativos.
 - *Fechas*: Se define como un conjunto de números con los siguientes formatos: DD/MM/AAAA, DD-MM-AAAA, DD/MM/AA, DD-MM-AA. Se expande mediante una función del sistema.
 - *Horas*: Se define como dos o tres grupos de dígitos, separados por dos puntos (:), con los siguientes formatos: HH:MI:SS (hora:minuto:segundo) y HH:MI (hora:minuto). Se expande mediante una función del sistema.
 - *Moneda*: Se consideran dos tipos de monedas, las que corresponden a los siguientes símbolos S/. (Soles) y \$ (Dólares). Si existe algún número después de estos símbolos se lee como una cantidad monetaria.
 - *Palabra Compuesta*: Dos palabras que se caracterizan por estar unidas por un guión. El sistema simplemente las separa en dos palabras distintas.
2. *Silabificación*: Consiste en dividir en sílabas las palabras de una frase, marcando los límites que separan una sílaba de otra. La información sobre la división en sílabas de la palabra es necesaria para poder determinar la acentuación fonética, influye además al momento de realizar la transcripción fonética, y controla las reglas de entonación y pausado. Con la finalidad de hacer más rápido y eficiente el procesamiento del sistema, se reemplazó el módulo de silabificación de la versión anterior. En este caso hemos utilizado un conjunto de reglas, que se describen en

[Torrejón, 1999], [Díaz, 1970], [Cabral, 1978] y [Mateo, 2000]. A continuación se detallan las reglas implementadas:

- Regla 1: En las sílabas, por lo menos, siempre tiene que haber una vocal. Sin vocal no hay sílaba.
- Regla 2: Cada elemento del grupo de consonantes inseparables, que forman el grupo consonántico inicial (br, bl, cr, cl, dr, fr, fl, gr, gl, kr, ll, pr, pl, rr, tr, ch), y el grupo consonántico final (bs, ns); no pueden ser separados al dividir una palabra en sílabas. Esta regla no se cumple cuando se trata del prefijo “sub” seguido de “r”, en este caso “br”, se separa (subrayar / sub-ra-yar).
- Regla 3: Una consonante entre dos vocales, siempre forma sílaba con la segunda vocal.
- Regla 4: Cuando hay dos consonantes entre dos vocales, cada vocal se une a una consonante. Esta regla no se cumple, cuando las dos consonantes forman un grupo consonántico inicial.
- Regla 5: Si son tres las consonantes colocadas entre dos vocales, puede suceder: que la segunda y tercera consonantes formen un grupo consonántico inicial, en tal caso, estas forman sílaba con la segunda vocal; o puede ser, que la primera y segunda consonantes formen un grupo consonántico final, en tal caso, forman sílaba con la primera vocal.
- Regla 6: La única forma en que cuatro consonantes aparezcan juntas entre vocales, es que, las dos primeras formen un grupo consonántico final (forman sílaba con la primera vocal), y la tercera y cuarta consonantes formen un grupo consonántico inicial (forman sílaba con la segunda vocal).
- Regla 7: Las palabras que contienen una “h” precedida o seguida de otra consonante, se dividen separando ambas letras.
- Regla 8: El diptongo es la unión inseparable de dos vocales consecutivas. Se forman de las siguientes formas: (vocal abierta +

vocal cerrada) o (vocal cerrada + vocal abierta) o (vocal cerrada + vocal cerrada). La “y” al final de sílaba se comporta como “i”, es decir, como una vocal cerrada, en este caso puede formar parte de un diptongo.

- Regla 9: La unión de dos vocales abiertas o semiabiertas no forma diptongo.
- Regla 10: La “h” entre dos vocales, no destruye un diptongo.
- Regla 11: La acentuación sobre la vocal cerrada de un diptongo provoca su destrucción y constituye un hiato.
- Regla 12: La unión de tres vocales forma un triptongo. La única disposición posible para la formación de triptongos es la siguiente: vocal cerrada + (vocal abierta o vocal semiabierta) + vocal cerrada.

El algoritmo seguido para la implementación de estas reglas se muestra en el Anexo B.

3. Acentuación Fonética: En el idioma Castellano, las palabras normalmente tienen un único acento fonético cuya posición queda determinada por la acentuación ortográfica, el número de sílabas y las letras en que terminan las palabras. Pero hay un grupo de palabras que tienen dos acentos fonéticos, estos son los adverbios procedentes de adjetivos terminados en “mente”. En ellos un acento recae sobre la raíz de la palabra que corresponde al adjetivo, y el otro en la primera “e” de “-mente”, este caso se incluye en las reglas de acentuación.

Existen también algunas palabras inacentuadas, estas son las palabras llamadas de función que no tienen sentido propio, entre estas tenemos los artículos, algunos pronombres, las conjunciones, entre otras. Estas palabras afortunadamente pertenecen a series cerradas, es decir, en el Castellano existe un conjunto limitado de palabras inacentuadas y que son fácilmente identificables. El sistema maneja una tabla de palabras átonas para identificarlas, esta tabla se lista en el Anexo C (Sección C.1). En esta tabla, adicionalmente se guarda la categoría gramatical a la que pertenece la palabra, debido a que en algunos casos, las palabras tienen

la misma grafía, pero dependiendo de la categoría gramatical a la que pertenecen son palabras acentuadas con tilde o inacentuadas.

Existen, también palabras con la misma grafía, que son acentuadas o inacentuadas fonéticamente dependiendo de la categoría gramatical a la que pertenecen, pero a diferencia del caso anterior no se tildan para diferenciarlos, en este caso el sistema lo único que puede hacer en esta etapa es marcarlas como ambiguas. Para hacer esto, el sistema cuenta con una tabla de palabras ambiguas que se muestra en el Anexo C (Sección C.2).

Para identificar la sílaba acentuada el sistema hace uso de las siguientes reglas, que se definieron sobre la base de los estudios realizados en [Navarro 1948], [Navarro 1972], [Ríos 1999] y [Mateo 2000]:

- Regla 1: La vocal tónica de la sílaba tónica de una palabra sin tilde, será reemplazada por una vocal con tilde [(a,e,i,ou) → (á,é,í,ó,ú)] .
- Regla 2: Palabras que terminan en vocal o en las consonantes “n” o “s” y no tienen tilde, llevan el acento tónico en la penúltima sílaba, entonces, se tilda la penúltima sílaba y se le marca como acentuada. Se marca la palabra como acentuada prosódicamente y como definitivamente acentuada.
- Regla 3: Palabras que terminan en vocal o en las consonantes “n” o “s” y llevan tilde en la última sílaba, se les marca como acentuadas ortográficamente y definitivamente acentuadas. Se marca la sílaba tónica como acentuada.
- Regla 4: Palabras que terminan en consonante, que no sea ni “n” ni “s” y no llevan tilde, llevan el acento tónico en la última sílaba, entonces, se tilda la última sílaba y se le marca como acentuada. Se marca la palabra como acentuada prosódicamente y como definitivamente acentuada.
- Regla 5: Palabras que terminan en consonante que no sea ni “n” ni “s” y llevan tilde en la penúltima sílaba, se les marca como acentuadas

ortográficamente y definitivamente acentuadas. Se marca la sílaba tónica como acentuada.

- Regla 6: Palabras que tienen la sílaba tónica antes de la penúltima sílaba deben estar indicadas mediante una tilde, entonces, se marca la palabra como acentuada ortográficamente y como definitivamente acentuada. Luego, se marca la sílaba tónica como acentuada.
- Regla 7: Cuando en la sílaba tónica hay un diptongo formado por una vocal fuerte y una débil, el acento tónico cae sobre la vocal fuerte.
- Regla 8: Si en la sílaba tónica hay un diptongo de dos vocales débiles, el acento tónico cae sobre la segunda vocal.
- Regla 9: Si la palabra es un adverbio terminado en “mente” (palabramente), se acentúa la primera “e” de “-mente”, y la raíz se acentúa según las reglas de acentuación fonética.
- Regla 10: Las excepciones son las indicadas en las tablas de palabras átonas y ambiguas. En el primer caso, se marca la palabra como átona y definitivamente inacentuada; en el segundo caso, se marca la palabra como ambigua con acentuación no definida.

2.2.1.4 Categorización Gramatical

La principal tarea de este módulo es la de asignar categorías gramaticales a las palabras con el objetivo de que el conversor haga una lectura “con sentido” de los textos, debido a que el análisis sintáctico del texto puede introducir mejoras en la determinación de los grupos entonativos y la duración de las pausas.

Las categorías gramaticales definidas como parte del trabajo de esta Tesis, no son exactamente las mismas categorías gramaticales definidas por la gramática tradicional, sino un conjunto de códigos que en muchos casos se corresponden con verdaderas categorías gramaticales. Estos códigos se definen en el programa como un tipo enumerado y son los que se muestran en la Tabla 2.3.1.4.1.

Los códigos se definen sobre la base de la utilidad que estos pueden tener para el sistema conversor al momento de decidir la prosodia de la frase. Es decir, se ve a la categorización gramatical no como un fin sino como un medio para obtener información relevante que puede ayudar en el marcado prosódico, más específicamente en la inserción de pausas no marcadas ortográficamente; por ejemplo, el decidir si una palabra es adjetivo o un sustantivo puede no ser relevante para la prosodia, pero sí lo será distinguir adecuadamente un verbo, porque hay la posibilidad de que pueda existir una pausa inmediatamente antes de un grupo verbal.

TABLA 2.3.1.4.1. CATEGORÍAS GRAMATICALES EMPLEADAS EN EL ANÁLISIS LINGÜÍSTICO

Código	Descripción
NOMCOMUN	Nombre común o sustantivo
NOMBPROPIO	Nombre propio
PRONPERSONAL	Pronombre personal
PRONPOSESIVO	Pronombre posesivo
PRONRELATIVO	Pronombre relativo
PRONINTERROGATIVO	Pronombre interrogativo
ADJETIVO	Adjetivo
ARTICULO	Artículo determinado y las contracciones(al, del)
ARTINDETERMINADO	Artículo indeterminado
VERBO	Categoría general de verbo
VERBOHABER	Verbo Haber y sus conjugaciones
VERBOSER	Verbo Ser y sus conjugaciones
VERBOESTAR	Verbo Estar y sus conjugaciones
ADVERBIO	Categoría general de adverbio
ADVESPECIFICADOR	Adverbio especificador
PREPOSICION	Categoría general de preposición
PREPVACIA	Preposiciones vacías
CONJUNCIONCOORD	Conjunción coordinante
CONJUNCIONSUB	Conjunción subordinante
INTERJECCION	Interjección
PRONPREVERBAL	Pronombres preverbales
ARTPREV	Pronombre preverbal o artículo
NOMBREADJETIVO	Nombre común o adjetivo
SIGNOORTOGRAFICO	Signo ortográfico
PRONDETERMINE	Pronombre o determinante
NODEFINIDO	Para palabras no identificadas

Otra consideración importante es que a los signos ortográficos se les considera palabras capaces de tener una categoría gramatical, ya que juegan un papel importante en el marcado prosódico, debido a que no sólo indican el tipo de sentencia (enunciación, interrogación, exclamación, etc.), sino que además son fronteras naturales de los grupos entonativos.

El sistema realiza la categorización de las palabras en dos etapas principales, llamadas categorización por diccionario y categorización por reglas:

1. Categorización por Diccionario: Para el sistema un diccionario se define como un arreglo de forma “elemento - categoría gramatical” , donde “elemento” puede ser una palabra, una raíz, un sufijo o prefijo y “categoría gramatical” vienen a ser uno de los códigos definidos en la Tabla 2.3.1.4.1. En el caso, en que un elemento pueda pertenecer a varias categorías gramaticales, aparecerá en el diccionario tantas veces como categorías a las que pertenezca existan.

El diccionario léxico utilizado por el sistema, dispone de reglamentos con palabras pertenecientes a series cerradas (preposiciones, conjunciones, adverbios, pronombres, etc.), y con una tabla de terminaciones (desinencias verbales, sufijos, etc.) que se usa para intentar determinar la categoría gramatical de las palabras pertenecientes a series abiertas (verbos, sustantivos, adjetivos, etc.). El listado de las tablas del diccionario léxico utilizados en el sistema se encuentran en el Anexo C (Secciones C.1, C.3 y C.4).

En el castellano el uso de las reglas de terminaciones es muy útil, debido a que es una lengua que utiliza masivamente la flexión para la formación de las palabras. Se distinguen dos tipos de flexiones: la verbal o conjugación si se realiza con verbos; y la nominal o declinación, si se hacen con formas que admiten género y / o número, como sustantivos y adjetivos.

Usando la tabla de terminaciones, el sistema asigna una categoría en función de la terminación de la palabra considerada. Por ejemplo, la terminación “-able”, se utiliza para formar adjetivos (amigable, saludable), la terminación “-mente” para formar adverbios a partir de adjetivos (ansiosamente, serenamente). Por supuesto, hay palabras que no siguen la regla general, por ejemplo, “demente” no es un adverbio ni sabe un adjetivo. Por esta razón, las reglas de terminación se aplican únicamente

a aquellas palabras que no hayan sido categorizadas por el diccionario de excepciones gramaticales.

En esta etapa, el sistema genera para cada palabra, una lista de sus posibles categorías gramaticales, debido a que una palabra de acuerdo a su grafía o terminación (sufijo o desinencia) podría pertenecer a más de una categoría gramatical. En la siguiente etapa es donde se realizará la desambiguación gramatical, y se le asignará definitivamente una sola categoría gramatical.

El sistema subdivide la “categorización por diccionario” en tres subtarefas, una por cada diccionario utilizado:

- En la primera, verifica si se trata de una palabra átona, si es así, busca la categoría gramatical de la palabra en la tabla de palabras átonas; si lo encuentra se marca la palabra como definitivamente categorizada.
- Si la palabra no está marcada como átona, el sistema busca la palabra en la tabla de excepciones gramaticales, si lo encuentra, esta rutina recupera todas las categorías gramaticales a las que pudiera pertenecer la palabra, es decir, llena su lista de categorías gramaticales, que será desambiguada en una etapa posterior.
- Si en el paso anterior, a la palabra no átona no se le ha asignado ninguna categoría gramatical, se intenta categorizarlo usando la tabla de terminaciones, para tal fin, el sistema iguala una terminación del diccionario con la terminación de la palabra. Esta búsqueda se realiza, empezando por la terminación más larga que hubiera en el diccionario por ser la más restrictiva, una vez que se encuentre la terminación más larga que concuerde con una secuencia de caracteres final de la palabra analizada, se fija la terminación, y la rutina recupera todas las categorías gramaticales asignada a esta terminación y las asigna a la palabra.

- Si en ninguna de las subtarefas anteriores, se logró asignar una categoría gramatical a la palabra, el sistema le asigna la categoría gramatical definida como “NODEFINIDO”.

En la Tabla 2.3.1.4.2 se muestra el resultado de aplicar la categorización por diccionario a la frase, “El niño gritaba y lloraba antes de comer”.

TABLA 2.3.1.4.2. EJEMPLO DE ASIGNACIÓN DE CATEGORÍAS GRAMATICALES A UNA FRASE

Frase	Arreglo de Categorías Gramaticales Asignadas	
	0	1
el	ARTICULO	
niño	NOMBREADJETIVO	
gritaba	VERBO	
y	CONJUNCIONCOORD	
lloraba	VERBO	
antes	ADVERBIO	
de	PREPVACIA	
comer	NOMBREADJETIVO	VERBO
.	SIGNOORTOGRAFICO	

2. Categorización por Reglas Contextuales: Una vez realizado el análisis morfológico y aplicadas las reglas de terminación, se determina definitivamente la categoría de las palabras analizando su contexto gramatical. La finalidad de este paso es reducir la ambigüedad gramatical en aquellas palabras que puedan pertenecer a varias categorías. El problema de la desambiguación léxica para el sistema consiste en que: dada una frase tal que cada una de sus componentes (palabras) tiene asignada una lista de posibles categorías gramaticales, obtener una frase tal que cada palabra tenga una sola categoría gramatical asociada.

La función de las reglas de contexto, es indicar en un contexto dado que categoría debe asignársele a la palabra ambigua; por ejemplo, detrás de una palabra que sólo puede ser preposición seguida de una palabra que puede ser “artículo determinado” o pronombre (nominal o preverbal) aparece un nombre (“por los curas”). El contexto (o entorno) puede referirse a una o varias posiciones anteriores y posteriores al elemento ambiguo analizado.

En ocasiones se puede aplicar una regla a una palabra ambigua considerando un contexto (izquierdo o derecho) y puede suceder que en el contexto también existan palabras ambiguas. Por esta razón, es muy importante el orden en que se apliquen las reglas; el mismo conjunto de reglas aplicado en distinto orden conduce a resultados distintos. Aún más, las reglas deben estar ordenadas comenzando por las más restrictivas.

Las reglas gramaticales implementadas en el sistema tienen el siguiente formato:

Regla(n) \rightarrow [CG (CGI1 CGI2 ... CGI n) | (CGD1 CGD2 ... CGD m)]

Donde:

CG : Categoría gramatical que cumple el contexto de la regla.

CGI n : n-ésima categoría gramatical del contexto izquierdo de la regla que debe estar presente en la frase en la posición “n” a partir de la posición de la palabra analizada hacia la izquierda para que se cumpla la condición.

CGD m : m-ésima categoría gramatical del contexto derecho de la regla que debe de estar presente en la frase en la posición “m” a partir de la posición de la palabra analizada hacia la derecha para que se cumpla la condición.

El sistema realiza la evaluación de las reglas de contexto de la siguiente manera:

- Verifica si la palabra tiene más de una categoría gramatical asignada, si es así, recorre la lista de categorías de la palabra que se está analizando, y verifica primero si el contexto izquierdo de la palabra cumple las condiciones que se especifican en la regla de contexto izquierdo correspondiente a la categoría gramatical en estudio. Si cumple se analiza el contexto derecho. Si ambas cumplen se termina el proceso y se asigna la categoría gramatical que cumplió las condiciones como la categoría final de la palabra. Caso contrario, se continua con la siguiente categoría que tuviera la palabra en su lista.

Finalmente, en caso que ninguna categoría asignada provisionalmente cumpla los requisitos de contexto, se le asigna la categoría “NODEFINIDO”, lo que significa que el sistema con las reglas consideradas no pudo reconocer la categoría gramatical a la que pertenece la palabra.

- Si la palabra no tiene ninguna categoría gramatical asignada o tiene sólo la categoría “DESCONOCIDO”, el sistema intenta asignarle una categoría usando las reglas de contexto, para lo cual se busca una regla que cumpla las condiciones de contexto (izquierdo o derecho) o entorno de la palabra. Si la encuentra se asigna la cabeza de la regla (CG, del formato definido), como la categoría gramatical de la palabra.
- Finalmente, se tiene una rutina que verifica si la categoría gramatical asignada a las palabras de la frase en los pasos anteriores es la correcta. Esta rutina hace uso de ciertas excepciones de construcciones gramaticales que no se pueden expresar en las reglas contextuales.

En la Tabla 2.3.1.4.2 se observa que todas las palabras de la frase “El niño gritaba y lloraba antes de comer.” Han sido categorizadas definitivamente (La categorización por diccionario les ha asignado una sola categoría gramatical) a excepción de la palabra “comer” que tiene asignado dos categorías NOMBREADJETIVO y VERBO. Para esta palabra el sistema determina que tiene asignada dos categorías y que necesita ser desambiguada y procede a aplicar las reglas de contexto para cada categoría asignada a la palabra, iniciando con la categoría NOMBREADJETIVO. El sistema se ubica en la primera regla cuyo foco (cabeza de regla) sea la categoría NOMBREADJETIVO y evalúa si el contexto izquierdo de la regla coincide con el contexto izquierdo de la palabra que se está analizando. En el ejemplo, como se muestra en la Tabla 2.3.1.4.3 la Regla 1 cuyo foco es NOMBREADJETIVO tiene como primera categoría de contexto izquierdo a la categoría PRONPOSESIVO, pero la

palabra “comer” según lo mostrado en la Tabla 2.3.1.4.2 tiene como primera categoría gramatical de contexto izquierdo a la categoría PREPVACIA por la que la evaluación de la Regla 1 falla. Por lo que el sistema continua con la siguiente regla que tenga el mismo foco, hasta la Regla 15 cuyo contexto izquierdo coincide con el de la palabra, y el contexto derecho de la regla es verdadero debido a que es NULO, es decir, para esta regla basta que se cumpla el contexto izquierdo para que la categoría gramatical indicada por el foco sea aceptada. El sistema al aceptar la categoría gramatical NOMBREADJETIVO como válida asigna esta categoría a la palabra “comer” y descarta la categoría VERBO para este contexto, finalizando el proceso de asignación de categorías gramaticales a la frase. En la siguiente tabla se muestra las reglas analizadas por el sistema antes de aceptar como válida la categoría gramatical NOMBREADJETIVO para la palabra “comer” usando la regla 15.

TABLA 2.3.1.4.3. EJEMPLO DE DESAMBIGUACIÓN GRAMATICAL

Nro	Foco	Contexto Izquierdo			Contexto Derecho
		0	1	2	0
Regla 1	NOMBREADJETIVO	PRONPOSESIVO			
Regla 2	NOMBREADJETIVO	PRONRELATIVO			SIGNOORTOGRAFICO
Regla 3	NOMBREADJETIVO	ADJETIVO			SIGNOORTOGRAFICO
Regla 4	NOMBREADJETIVO	ARTICULO			
Regla 5	NOMBREADJETIVO	ARTINDETERMINADO			VERBO
Regla 6	NOMBREADJETIVO	ARTINDETERMINADO			SIGNOORTOGRAFICO
Regla 7	NOMBREADJETIVO	ARTINDETERMINADO	VERBOSER		
Regla 8	NOMBREADJETIVO	ARTINDETERMINADO			ADVESPECIFICADOR
Regla 9	NOMBREADJETIVO	ARTINDETERMINADO			PREPOSICION
Regla 10	NOMBREADJETIVO	ARTINDETERMINADO			PREPVACIA
Regla 11	NOMBREADJETIVO	PREPOSICION	PRONPREVERBAL		PREPOSICION
Regla 12	NOMBREADJETIVO	PREPVACIA	VERBO	VERBOESTAR	ARTINDETERMINADO
Regla 13	NOMBREADJETIVO	PREPVACIA	NOMBREADJETIVO		CONJUNCIONCOORD
Regla 14	NOMBREADJETIVO	PREPVACIA	NOMCOMUN		CONJUNCIONCOORD
Regla 15	NOMBREADJETIVO	PREPVACIA	ADVERBIO		

2.2.1.5 Inserción de Pausas

La tarea principal de este módulo consiste en la localización de pausas no marcadas ortográficamente. Esto debido a que al leer un texto, es muy frecuente que los lectores hagan más pausas que las que vienen marcadas por los signos ortográficos. Este hecho normalmente viene motivado por la

necesidad que tiene el hablante humano de recuperar el aliento. Además, la inserción de pausas, junto con la utilización de la prosodia, permiten transmitir parte del contenido del mensaje y facilitar así la comprensión de la frase por el oyente.

Las pausas insertadas automáticamente, no sólo ayudan a no dar la desagradable sensación de ahogo al oyente que escuchará leer un texto largo sin pausa alguna, sino que también determina la evolución del contorno entonativo de la frase. Este contorno entonativo, llamado también grupo entonativo según [Navarro, 1978] es la cadena de grupos acentúales que la prosodia distingue de otros grupos adyacentes. Estos grupos entonativos suelen coincidir con los grupos fónicos, que son unidades delimitadas por dos pausas consecutivas en la elocución. Según el mismo autor, el grupo entonativo es en general una unidad inferior al grupo fónico, porque no necesita siempre de pausas, puede ir marcado por depresiones de intensidad, retraso de la articulación y cambios bruscos en la altura musical. En este trabajo por razones prácticas, se considera que un grupo entonativo siempre esta marcado por pausa, por lo que para nosotros será equivalente a un grupo fónico.

El criterio escogido para decidir si una secuencia de palabras sin pausa entre ellas es "demasiado larga" consiste en contar el número total de sílabas que hay en la frase entre dos pausas marcadas ortográficamente o pausas automáticas insertadas en la frase previamente. Si este número supera el límite prefijado, se considera que hay que introducir una pausa. El límite escogido en este trabajo es de 16 sílabas, este número se escogió sobre la base del resultado del análisis estadístico del número de sílabas de los grupos entonativos, mostrado en la Figura 2.3.1.5.1, y que se obtuvo de [López, 1993], como se observa la mayor parte de grupos entonativos tienen menos de 16 sílabas.

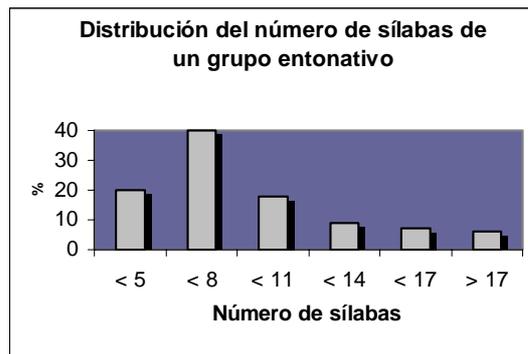


FIGURA 2.3.1.5.1. DISTRIBUCIÓN DEL NÚMERO DE SÍLABAS EN LOS GRUPOS ENTONATIVOS

Las pausas que se inserten en este módulo influirán en el momento de la conversión de grafemas a alófonos, ya que algunas reglas de conversión grafema-alófono consultan la existencia de pausa delante y detrás de la letra bajo estudio. Estas pausas también influyen en la acentuación de algunas palabras, y finalmente a la hora de decidir la duración de los alófonos en la frase.

En el sistema se implementa esta tarea mediante una función recursiva, la cual detecta una secuencia de sílabas en la frase que supere el límite (16 sílabas consecutivas sin pausa), una vez detectada procede a buscar la mejor posición para insertar una nueva pausa de acuerdo a las categorías gramaticales asignadas a las palabras. La idea es buscar palabras que por su categoría gramatical puedan ser separadores entre sintagmas*. La palabra cuya categoría gramatical tenga la prioridad más alta es el primer candidato para insertar una pausa. A continuación se describen las categorías gramaticales donde es posible insertar las pausas automáticas, estas categorías están ordenadas por su prioridad en orden descendente:

1. Conjunciones coordinantes: Son las que tienen la prioridad más alta, debido a que dos sintagmas coordinantes son prácticamente independientes cuando separan cláusulas en la oración.
2. Conjunciones subordinantes: Tienen prioridad cinco, debido a que dos sintagmas o proposiciones también mantienen una relación de

* Unión de varias palabras alrededor de una más importante que se llama núcleo, ejemplo: La trompeta suena

independencia entre ambas, pero menos que en el caso anterior. Por tanto definen una pausa de menor duración que en el caso anterior.

3. Pronombres relativos: Tienen prioridad cuatro, debido a que los pronombres relativos hacen referencia a un nombre ya citado en la oración, llamado antecedente, sin necesidad de repetirlo. Es una forma de subordinación.
4. Verbos: Ocupan la prioridad tres. El verbo es una de las partes más importantes de una oración, también es un candidato a estar precedido de una pausa cuando va precedido de un nombre, adjetivo o cualquier otra categoría gramatical con la cual se complementa mutuamente. El verbo define una pausa más o menos breve.
5. Preposiciones: Ocupan la prioridad dos. Las preposiciones se usan para enlazar palabras cuando van precedidas de una palabra de tipo contenido (sustantivos, adjetivos, etc.), generalmente se separan con una pausa breve.
6. Palabras de tipo función: Se llaman así al conjunto de palabras que pertenecen a las siguientes categorías: artículos, la mayoría de las preposiciones y conjunciones. Ocupan la prioridad más baja. Además de otras restricciones, se inserta pausa en su ubicación cuando van precedidas de una palabra de tipo contenido.

Cada una de estas categorías contemplan una serie de excepciones adicionales a las mencionadas, que impiden la introducción de pausa. Sólo en el caso que no se haya podido introducir pausa para una categoría gramatical, se pasa a la siguiente categoría de prioridad más baja.

La inserción de pausas se repite hasta que todos los grupos entonativos tengan una longitud inferior o igual a 16 sílabas.

En la tabla 2.3.1.2.2, se listan los códigos de las pausas y su correspondiente duración por omisión que tienen en el sistema.

Luego de la inserción de pausas, el sistema corrige la acentuación fonética de las palabras de acuerdo al siguiente criterio prosódico: las palabras átonas que se encuentran delante de una pausa no se pronuncian realmente como átonas. Por eso, siempre que se encuentra una pausa se verifica si la palabra anterior es átona o tiene acentuación ambigua, si lo es, se acentúa fonéticamente y se marca como tónica.

2.2.1.6 Conversión de Grafemas a Unidades Básicas

Este módulo se encarga de transcribir la representación en grafemas (letras) de la frase, a su equivalente en alófonos, y a partir de estas, en una segunda etapa se obtiene la secuencia de unidades básicas que se corresponden al inventario de voces pregrabadas que usa el sintetizador de voz.

Para reducir la complejidad en las reglas de transcripción de grafemas a unidades básicas, se ha desarrollado un procedimiento de análisis de dos fases: En la primera fase se obtiene la representación alofónica de los grafemas, que a su vez comprende dos subtareas: la conversión a alófonos propiamente dicha y la fonetización sintáctica; para luego en la segunda fase realizar la transcripción a unidades básicas, que en nuestro caso se conocen como “demisílabas”.

1. Transcripción de Grafemas a Alófonos: Los algoritmos para transformar un texto de entrada representado mediante caracteres grafemáticos a su representación en símbolos fonéticos, se pueden agrupar en tres grandes grupos por la forma en que realizan la transcripción:
 - La transformación se realiza mediante un conjunto de reglas.
 - La transformación se realiza mediante una tabla de correspondencia de un léxico transcrito fonéticamente.
 - Haciendo uso de sistemas de fonetización basados en el aprendizaje inducido, capaces de calcular una transcripción fonética a partir de un conjunto de datos previamente procesados.

Debido a que la ortografía española tiene un carácter “fonémico”, es decir, tiene la característica de que cada fonema se representa con un único

grafema y cada grafema representa un único fonema, con contadas excepciones. Se decidió usar en este trabajo un conjunto de reglas de transcripción.

Para realizar una correcta transcripción, las reglas definidas toman en cuenta la siguiente información:

- Caracteres (letras) componentes de la frase.
- Límites entre palabras dentro de la frase.
- Límites entre sílabas dentro de las palabras.
- La acentuación fonética dentro de las palabras.
- La división silábica de las palabras.
- La localización de las pausas.

La clasificación fonética de los sonidos y la de sus variedades alofónicas varían en el castellano dependiendo del autor. En nuestro trabajo se ha considerado una clasificación apropiada para un conversor de texto a voz por el método de concatenación de unidades, que en muchos casos puede diferir de los estudios fonológicos tradicionales. La explicación y justificación de la elección de los alófonos usados por el sistema se encuentra en detalle en [Mateo 2000].

La construcción de las reglas finales de transcripción alofónica, se realizó basado en la revisión bibliográfica de [Navarro 1972], [Mateo 2000], [Ríos 1999] y [López 1993], obteniéndose un conjunto de 97 reglas de transcripción de grafemas a alófonos, divididos en dos grupos: Transcripción alofónica y fonetización sintáctica. Incrementándose el número de reglas considerablemente con respecto a la versión anterior del sistema que usa un conjunto de 28 reglas. Este incremento en el número de reglas asegura una transcripción grafema-alófono más confiable.

En el Anexo D, se listan los dos conjuntos de reglas usadas en la transcripción grafema-alófono y la fonetización sintáctica.

- a. Transcripción grafema-alófono(64 reglas): Este conjunto de reglas interpreta el conjunto de signos del alfabeto español, de modo que al aplicarlas se obtenga una representación de las palabras silabeadas como una cadena de alófonos. Tomando en cuenta la variación contextual de cada grafema para realizar esta tarea, se usan reglas llamadas dependientes del contexto, que pueden expresar distintos tipos de relaciones entre la entrada y la salida:
1. Correspondencia biunívoca: Una letra se transcribe como un alófono ($p \rightarrow p1$).
 2. Correspondencia dos a uno: dos letras adyacentes se transcriben como un fonema ($ch \rightarrow c1$).
 3. Correspondencia uno a dos: una letra se transcribe como una secuencia de dos fonemas($x \rightarrow c2s1$).
 4. Correspondencia uno a cero: una letra no tiene transcripción fonética ($h \rightarrow \text{""}$).

El formalismo habitual de las reglas de transcripción sigue el modelo de Chomsky, son reglas de reescritura contextuales, con el siguiente formato:

Foco \rightarrow (contexto izquierdo) (contexto derecho) / cambio, donde:

El foco y los contextos pueden ser unidades o grupos de grafemas, rasgos de fonemas, o elemento vacío. Mientras que el cambio puede ser uno o más fonemas.

Adicionalmente de que los contextos izquierdos o derechos al foco se refieran a los grafemas, también toman en cuenta las siguientes posiciones: Posición de la pausa (0), división silábica (-), división entre palabras (#).

La regla asigna un cambio estructural a un foco a condición de que este vaya precedido por el "contexto izquierdo" y seguido del

"contexto derecho". Así los grafemas se relacionan con los fonemas a través de las reglas.

- b. Fonetización sintáctica(33 reglas): Corrige la transcripción alofónica hecha en el paso anterior en los alófonos ubicados en los límites entre palabras. Esta técnica es conocida como fonetización de textos, y comprende dos operaciones: La resilabación obligatoria en los límites entre palabras (cuando se dé el contexto adecuado) y la variación alofónica de los segmentos que allí se encuentren. En el presente trabajo, se toman en cuenta los siguientes tipos de combinaciones segmentales en los límites entre palabras:

1. Secuencias [C#V]: Se presenta cuando el fin de la palabra precedente es una consonante y el inicio de la palabra siguiente es una vocal. De la definición encontrada en [Navarro 1972], "tanto en la palabra aislada como el discurso toda consonante forma sílaba con la vocal que le sigue". Se debe realizar la resilabación correspondiente. Y como la consonante pasa de coda(final) silábica a ataque(inicio) silábico se debe realizar su variación alofónica correspondiente. Por ejemplo:

Palabra en grafemas: "club andaluz"

Palabra en alófonos: "c2l1u2**b3#a3**n2-d1a1-l1u2s1"

Luego de la fonetización sintáctica: "c2l1u2# **b2a3**n2-d1a1-l1u2s1"

2. Secuencias [C#C]: cuando el fin de la palabra precedente es una consonante y el inicio de la palabra siguiente también es una consonante. Se toman en cuenta los casos:
 - Oclusivas sordas [p1,t1,k1]#Consonante sonora: Los fonemas oclusivos sordos[p1,t1,k1] se realizan como sonidos oclusivos sordos en posición final de palabra, pero en el discurso se sonorizan ante consonante sonora. En este caso no se realiza la resilabación, por tratarse de dos

consonantes consecutivas, pero se realiza la variación alofónica indicada. Por ejemplo:

Palabra en grafemas: “eslip naranja”

Palabra en alófonos: “e3s1-l1i2p1#n1a1-r1a2n2-x1a1”

Luego de la fonetización sintáctica: “e3s1-l1i2**b3**#n1a1-r1a2n2-x1a1”

- Oclusivas sordas [p1,t1,k1] o variante2[c2]#Consonante sonora [p1, t1, k1, b1, b2, d1, d2, g1, g2]: En este caso las oclusivas sordas [p1, t1, k1] o la variante 2 [c2] se eliden, es decir, se borran de la palabra. Por ejemplo:

Palabra en grafemas: “bistec gustoso”

Palabra en alófonos: “b1i1s1-t1e2**c2**#g2u1s1-t1o2-s1o1”

Luego de la fonetización sintáctica: “b1i1s1-t1e2#g2u1s1-t1o2-s1o1”

- Las fricativas se eliden cuando van seguidas de sí mismas, f1#f1, z1#z1, s1#s1, x1#x1. Por ejemplo:

Palabra en grafemas: “las sábanas”

Palabra en alófonos: “l1a1**s1**#s1a2-b2a1-n1a1s1”

Luego de la fonetización sintáctica: “l1a1#s1a2-b2a1-n1a1s1”

- Vibrante simple [r1]#Vibrante múltiple[r2]: En este caso se elide la vibrante simple [r1]. Por ejemplo:

Palabra en grafemas: “mar rugiente”

Palabra en alófonos: “m1a1**r1**#r2u1-x1i1e2n2-t1e1”

Luego de la fonetización sintáctica: “m1a2#r2u1-x1i1e2n2-t1e1”

- Variantes [g3]#Consonante sorda [p1,t1,k1,c1,f1,z1,s1, x1]:
Los fonemas oclusivos sonoros se realizan como alófonos variantes en posición final de palabra, pero ante consonante sorda en el discurso se realizan como variantes ensordecidas. Por ejemplo:

Palabra en grafemas: “bulldog perdiguero”

Palabra en alófonos: “b1u1l1-d1o2g3#p1e1r1-d2i1-g2e2-r1o1”

Luego de la fonetización sintáctica: “b1u1l1-d1o2c2#p1e1r1-d2i1-g2e2-r1o1”

Las reglas utilizadas en la fonetización sintáctica tienen el mismo formato que en el caso anterior, pero con la diferencia que el foco, cambio y los contextos solo pueden ser alófonos y vacío, además de los separadores de palabras y sílabas.

2. Transcripción a Unidades Básicas: En esta etapa la secuencia de alófonos obtenida en la etapa anterior, se pasa a Demisílabas. El procedimiento usado en esta etapa, es la misma que se usa en [Mateo 2000], con las modificaciones pertinentes a las nuevas necesidades; por ejemplo: el manejo de pausas como parte del párrafo a procesar. En la versión anterior las pausas no se consideraban como parte del párrafo o frase que se leía, sino que se iba leyendo porciones de texto limitadas por cualquier signo ortográfico, no existía el concepto de signos ortográficos delimitadores de frase.

2.2.1.7 Procesado Prosódico

Una vez obtenida la secuencia de unidades básicas (Demisílabas), es necesario añadirles más información para determinar la manera en que deben ser pronunciadas. La tarea de este último bloque del procesamiento lingüístico-prosódico, es justamente generar esta información llamada prosodia. La cual en nuestro caso, se divide en dos tareas uno por cada parámetro tomado en cuenta: la duración segmental y el contorno de

frecuencia fundamental. Para lo cual se hace uso de la información suprasegmental y segmental, obtenidas en las tareas anteriores.

En [Navarro 1948] el autor menciona “La frase no ofrece sentido concreto mientras no recibe una forma tónica determinada”, es decir, los cambios melódicos, las modificaciones en la duración e intensidad, y la mayor o menor duración de las pausas, conque se pronuncia o lee una frase, sirven lingüísticamente para precisar el sentido e intención de las palabras. Razón por la cual la existencia de este bloque en sistemas de conversión de texto a voz es indispensable si se le desea dar naturalidad a la lectura de textos.

Según [López 1993] el punto de partida en el estudio de la entonación hay que situarlo en su dificultad, ya que depende del punto dentro del proceso de comunicación lector-oyente que se considere:

1. En el habla, es decir en la propia señal de voz donde la entonación se manifiesta mediante la variación temporal a lo largo de la elocución de la frecuencia fundamental, duración y energía.
2. En la sensación de tono, duración y volumen que tiene un oyente al excitar el oído humano los parámetros físicos (frecuencia fundamental y duración).
3. En la interpretación lingüística e incluso extralingüística que el oyente hace del mensaje.

Las variaciones de los correlatos físicos que caracterizan la entonación son captadas por el sistema auditivo y transformadas adecuadamente, llegan al cerebro donde se interpretan y decodifican. El resultado de este proceso de decodificación puede ser de tipo lingüístico, de tipo social (acento del locutor) o de tipo expresivo (situación o estado de ánimo del hablante). En nuestro caso nos interesa sobretodo modelar la función lingüística que nos impondrá su acento particular.

1. Duración: Por duración se entiende como el intervalo de tiempo que mantiene un alófono en el discurso, y los silencios correspondientes a cada tipo de pausa. El modelado de este parámetro es bastante complejo

pues depende de factores segmentales (tipo de sonido que se considera) y de factores suprasegmentales (del sonido en comparación con otros sonidos del discurso). Es por eso, que para el estudio de las duraciones se han usado muchas técnicas como análisis de factores, análisis ordinal, modelos estadísticos o de inducción basados en árboles de clasificación. Y se han creado modelos multiplicativos o sumas de productos en el caso más general.

En nuestro caso, el sistema contempla un modelo multiplicativo para calcular la duración de cada alófono: a cada alófono se le asigna una duración base o típica, que luego se multiplica con factores dependientes de: Los alófonos de su entorno, su posición respecto al acento y la posición del alófono dentro del grupo fónico.

Debido a que el sistema usa las demisílabas como unidades básicas para almacenar las muestras de voz a concatenar, y considerando la estructura de las demisílabas mostrada en la Tabla 2.3.1.7.1, en la que se observa que están formadas por uno o más alófonos (o semialófonos). Se decidió calcular directamente la duración de cada demisílaba sobre la base de la duración de los alófonos que lo conforman.

TABLA 2.3.1.7.1 ESTRUCTURA DE SÍLABAS Y DEMISÍLABAS USADAS EN EL SISTEMA

Sílaba	Demisílabas
V1	V1
V1C1	V1+v1C1
V1V2	V1V2
V1V2C1	V1V2+v2C1
C1V1	C1v1+V1
C1V1C2	C1v1+V1+v1C2
C1V1C2C3	C1v1+V1+v1C2C3
C1V1V2	C1v1+V1V2
C1V1V2C2	C1v1+V1V2+v2C2
C1V1V2V3	C1v1+V1V2V3
C1V1V2V3C2	C1v1+V1V2V3+v3C2
C1C2V1	C1C2v1+V1
C1C2V1C2	C1C2v1+V1+v1C2
C1C2V1C3C4	C1C2v1+V1+v1C3C4
C1C2V1V2	C1C2v1+V1V2
C1C2V1V2C3	C1C2v1+V1V2+v2C3
C1C2V1V2V3C3	C1C2v1+V1V2V3+v3C3

Leyenda:

Cn : Consonante n.

Vn: Vocal completa n.

vn: Parte de una vocal en la demisílaba compuesta

n: Índice identificador [1..4]

+ : Separador de demisílaba.

Para almacenar la duración de cada demisílaba de la frase se cuenta con un arreglo del tamaño del número de demisílabas que tiene la frase. En nuestro caso, la cadena de demisílabas de la frase contiene los códigos de las pausas que pudiera hacerse en la frase, por lo que una pausa se cuenta como si fuera una demisílaba más.

Existen tres grupos de reglas que modelan las duraciones correspondientes a las pausas, vocales y consonantes del español.

- a. Duración de las pausas: La tabla correspondiente a las pausas almacena la duración de estas, según lo mostrado en la Tabla 2.3.1.2.2. Cuando el sistema encuentra un código de pausa en la cadena de demisílabas de la frase, recupera la duración de la pausa correspondiente de la tabla y la coloca en el arreglo de duraciones en la posición donde encontró el código de pausa.
- b. Duración de las vocales: Para calcular la duración de las vocales se sigue un procedimiento multiplicativo. Para lo cual se ha tenido en cuenta los estudios desarrollados en [Navarro 1972, López 1993], y a partir de los cuales se definió la duración base de cada una de las vocales y una serie de reglas que permiten calcular la duración aproximada de las vocales tomando en cuenta los factores dependientes del contexto mencionados anteriormente. Los criterios tomados para la construcción de las reglas son los siguientes:
 1. Se toma como duración base de las vocales, la correspondiente a las vocales breves acentuadas, estos valores expresados en milisegundos (ms) usados como configuración inicial en el sistema son: [á = 90, é = 82, í = 80, ó = 85, ú = 95]. Los

valores definitivos se obtuvieron haciendo una serie de pruebas con el sistema para diferentes valores asignadas a cada vocal.

Para determinar los valores finales de duración usados en el sistema, se escogió una serie de palabras y frases pequeñas donde aparecían los alófonos en estudio en diferentes ubicaciones e intensidades (con acento y sin acento). Inicialmente se les dio las duraciones indicadas en la bibliografía pero el resultado no fue satisfactorio las sílabas y palabras se entrecortaban y no se entendía lo que el conversor leía, por lo que se fue modificando la duración de las vocales hasta encontrar el juego de valores indicados porque con estas se presentaba una menor distorsión de la inteligibilidad y naturalidad de la lectura. La determinación de los valores correctos se hizo de la siguiente manera, se escuchó la lectura de textos del sistema y se tomaron los valores de duración que mejor inteligibilidad producían. Los resultados experimentales demuestran que los valores tomados fueron los adecuados.

Estos valores finales escogidos son muy próximos a los valores que se obtuvieron en los estudios desarrollados en [Navarro 1972].

2. Es breve toda vocal inacentuada.
3. Los valores de las vocales acentuadas semilargas expresadas en milisegundos (ms) se obtienen mediante la multiplicación de los valores de las vocales breves acentuadas por un factor dependiendo de cada vocal. Los factores usados por cada vocal son: [á = 1.30, é = 1.35, í = 1.30, ó = 1.25, ú = 1.30]
4. La vocal acentuada duplica su duración, con relación a la breve, en las palabras agudas, siempre que estas no terminen en consonante “n” ni “l”.
5. La vocal acentuada es semilarga en palabras agudas terminadas en “n” o “l”, y en sílaba abierta de palabras llanas.

6. Una vocal débil (i, u) acentuada ve incrementada su duración en un 25% si va junto a una vocal fuerte.
7. Si la vocal es la última letra de una sílaba antes de una pausa, se incrementa su duración en un 10%.
8. La vocal inicial de un grupo entonativo alarga su duración un 20%.
9. Toda vocal que va precedida de una consonante sorda ve reducida su duración en un 10%.
10. La presencia de otra vocal junto a la vocal a tratar hace disminuir la duración de la vocal en un 10%.
11. Una consonante líquida o nasal (m, n, l, ll) adyacente hace disminuir la duración de la vocal en un 10%.
12. Si es una palabra compuesta de una sola vocal, esta vocal incrementa su duración en un 40%.
13. Si es una palabra que empieza con una vocal inacentuada, la vocal incrementa su duración en un 20%.

Una vez calculada la duración final de una vocal, y teniendo en cuenta la estructura de las sílabas y demisílabas mostrada en la Tabla 2.3.1.7.1; se distribuye entre la demisílaba donde está la vocal actual y las demisílabas adyacentes, tal como se ilustra en la Tabla 2.3.1.7.2. La primera columna muestra partes de sílabas donde existe una vocal, y la segunda, la división en demisílabas de estas partes.

TABLA 2.3.1.7.2 DISTRIBUCIÓN DE LA DURACIÓN DE UNA VOCAL ENTRE LAS DEMISÍLABAS QUE LA CONTIENEN.

Parte de Sílabla	Estructura Demisílabla	Distribución de Duración
CVC	Cv + V + vC	Se divide la duración de la vocal entre tres. Luego, se suma esta cantidad a la duración de la demisílabla anterior, actual y posterior.
CV	Cv + V	Se divide la duración de la vocal entre dos. Luego, se suma esta cantidad a la duración de la demisílabla anterior y la actual.
VC	V + vC	Se divide la vocal entre dos. Pero, la cantidad hallada se le agrega a la duración de la demisílabla actual y posterior.
V	V	La duración calculada completa, se suma a la demisílabla actual.

- c. Duración de las Consonantes: Para calcular la duración de las consonantes, al igual que en el caso de las vocales, se usa un enfoque multiplicativo pero complementado con un conjunto de reglas que asignan la duración de ciertas consonantes directamente. Estas reglas se obtuvieron del estudio realizado en [Laura 1999].

En ese estudio se toma en cuenta la influencia de tres variables independientes sobre la duración de las consonantes en la lectura:

1. El acento (+/- acentuada).
2. La posición de la consonante en la sílaba(+/- inicio de sílaba).
3. La posición de la consonante en el grupo fónico (+/- prepausal).

El resumen de los resultados de este estudio se muestra en el Anexo E, y es la base de las reglas de duración consonántica usadas en este trabajo. Como se observa en esta tabla no se ha tomado en cuenta todas las consonantes del castellano, sino sólo las que se indican a continuación:

1. Fricativas sordas: labiodental [f1], interdental [z1], alveolar [s1] y velar [x1].
2. Fricativa palatal sonora [y1].

3. Africada palatal sorda [c1].
4. Nasaes: bilabial [m1], alveolar [n1], y palatal [n3].
5. Laterales: alveolar [l1] y palatal [l2].
6. Vibrantes: simple [r1] y múltiple [r2].

Para el resto de las consonantes se usan los estudios realizados por [Navarro 1972, López 1993] para definir los criterios para el modelo multiplicativo de las consonantes, y son los que se listan a continuación:

1. Se toma como duración base de las consonantes la duración promedio de cada una de ellas. Estos valores se muestran en la Tabla 2.3.1.7.3.

TABLA 2.3.1.7.3. DURACIÓN BASE DE LAS CONSONANTES

Alófono	Duración(ms)	Alófono	Duración(ms)
p1	70	s1	108
b1	80	y1	60
b2	72	x1	111
b3	75	c1	117
t1	68	c2	112
d1	85	m1	74
d2	67	n1	78
d3	81	n2	80
k1	75	n3	90
g1	82	l1	59
g2	60	l2	67
g3	90	r1	46
f1	70	r2	96
z1	104		

2. El efecto de los grupos consonánticos sobre las consonantes que lo forman es de reducción de su duración en un 20%, excepto el alófono "r" simple que en un grupo consonántico aumenta su duración un 10%.
3. Una consonante en posición intervocálica aumenta su duración un 15%, excepto el alófono "r" simple que disminuye su duración en un 20%.
4. La consonante en sílaba final de frase aumenta su duración en un 20%.

5. La consonante en sílaba final, pero no en posición final absoluta (no antes de pausa), produce un aumento del 10%.
6. La consonante en sílaba acentuada aumenta su duración en un 10%

Los criterios 4, 5 y 6 sólo se aplican a las consonantes no consideradas en las reglas listadas en el Anexo E.

En el caso de los alófonos de las consonantes, su duración calculada se asigna a la demisílaba a la cual pertenece sin hacer ninguna división, porque en las demisílabas que utiliza el sistema las consonantes no se dividen.

2. Frecuencia Fundamental: Cuando se lee, la secuencia de fonemas y sílabas de la frase van acompañadas por una especie de melodía ejecutadas por las variaciones de la voz, las cuales dependen de la tensión de las cuerdas vocales y de los cambios de frecuencia de vibración de estas (frecuencia fundamental). Esta melodía con que “cantamos” o “entonamos” lo que decimos se llama entonación y sirve para unificar como un todo cada grupo fónico perteneciente a una frase, independientemente de su estructura interna en sílabas, fonemas y signos. Por eso se dice, que la entonación es un proceso suprasegmental, porque ocurre paralelamente a los fonemas segmentales.

La entonación es la que más influye en la calidad y naturalidad de la voz producida porque transmite información especial del mensaje: le indica al receptor si el enunciado es una pregunta, una enunciación, si manifiesta dudas, ira, temor, cariño, etc. Por ejemplo, si se comparan estas dos expresiones:

- Todavía no ha llegado el tren.
- ¿Todavía no ha llegado el tren?

Se observa que ambas tienen una misma secuencia de fonemas, de sílabas y palabras; y sin embargo, todo hablante reconoce que no significan lo mismo. Esto debido a que algo las distingue fonéticamente; lo

cual se señala en la ortografía con los signos de interrogación, y nuestro oído percibe las diferencias de las melodías con que se entonan ambos enunciados.

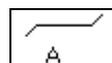
Los principales factores que pueden influir en la entonación de una frase son:

- Situación del acento temático o foco de atención del enunciado.
- Ubicación de los acentos fonéticos.
- Longitud de la frase (Influye en la división en grupos fónicos).
- Velocidad de elocución.
- Relaciones sintácticas entre grupos fónicos (Tipos de grupos fónicos).

Se ha dicho que la frase, según la variación de su entonación esta compuesta por grupos fónicos. El grupo fónico, en nuestro caso, es la unidad que determina la estructura musical propia de cada frase. Una frase puede tener uno o más grupos fónicos, y se identifican porque vienen separadas entre pausas.

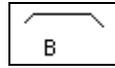
La entonación del grupo fónico varía constantemente según las circunstancias particulares de cada caso; pero luego de revisar diversas publicaciones referentes al tema para esta Tesis se ha definido seis patrones o modelos entonativos que se implementan en el sistema, los cuales pasamos a describir brevemente:

- Patrón A: Empieza con un tono grave que sigue con un ascenso inicial hasta la primera sílaba tónica. Continúa con un tono normal hasta la última sílaba tónica. Y a partir de allí ocurre un punto de inflexión para terminar con un ascenso del tono (más agudo).

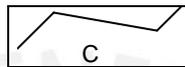


- Patrón B: Igual que el anterior, empieza con un tono grave, sigue con un ascenso inicial hasta la primera sílaba tónica. Continúa con un

tono normal hasta la última sílaba tónica. A partir de allí desciende hasta el final.



- Patrón C: Empieza con un tono grave, asciende hasta la primera sílaba acentuada por encima del tono normal. Desciende después gradualmente hasta la penúltima sílaba, y vuelve a elevarse sobre la última sílaba.



- Patrón D: Del tono grave asciende hasta la primera sílaba acentuada, un poco más alto del tono normal. Luego desciende en una pendiente suave hasta la última sílaba acentuada, a partir del cual vuelve a descender ligeramente.



- Patrón E: La voz se eleva algo menos que en el patrón C hasta la primera sílaba acentuada, desciende con una pendiente suave hasta la última sílaba tónica, en esta sílaba realiza un nuevo ascenso hasta más arriba del tono normal, para luego, volver a descender durante la sílaba o sílabas inacentuadas con que la frase termina; este movimiento circunflejo final se produce dentro de la última sílaba acentuada cuando no hay más sílabas inacentuadas después de esta.



- Patrón F: Este contorno entonativo es similar al definido para el patrón D, con la diferencia que el primer ascenso se realiza hasta el centro de la primera sílaba tónica.

Resumiendo podemos decir, que para la asignación de valores de frecuencia fundamental, cada grupo fónico se divide en tres zonas, tratadas independientemente:

- Desde el principio del grupo fónico hasta la primera sílaba tónica, excluyendo esta en muchos casos.

- Desde la primera sílaba tónica hasta la penúltima sílaba tónica, ambas incluidas.
- Desde la sílaba siguiente a la penúltima tónica, hasta el final del grupo fónico.

La primera tarea que realiza el sistema para generar la entonación de una frase, es intentar descubrir el tipo de frase que se está procesando, para que a partir de esta identificación se pueda asignar el o los patrones entonativos que le corresponden a cada grupo fónico de la frase. Los tipos de frase tomados en cuenta son los que se muestran en la Tabla 2.3.1.7.4, y para su identificación se toman en cuenta los signos ortográficos, palabras claves dentro de la frase como los pronombres interrogativos, conjunciones, y la cantidad de grupos fónicos que tiene la frase.

TABLA 2.3.1.7.4 TIPOS DE FRASE

Tipo de Frase	Descripción
ENUNCIACIÓN	Oración enunciativa de un sólo grupo fónico
ENUNCIACIONCOMPUESTA	Oración enunciativa con dos o más grupos fónicos
ENUNCIACIONPARENTESIS	Oración enunciativa con paréntesis interior
ENUNCIACIONENUMERATIVA1	Oración enunciativa enumerativa, TIPO 1
ENUNCIACIONENUMERATIVA2	Oración enunciativa enumerativa, TIPO 2
PREGUNTAABSOLUTA	Oración interrogativa absoluta con un sólo grupo fónico
PREGUNTAABSOLUTACOMP	Oración absoluta con dos o más grupos fónicos
PREGUNTACONJUNCION	Oración interrogativa con conjunción (o,u)
PREGUNTARELATIVA	Oración interrogativa relativa con un sólo grupo fónico
PREGUNTARELATIVACOMP	Oración interrogativa relativa con dos o más grupos fónicos
EXCLAMATIVA	Oración exclamativa que no tiene pronombre interrogativo
EXCLAMATIVAPRONOMINAL	Oración exclamativa con pronombre interrogativo

En este trabajo se asume, que las voces pregrabadas que sirven para obtener los parámetros del filtro de predicción lineal y que se usan para generar la voz sintética tienen una frecuencia fundamental constante neutra, es decir, no tienen variaciones de entonación.

Esto nos permite manejar un modelo proporcional para generar el contorno de frecuencia fundamental. Asumiendo que las voces pregrabadas tienen el tono normal, es decir, como uno, y que cualquier

variación de la frecuencia se hace colocando valores positivos mayores (más agudo) o menores (más grave) a uno.

El módulo de procesamiento de señal necesita recibir información sobre el tono con un muestreo uniforme, razón por la cual se genera un valor por cada trama de síntesis de voz, lo que supone calcular un valor cada 8 ms. que es el tamaño de trama utilizado por el sistema. La generación de estos valores se hace en dos pasos:

- En primer lugar, se marcan los puntos de inflexión correspondientes a los patrones entonativos descritos anteriormente de acuerdo al tipo de frase y el número de grupos fónicos que esta tiene. El formato de contorno de frecuencia fundamental para cada tipo de frase es la siguiente:
 1. Enunciación: Llamada también afirmación simple, solo tienen un grupo fónico de tipo “B”.
 2. Enunciación compuesta: Es una afirmación con dos o más grupos fónicos. Los grupos anteriores al último son de tipo “A” y el último grupo fónico es de tipo “B”.
 3. Enunciación con paréntesis: Son enunciaciones que tienen una cláusula encerrada entre paréntesis, llamada grupo parentético. Los grupos anteriores al grupo parentético son del tipo “A”. El grupo parentético en sí es de la forma “B” pero multiplicada por un factor menor a 1, debido a que el tono normal de las cláusulas entre paréntesis es más bajo que el resto de la línea melódica. Y los grupos siguientes al parentético son del tipo “B”.
 4. Enunciación enumerativa(TIPO 1): Oración enumerativa cuyo último grupo fónico está precedida por una conjunción (y, e). Se caracterizan porque los grupos anteriores al antepenúltimo grupo y el último grupo son del tipo “B”. Siendo el penúltimo grupo de tipo “A”.

5. Enunciación enumerativa(TIPO 2): En este caso todos los grupos fónicos de la enumeración están separados por comas incluyendo el último. Es por eso que también todos sus grupos fónicos son de tipo “B”.
6. Pregunta absoluta: Se refiere a aquellas preguntas que esperan una respuesta del tipo “sí” o “no”. Tienen un solo grupo fónico de tipo “C”. En forma general se puede decir, que el tono de las preguntas interrogativas es un poco más alta que en las enunciativas.
7. Pregunta absoluta compuesta: Tienen más de un grupo fónico. Los grupos anteriores son de tipo “D”, y el último es de tipo “C”.
8. Pregunta con conjunción: Se refiere a las preguntas que tienen grupos fónicos (generalmente dos) unidos por una conjunción (o, u), que generan una interrogación de tipo contrastiva. En este caso los grupos anteriores al último son de tipo “C”, mientras que el último es de tipo “D”.
9. Pregunta relativa: Se trata de preguntas que buscan confirmar algo que no se sabe con seguridad pero que de alguna manera se intuye. En este caso se trata de preguntas con un solo grupo fónico de tipo “E”.
10. Pregunta relativa compuesta: Tienen dos o más grupos fónicos. Los grupos anteriores al último son de tipo “D”, y el último grupo es de tipo “E”.
11. Exclamativa: Son frases exclamativas que no tienen pronombre interrogativo. Sólo tienen un grupo fónico de tipo “E”.
12. Exclamativa Pronominal: Oración exclamativa con pronombre interrogativo. Tiene un grupo fónico de tipo “F”.

En cada punto de inflexión se colocan valores predeterminados, de acuerdo a lo establecido para el tipo de grupo fónico.

- En segundo lugar, se interpolan los puntos de inflexión marcadas en la etapa anterior para generar puntos de frecuencia fundamental correspondientes a cada trama de voz. El algoritmo utilizado es el mismo que se usa para unir dos puntos de una recta (Interpolación lineal).

Al finalizar este proceso, el sistema tiene generado las demisílabas de la frase a sintetizar, un arreglo de duraciones correspondiente a cada demisílaba, y otro arreglo conteniendo los factores del contorno de frecuencia fundamental. Toda esta información se envía al bloque sintetizador de voz encargado de generar la voz correspondiente a la frase que se está procesando.

2.2.2 Procesado Acústico

El sintetizador desarrollado como parte del trabajo de Tesis [Mateo 2000], es la base del procesado acústico de este trabajo. El sintetizador usa el método de concatenación de unidades que se caracteriza por tener un conjunto de segmentos pequeños de voz tomados de un locutor llamado donante y que se van concatenando para formar el discurso deseado. Las unidades básicas que usa el sistema son las demisílabas, que representan las partes inicial y final de las sílabas. Las ventajas más importantes de trabajar con demisílabas son:

- Se requiere una cantidad relativamente pequeña de unidades básicas, actualmente el sistema cuenta con 540 demisílabas. La versión anterior tenía 530 demisílabas debido a que fue desarrollado para emular a un hablante criollo, por lo que no se consideraba al alófono lateral palatal sonoro l2 (ll), por ejemplo, un criollo pronuncia la palabra “pollo” como “poyo”.
- Al usar demisílabas se requiere menos puntos de concatenación.
- La demisílabas toman en cuenta la mayoría de las transiciones, y por consiguiente, un gran número de efectos de coarticulación.

- También, toman en cuenta una gran cantidad de variaciones alofónicas debido a la separación de los alófonos de las consonantes en iniciales y finales.

Entre las desventajas de usar demisílabas, encontramos:

- El requerimiento de memoria para almacenar las muestras de voz de todas la demisílabas es todavía alto, pero tolerable. En nuestro caso es un archivo de 8 MB aproximadamente de tamaño en disco, que lo hace poco transportable pero que con el avance de la tecnología de almacenamiento y el abaratamiento de las memorias está dejando de ser un problema crítico. Además, de que usando algoritmos de codificación más eficientes se puede reducir el tamaño actual del archivo considerablemente.
- Comparado con los fonemas y dífonemas, el número exacto de demisílabas de un idioma no puede ser definido.
- Con un sistema basado solamente en demisílabas, todas las palabras no pueden ser sintetizadas adecuadamente, por eso se requiere usar otras unidades de menor longitud como los fonemas de las vocales, diptongos y triptongos.

El algoritmo del método de síntesis de voz usado es el mismo que se utiliza para la codificación de las unidades básicas que es predicción lineal multipulso (MPLPC) con 20 coeficientes.

El procesado acústico se divide en dos partes: La primera en que se obtienen (codifican) las unidades de voz que se concatenarán para generar la lectura, y la segunda, donde se hace la concatenación mediante síntesis de las muestras de voz.

2.2.2.1 Codificación del Inventario de Voces (Demisílabas)

La calidad del inventario de voces juega un papel muy importante en la calidad de la voz que generan los sistemas conversores de texto a voz. Por esta razón dentro de lo limitado del tiempo y recursos que se tenía para

desarrollar el trabajo, se tuvo especial cuidado en la obtención de las palabras que se usaron para obtener las demisílabas.

Para la obtención del juego final del inventario de voces (demisílabas, vocales, diptongos y triptongos) se procedió de la siguiente manera:

- Se hizo una grabación inicial de las palabras de donde se obtuvieron las 540 unidades de voz, esta tarea se hizo haciendo uso de un aplicativo que permite crear y editar archivos de sonido. Cada palabra fue grabada con una duración de 2 segundos y usando una de las funcionalidades del aplicativo se les pasó por un filtro para eliminar el ruido del ambiente no deseado.
- Para calcular la duración que cada unidad de voz del inventario debería tener se usó el resultado dado por el módulo prosódico del sistema conversor texto voz. Teniendo la duración de cada demisílaba se procedió a cortar las palabras grabadas en la etapa anterior, editando la duración de la voz ya sea eliminando o agregando muestras para ajustarla a la duración calculada por el sistema.
- Una vez obtenidas las muestras de voz, se ejecutó el proceso de análisis de voz desarrollado como parte del trabajo de tesis [Mateo 2000] con lo cual se obtuvo el archivo de parámetros LPC deseado. Las pruebas realizadas usando este nuevo archivo en el sistema no dieron los resultados esperados, debido a que la duración de las vocales y consonantes que usaba el sistema para calcular la duración de las unidades básicas era la que se encontró en los textos referentes al tema como [Navarro 1972] y [Laura 1999], sin considerar que la duración de los alófonos varía dependiendo del locutor.
- Para solucionar el problema anterior y con la finalidad de obtener la duración aproximada de las vocales y consonantes correspondientes al donante de la voz se grabó una serie de palabras adicionales a las usadas en la obtención de las demisílabas. Luego usando el editor de sonidos se obtuvo la duración de cada alófono. En la mayoría de los casos las duraciones obtenidas coinciden o son muy próximas a los

valores registrados en los estudios desarrollados en [Navarro 1972] y [Laura 1999].

- Se procedió a cargar las duraciones obtenidas en las tablas de duraciones del sistema. Usando esta información se volvió a generar la duración que debería de tener cada demisílaba y se procedió a cortarlas de las palabras. Luego, se generó el nuevo archivo de parámetros de coeficientes LPC. Al hacer las pruebas del sistema usando el nuevo archivo LPC se obtuvieron mejores resultados, lográndose mejorar la naturalidad de la lectura.

2.2.2.2 Síntesis de Voz

El objetivo de este módulo consiste en generar la voz correspondiente a la cadena de demisílabas que recibe del bloque lingüístico, aplicando las modificaciones de la duración segmental y del contorno de frecuencia fundamental calculados en el módulo prosódico. El módulo se puede dividir en tres bloques principales:

1. Obtener las muestras de voz correspondientes a la secuencia de unidades básicas que se quiere sintetizar. En esta etapa el sistema recupera los parámetros de predicción lineal, así como, las ubicaciones y amplitudes de la señal de excitación multipulso correspondientes a cada una de las demisílabas que forman la frase. Con estos parámetros y usando el algoritmo de síntesis descrito en [Mateo 2000] se forman las tramas de la señal de voz correspondiente a cada unidad.
2. Modificar la frecuencia fundamental de la señal correspondiente a la unidad básica que se está procesando. Como resultado del modelado prosódico se obtuvo un arreglo con valores reales positivos que indica el patrón del contorno de frecuencia fundamental de la señal resultante correspondiente a un tipo de frase (enunciación, interrogación, exclamación, etc.), en el sistema desarrollado se tiene un valor para cada 8 ms. que se corresponde con la duración de una trama de la señal de voz usada por el sistema.

En los sistemas de síntesis LPC la modificación de la frecuencia fundamental se logra alterando la distancia entre los pulsos de la señal de excitación periódica. Pero al usarse el multipulso como señal de excitación la modificación del contorno de frecuencia fundamental no es tan directo, por lo que se hace necesario generar algoritmos más complejos que requerirían mayores recursos para su implementación. Por esta razón y con la finalidad de demostrar el uso del arreglo del contorno de frecuencia fundamental. En este bloque se considera a la señal de excitación multipulso como si fuera una secuencia de pulsos con periodo cuasi constante y se modifica la ubicación de la señal de pulsos de acuerdo al factor de frecuencia fundamental que le corresponde a la unidad básica.

3. Modificación de la Duración de la Unidad Básica: Adicionalmente a la frecuencia fundamental el módulo prosódico calcula la duración que debe tener la demisílaba en el contexto que se presenta. Para usar esta información se agregaron funcionalidades al sintetizador de voz desarrollado en [Mateo 2000]. Estas funcionalidades implementan dos conceptos que permiten disminuir y agregar muestras a las señales de voz, estas son la decimación e interpolación lineal.

En la decimación el conjunto original de muestras es reducido en número (diezmado), esto se logra reduciendo la frecuencia de muestreo para lo cual se incrementa el paso del tiempo de T_1 a T_2 de acuerdo a $T_2 = R \cdot T_1$, donde el ratio R es el ratio del tiempo de paso y es mayor que uno. En la interpolación se colocan nuevas muestras entre las originales para lo cual la frecuencia de muestreo es incrementada.

En la decimación e interpolación lineales una simple regla de proporciones es usada para calcular un valor intermedio, $x(k + d)$, entre dos elementos sucesivos, $x(k)$ y $x(k+1)$, de una secuencia.

Para hacer la modificación de la duración de una unidad básica el sistema verifica si la duración calculada y asignada a la demisílaba prosódicamente es mayor que la correspondiente a la muestra de voz

recuperada del archivo LPC, si es así se generan nuevas muestras mediante la interpolación lineal de las existentes. Si por el contrario, la duración asignada es menor que la duración derivada de las muestras de voz se procede a la eliminación de estas muestras para ajustarla a la duración calculada.



3

Análisis y Diseño Orientado a Objetos(UML)**3.1 Metodología de Desarrollo**

En forma resumida, la metodología Orientada a Objetos aplicada en el desarrollo de esta tesis toma en cuenta las siguientes fases:

3.1.1 Definición de Requerimientos

En esta fase se definió las necesidades que debería satisfacer el producto final, así como los objetivos del trabajo. Como producto entregable de esta etapa se tiene el documento “Definición y Requerimientos.Doc” que se muestra en el Diagrama de Casos de Uso de la Figura 3.2.1.1.1.

3.1.2 Análisis de Requerimientos

En esta fase se definió con mayor claridad lo que el sistema final debería hacer. Basándonos en la notación UML en esta etapa se hizo uso de los Diagramas de Casos de Uso y se hizo un bosquejo inicial de los Diagramas de Clase.

3.1.3 Diseño Preliminar

En esta etapa se definió la arquitectura del sistema. Los componentes y paquetes en los que se iba a dividir, y las librerías de las que estaría compuesto el programa, para lo cual se hizo uso de los Diagramas de Paquetes y Componentes. Siguiendo con la notación UML , se usó los Diagramas de Estado y de Secuencias para definir el comportamiento de las interfaces y relaciones entre paquetes, actores, y los objetos de clases.

3.1.4 Diseño Detallado

En esta fase, entramos ya a detallar el comportamiento de las clases mediante la definición de sus métodos, teniendo en cuenta la división en paquetes realizada en el paso anterior y siguiendo siempre los casos de uso. También, en esta etapa se definió el diseño de los archivos de datos a usar y se hizo el diseño de la interfaz de usuario. Como documentos entregables de esta etapa se tiene: Diagrama de Clases Detallado, Casos de Uso Detallado, Interfaces de Clases, Diseño de Base de Datos, Diseño de Interfaz de Usuario.

3.1.5 Implementación y Pruebas

En esta fase se procedió a la codificación del software, previa definición de los estándares de programación. Y posteriormente, se llevaron a cabo las pruebas unitarias y funcionales del sistema, esta última se realizó en el laboratorio de “Procesamiento Digital de Señales”.

La metodología utilizada tuvo un enfoque iterativo en la aplicación, debido a que en cada etapa del desarrollo se iban haciendo mejoras del análisis y diseño realizados en etapas anteriores.

3.2 Diseño de Sistemas

3.2.1 Arquitectura del Sistema

Se modelan los componentes lógicos y físicos que conforman el sistema conversor texto-voz.

3.2.1.1 Diagrama de Paquetes

Muestra la división lógica de las clases del sistema y las relaciones al más alto nivel que existen entre paquetes.

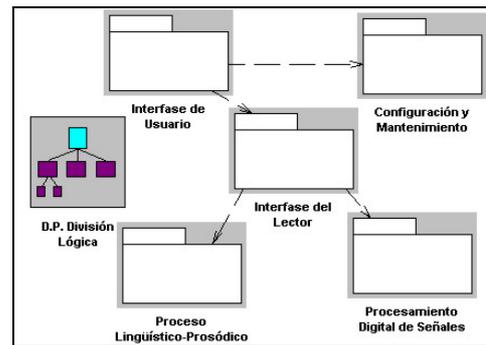


FIGURA 3.1.1.1.1 DIAGRAMA DE PAQUETES DEL SISTEMA CTVLECTOR

1. Interfase de Usuario: Contiene las clases, ventanas, menús, funciones y procedimientos que permiten interactuar al usuario con el sistema. Este paquete interactúa con los paquetes de "Configuración y Mantenimiento" y "Proceso Lingüístico-Prosódico". Físicamente está representado por dos programas (Lector.exe y LectorLC.exe).
2. Configuración y Mantenimiento: Paquete que agrupa las clases y funciones que permiten modificar los valores por omisión del sintetizador (volumen, velocidad de lectura, distancia entre palabras), así como el mantenimiento de los datos usados por el sistema.
3. Interfase del Lector: Este paquete sirve de enlace entre los programas ejecutables (Lector.exe, LectorLC.exe) y la librería (Lector.dll). Por medio de la cual se hace la sintetización de la frase y la codificación de los parámetros LPC. Interactúa con los paquetes "Interfase de Usuario", "Proceso Lingüístico-Prosódico" y "Procesamiento Digital de Señales".
4. Procesado Lingüístico-Prosódico: Agrupa las clases que se utilizan para dividir la frase en palabras, realizar el análisis lingüístico de estas y asignarles la categoría gramatical que les corresponde para luego hacer la identificación de los grupos fónicos, y generar la representación en demisílabas de la frase. Finalmente, calcula los parámetros prosódicos (Duración y Contorno de Frecuencia Fundamental) que se usarán en la generación del habla en el paquete de "Procesamiento Digital de Señales".
5. Procesamiento Digital de Señales: Agrupa las clases que se encargan del procesamiento digital de señales que comprende tanto la codificación de

los parámetros del filtro LPC y las de la señal de excitación multipulso, así como las clases usadas en la síntesis de voz:

- En el primer caso calcula los parámetros del filtro de predicción lineal (LPC) para todas las unidades básicas que se usaran en la generación del habla (en nuestro caso 541 demisílabas), y las amplitudes y ubicaciones de la señal multipulso que se usará como señal de excitación. Estos parámetros por tramas de demisílaba son almacenados en un archivo binario con extensión LPC.
- En la síntesis, usando el archivo LPC creado en la codificación se reconstruye la señal de voz correspondiente a la cadena de demisílabas, y los arreglos de duraciones y contorno de frecuencia fundamental que recibe del “Procesado Lingüístico-Prosódico”.

3.2.1.2 Diagrama de Componentes

Se modela los componentes de software desarrollados que conforman el sistema conversor de texto-voz, mostrando las relaciones de dependencia que hay entre cada componente. En la Figura 3.1.1.2.1 se muestran los componentes del sistema:

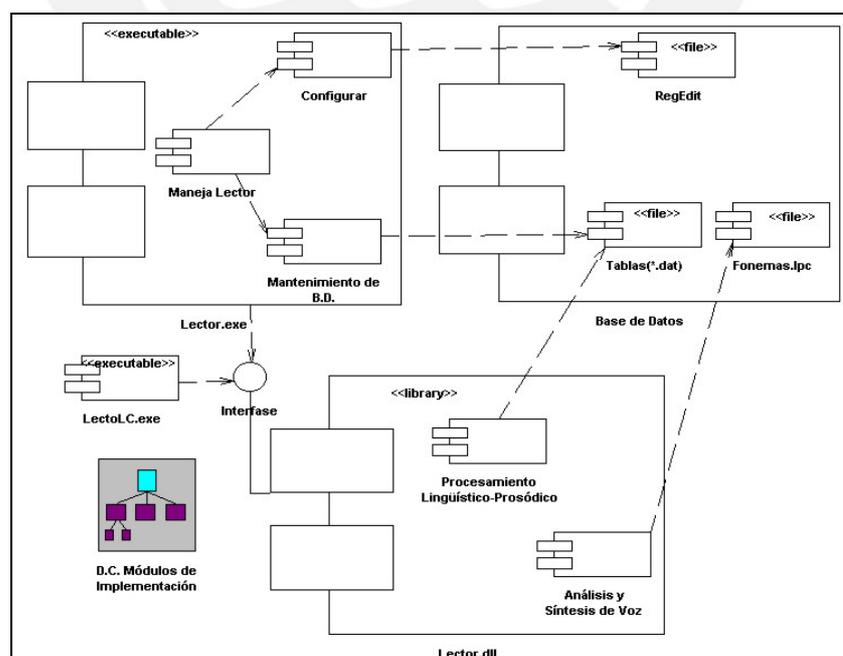


FIGURA 3.1.1.2.1 DIAGRAMA DE COMPONENTES

1. Lector.Exe: Es el programa ejecutable principal, agrupa sub-módulos con los cuales el usuario final interactúa con el sistema para indicarle las tareas que desea que realice. Estas tareas, entre otras, pueden ser: leer un texto, consultar los resultados del procesamiento de una frase (duración de demisílabas, gráfico de contorno de frecuencia fundamental, categorías gramaticales asignadas, etc.), modificar la configuración del sistema. Representa al programa "Lector.exe", que funciona en el entorno Windows, y está desarrollado en Borland Delphi 5.0 (La justificación de la elección de esta herramienta se encuentra en punto 3.2.2) . Contiene tres sub-módulos:
 - Maneja Lector: Es el sub-módulo principal del Lector.exe, permite acceder a las diferentes funcionalidades del sistema. Inicia la lectura de textos, permite modificar los parámetros de configuración y archivos de datos, así como, la visualización de los resultados del procesamiento de una frase. Interactúa con los sub-componentes "Configurar" y "Mantenimiento de B.D." del componente "Lector.exe" y con la librería "Lector.dll" a través de la interfase llamada "Interfase".
 - Configurar: Permite modificar la configuración del sintetizador que se encuentra almacenado en una entrada del registro del sistema operativo Windows. Interactúa con el componente "RegEdit" del componente "Base de Datos".
 - Mantenimiento de B.D: Permite consultar y hacer el mantenimiento de los archivos de datos usados por el sistema (abreviaturas, categorías gramaticales, reglas de duración consonántica, reglas de categorización gramatical, reglas de transcripción fonética, etc.).
2. LectorLC.Exe: Representa al programa ejecutable que realiza la lectura de un archivo de texto directamente desde la línea de comandos de Windows (D.O.S). Esta representado por el programa llamado "LectorLC.exe", y está desarrollado en Visual C++ 6.0 (La justificación de la elección de esta herramienta se encuentra en punto 3.2.2). Se

relaciona con el componente "Lector.dll" por medio del módulo de "Interfase".

3. Base de Datos: Agrupa a los archivos planos que se usarán como base de datos del sistema conversor texto-voz ((La justificación de la elección de este tipo de archivos se encuentra en punto 3.2.2), los cuales se usan para almacenar la configuración, los parámetros del filtro de predicción lineal, y el diccionario léxico-prosódico que permiten realizar el análisis lingüístico-prosódico de una frase dada.
 - RegEdit: Archivo de configuración del sistema operativo Windows usado para almacenar la configuración del sistema.
 - Tablas(*.dat): Representa los archivos de datos y reglas del sistema.
 - Fonemas.lpc: Archivo que almacena los parámetros de predicción lineal correspondientes a las muestras de voz de las demisílabas que se usan en la concatenación de unidades, así como las ubicaciones y amplitudes de la señal de excitación multipulso.
4. Lector.Dll: Librería que agrupa las clases que permiten realizar el análisis lingüístico-prosódico y el procesamiento digital de señales del sistema conversor texto-voz. Este componente viene a ser el corazón del sistema, ya que toda la lógica necesaria para la conversión de texto-voz se encuentra dentro de este componente. Realiza el procesamiento Lingüístico-prosódico y acústico de la frase:
 - Procesamiento Lingüístico-Prosódico: Realiza el análisis lingüístico de la "frase" que recibe, y sobre la base de este análisis genera los parámetros prosódicos (duración y frecuencia fundamental), necesarios para una correcta sintetización de la voz, interactúa con el sub-componente "Tablas" del componente de "Base de Datos".
 - Análisis y Síntesis de Voz: En una primera etapa es la encargada de calcular los parámetros del filtro de predicción lineal y la señal multipulso. Y en una segunda, genera la voz correspondiente a una "frase" usando los parámetros de predicción lineal y la información

prosódica generada en el análisis lingüístico-prosódico, interactúa con el sub-componente "Fonemas.lpc" del componente "Base de Datos".

5. Interfase: Agrupa las funciones que sirven para acceder a la funcionalidad definida en la librería "Lector.dll". Tiene cinco funciones:
 - Inicializar el sintetizador de voz (IniciarSintetizador).
 - Procesar (sintetizar) la frase obtenida del texto (LeerCadena).
 - Terminar el enlace con el sintetizador (TerminarSintetizador).
 - Concatenar dos archivos de sonido (Copiar2doWav).
 - Crear el archivo de parámetros LPC (CreaDemiSilabas)

3.2.2 Diseño de la Implantación

3.2.2.1 Herramientas de Desarrollo

Como se ha especificado en el diagrama de componentes, el sistema está dividido en dos partes principales: un programa ejecutable y una librería.

Debido a que el análisis y diseño del sistema en su conjunto se ha realizado usando una metodología orientada a objetos, y con la finalidad de aprovechar las múltiples ventajas que ofrece una programación orientada a objetos (reutilización de código, herencia, polimorfismo, programas más entendibles y de fácil mantenimiento) en comparación de la programación estructurada. Se ha decidido utilizar lenguajes de programación orientados a objetos para desarrollar los dos componentes del sistema (ejecutable y librería), estos son:

1. Delphi 5.0: Es un lenguaje de programación muy potente totalmente orientado a objetos, cuyo lenguaje de base es el Object Pascal. Una ventaja que tiene Delphi en comparación de otros lenguajes visuales, es que tiene un compilador altamente optimizado, que genera código directamente ejecutable, no pseudo-código que debe ser interpretado posteriormente en tiempo de ejecución. Esto tiene básicamente dos consecuencias: el ejecutable es más rápido y el tamaño final total de la aplicación suele ser inferior, ya que no es necesaria la distribución

adicional del programa que interpreta y ejecuta el código al que suele llamarse runtime. Con esta herramienta se desarrollará la parte “ejecutable” del sistema(Lector.exe), ya que permite crear la interfaz con el usuario de manera rápida.

2. Microsoft Visual C++ 6.0: Como se sabe el lenguaje C es una herramienta muy potente para desarrollar sistemas que requieran una velocidad de ejecución rápida, C++ es la versión orientada a objetos de C, y a la característica mencionada anteriormente del C le añade la potencia de crear programas más entendibles y de fácil mantenimiento. Esta herramienta se ha elegido para desarrollar la “librería” del sistema (Lector.dll).

3.2.2.2 Administración de los Datos

El sistema maneja una serie de archivos binarios para almacenar la información que requiere para cumplir con su tarea. Se eligió trabajar con archivos binarios debido a que la cantidad de información que se maneja no es muy grande: Los archivos de reglas que maneja el sistema tienen un tamaño total aproximado de 43 KB, y el archivo de unidades básicas (fonemas.lpc) tiene un tamaño aproximado de 8 MB, y al iniciarse la ejecución del sistema toda la información que almacenan se carga en la memoria RAM del computador, esto con la finalidad de hacer más rápido el acceso a las reglas y datos del sistema para no tener retardos en la lectura de los textos.

El manejo de Base de Datos relacionales como Oracle o SQL Server quedó descartado, debido a que se requeriría demasiados recursos del computador para ejecutar el programa, tendría que alojar al manejador de base de datos, y para la cantidad de información que se requiere no se justifica tal uso de recursos. Además, que se estaría sacrificando la portabilidad del sistema a cualquier máquina ya que se requeriría que estuviera instalado el RDBMS a utilizar.

3.2.3 Estándares

Los estándares de programación son importantes porque nos facilitan la comprensión de los programas y su mantenimiento, por eso en el Anexo G definimos los estándares usados en el desarrollo del sistema CTVLector.

3.3 Diseño de Objetos

3.3.1 Diseño Funcional

3.3.1.1 Diagrama de Casos de Uso

En este diagrama se presenta los de casos de uso identificados para el sistema conversor de texto-voz, sus actores y la relación entre ambos.

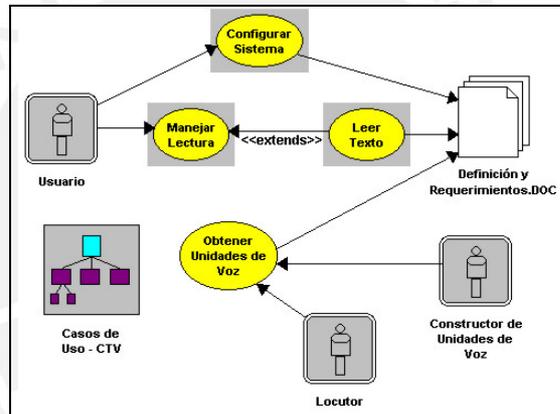


FIGURA 3.2.1.1.1 DIAGRAMA DE CASOS DE USO DEL SISTEMA CTVLECTOR

1. Especificación de Actores:

TABLA 3.2.1.1.1 DEFINICIÓN DE ACTORES.

1.- Nombre	Usuario
Descripción	Representa al usuario final que usa el sistema CTVLector, ingresando el texto que será leído, ya sea digitándolo en el editor del sistema o abriendo un archivo de texto previamente almacenado e iniciando la lectura del texto por el sistema. El usuario también, podrá cambiar la configuración del sistema y los valores de los archivos de datos. Así como consultar el resultado del procesamiento de una frase.
2.- Nombre	Locutor
Descripción	Persona cuya voz se usa para grabar las palabras a partir de las cuales se obtiene las voces correspondientes a las demisílabas a codificar (análisis).

3.- Nombre	Constructor de Unidades de Voz
Descripción	Representa al analista que obtiene las voces correspondientes a las Unidades Básicas (Demisílabas) que serán usadas como parámetros en la sintetización texto. Estas Unidades Básicas son obtenidas a partir de las palabras grabadas de un locutor que tenga buena pronunciación, a partir de estas unidades de voz el analista obtiene las demisílabas que se codifican para obtener los parámetros del filtro de predicción lineal y la señal de excitación multipulso a usar.

2. Especificación de los Casos de Uso:

TABLA 3.2.1.1.2 CASO DE USO CONFIGURAR SISTEMA

Nombre	Configurar Sistema	
Actores	Usuario	
Descripción	Permite modificar la configuración del sistema como son el volumen, velocidad de lectura, tamaño de interpolación, distancia en ms entre palabras, sílabas y demisílabas, etc. Adicionalmente permite hacer la consulta y mantenimiento de los archivos de reglas y datos usados en el sistema.	
Pre-Condicion	<ul style="list-style-type: none"> ➤ Los valores de la configuración del sistema deben existir en el registro del sistema operativo Windows (RegEdit). ➤ Los archivos de datos y reglas deben haber sido creadas. 	
Flujo de Eventos	Paso	Acción
	1	El Use Case comienza cuando el usuario selecciona la opción de CONFIGURACION o MANTENIMIENTO en el menú del sistema.
	2	El sistema presenta opciones de configuración y mantenimiento de tablas del sistema.
	3	El usuario modifica el valor de la configuración o del archivo de datos seleccionado.
	4	Si el usuario acepta los cambios, el sistema guarda la configuración del sistema o los datos en el archivo de datos respectivo.
5	El sistema oculta la pantalla de configuración o mantenimiento de la tabla elegida y termina.	
Post-Condicion	La información de la configuración del sistema o de los archivos de datos y reglas debe estar actualizada.	
Excepciones	Paso	Acción
	4	Si el usuario elige cancelar, se deshacen los cambios de la configuración del sistema o de los archivos de datos y se termina este caso de uso.

TABLA 3.2.1.1.3 CASO DE USO MANEJAR LECTURA

Nombre	Manejar Lectura	
Actores	Usuario	
Descripción	Permite manejar la lectura de un texto, las opciones disponibles son: iniciar lectura, detenerlo, reiniciar la lectura, repetir la última frase, modificar el volumen y la velocidad de la lectura.	
Pre-Condicion	<ul style="list-style-type: none"> ➤ Se haya iniciado el sistema correctamente. ➤ El sistema debe de haber cargado la configuración del sistema, los datos y reglas ha usar en la síntesis de textos en la memoria del computador. 	
Flujo de Eventos	Paso	Acción
	1	El Use Case comienza cuando el usuario carga el texto a leer e inicia su lectura.
	2	El usuario inicia la lectura del texto (P.Ext. 1)
	3	El usuario elige una de las opciones que permiten alterar el normal desarrollo de la lectura.
	4	El sistema ejecuta la opción seleccionada.
Post-Condicion	Se modifica la lectura normal del texto de acuerdo a la opción seleccionada.	
Excepciones	Paso	Acción
	-	-

TABLA 3.2.1.1.4 CASO DE USO LEER TEXTO

Nombre	Leer Texto	
Actores	Usuario	
Descripción	Genera la representación acústica de un texto. El cual puede estar almacenado en un archivo de texto o ser digitado en el editor del sistema.	
Pre-Condicion	<ul style="list-style-type: none"> ➤ El texto a leer debe estar sin errores tipográficos y ortográficos. 	
Flujo de Eventos	Paso	Acción
	1	(P.Ext 1) El Use Case comienza cuando el usuario ordena leer el texto indicado.
	2	Mientras exista texto que procesar
	A	El sistema obtiene la frase a procesar.
	B	El sistema normaliza la frase.
	C	El sistema hace el Preproceso de la frase
	D	El sistema hace el análisis gramatical de la frase.
	E	El sistema realiza la inserción de las pausas no marcadas ortográficamente.
	F	El sistema convierte los grafemas de la frase a alófonos y luego a Demisílabas.
	G	El sistema hace el análisis prosódico de la frase generando la duración de sus demisílabas y su contorno de Frecuencia Fundamental.
H	El sistema envía las demisílabas y los parámetros prosódicos al sintetizador de voz.	

	I	El sistema escoge la secuencia de unidades de voz de la base de datos de parámetros lpc que se corresponde con la secuencia de demisílabas de la frase analizada.
	J	El sistema reconstruye la señal de voz correspondiente a la unidad básica procesada, modificando su frecuencia fundamental.
	K	El sistema agrega o elimina muestras de la señal reconstruida para adecuarla a la duración calculada.
	3	De acuerdo a lo definido por el usuario, el sistema: <ul style="list-style-type: none"> ➤ Toca en ese instante la secuencia de sonidos generada para la frase. ➤ Almacena la señal de voz en un archivo de sonido para su posterior uso. ➤ Realiza los dos pasos anteriores a la vez.
Post-Condicion	Se genera la representación acústica del texto dado para su lectura.	
Excepciones	Paso	Acción
	1	El sistema no encuentra el archivo de texto indicado para leer, en ese caso el sistema no inicia la lectura.
	3	El sistema no encuentra el reproductor de archivos de sonidos, por medio del cual reproducir la representación acústica del texto, se emitirá un mensaje de error indicando lo sucedido.

TABLA 3.2.1.1.5 CASO DE USO OBTENER UNIDADES DE VOZ

Nombre	Obtener Unidades de Voz	
Actores	Locutor, Constructor de Unidades de Voz	
Descripción	A partir de las palabras grabadas por el locutor se obtiene las muestras de voz de las Unidades Básicas (demisílabas), y con ellas calcula los parámetros del filtro de predicción lineal, las ubicaciones y amplitudes de la señal multipulso; para luego almacenarlos en un archivo binario con extensión lpc.	
Pre-Condicion	<ul style="list-style-type: none"> ➤ Tener la lista de demisílabas necesarias para el sistema conversor texto-voz, en nuestro caso hay 541 unidades básicas identificadas como necesarias. ➤ Tener la lista de palabras que se necesitan para obtener las demisílabas requeridas. 	
Flujo de Eventos	Paso	Acción
	1	El Use Case inicia grabando las palabras pronunciadas por el locutor donante de la voz.
	2	El constructor de unidades de voz corta las palabras grabados previamente para obtener las unidades de voz correspondientes a las demisílabas
	3	El constructor de unidades de voz almacena todas las unidades de voz en un directorio de archivos.
	4	El constructor de unidades de voz ejecuta el proceso de análisis de voz (codificación).
	5	Para cada unidad de voz hacer

	A	El proceso filtra la señal original para quitarle las características irrelevantes.
	B	El proceso divide la señal en pequeñas tramas, debido a que se asume que las características y dimensión del tracto bucal permanecen fijas para periodos de 10 o 20 milisegundos, en este trabajo el eventanamiento se realiza a 8 ms.
	C	El proceso calcula los parámetros de la señal de voz que se usaran en la concatenación de voz, esto es de acuerdo al método a usar, en nuestro caso Predicción Linear con entrada Multipulso.
	D	El proceso almacena los parámetros calculados en un archivo binario (.lpc) y quedan disponibles para ser usados en la síntesis de voz.
Post- Condiciones	Archivo de datos con los parámetros que permitan reconstruir la señal de voz correspondiente a las unidades básicas y sus respectivas etiquetas creada (Fonemas.lpc).	
Excepciones	Paso	Acción
	-	-

3.3.1.2 Diagrama de Clases

Modela los aspectos estructurales o requerimientos de información del sistema conversor texto-voz. Se han organizado de acuerdo a la división de los paquetes mostrados en el “Diagrama de Paquetes” de la Figura 3.1.1.1.1. Con la finalidad de facilitar el entendimiento del modelo de clases del sistema conversor texto-voz en su conjunto se muestra tanto los diagramas de clases modelados en el presente trabajo como en el anterior [Mateo 2000], pero para el diccionario de datos sólo se tomaran en cuenta las clases, atributos y métodos desarrollados en el presente trabajo.

1. Interfase de Usuario: Agrupa las clases que permiten interactuar al usuario con el sistema. Estas clases se han dividido adicionalmente en dos grupos: La primera correspondiente a las clases que permiten construir una interfase visual del sistema y la otra que corresponde a la interfase por línea de comandos.
 - Clases de Interfase de Usuario Visual: Permiten interactuar al usuario con el sistema en entorno Windows. Usados en la implementación del programa Lector.exe.. Su diagrama se muestra en la Figura 3.2.1.2.1.

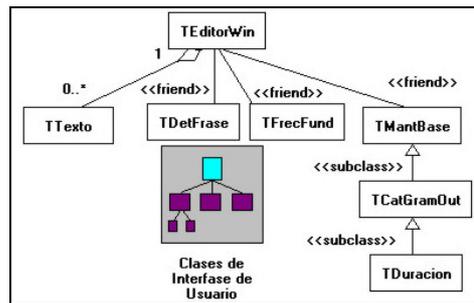


FIGURA 3.2.1.2.1 CLASES DE INTERFASE VISUAL

Diccionario de Datos:

TABLA 3.2.1.2.1 DICCIONARIO DE DATOS DE LA CLASE TEDITORWIN

Clase	Definición	
TEditorWin	Clase activa que representa la ventana principal del sistema desde la cual se invocan las diferentes funcionalidades y operaciones que tiene el sistema.	
Atributos		
Nombre	Tipo	Definición
-	-	-
Operaciones		
Descripción	Alias	Definición
RecuperaParametros (-)	-	Lee los parámetros de configuración del sistema desde el registro del sistema operativo y los almacena en memoria.
Abrir (+)	actAbrir	Método que permite buscar un archivo de texto y cargarlo en una nueva instancia de la clase TTexto para poder leerlo.
Borrar (+)	actBorrar	Borra el texto seleccionado.
Buscar (+)	actBuscar	Busca una cadena de caracteres en el texto.
BuscSgte	actBuscSgte	Continúa la búsqueda de una cadena.
Cerrar (+)	actCerrar	Cierra la ventana actual, eliminando una instancia de la clase TTexto.
Continuar (+)	actContinuar	Continúa con la lectura del texto actual que fue detenido o pausado.
Copiar (+)	actCopiar	Copia el texto seleccionado.
Cortar (+)	actCortar	Corta la cadena de texto.
Detener (+)	actDetener	Detiene la lectura de un texto.
DetFrase (+)	actDetFrase	Muestra el detalle de la última frase procesada en sílabas, fonemas y unidades básicas(demisílabas). Para lo cual crea una instancia de la clase TDetFrase.
Duración (+)	actDuracion	Muestra la duración asignada a cada una de las Demisílabas de la frase luego de su procesamiento. Para lo cual crea una instancia de la clase TDuracion.

F0 (+)	actF0	Muestra el gráfico correspondiente al Contorno de Frecuencia Fundamental de la última frase procesada.
FecHora (+)	actFecHora	Inserta la fecha y horas actuales en el texto en la posición del cursor.
FraCatGram (+)	actFraCatGram	Muestra la categorización gramatical de la última frase procesada. Para lo cual crea una instancia de la clase TCatGramOut.
GenWav (+)	actGenWav	Genera un archivo de sonido en formato Wav con la lectura del texto completo.
Guardar (+)	actGuardar	Guarda el texto que se encuentra en la ventana actual en el archivo del cual proviene, o si es un archivo nuevo pide la ruta y nombre del archivo con el cual se desea guardar.
GuardarComo (+)	actGuardarComo	Solicita la ruta y nombre del archivo con el cual se desea guardar el texto actual. Y almacena el texto bajo ese nombre.
Leer (+)	actLeer	Inicia la lectura del texto actual.
LeeWav (+)	actLeeWav	Lee al instante el texto actual, y además, genera un archivo de sonido de formato Wav con la lectura.
Nuevo (+)	actNuevo	Método que permite crear una instancia de la clase Ttexto, donde se puede ingresar un nuevo texto a leer.
Pausar (+)	actPausar	Detiene la lectura del texto.
Pegar (+)	actPegar	Pega el texto cortado o copiado en la posición del cursor.
Repetir (+)	actRepetir	Repite la lectura del texto desde el inicio.
RepetirFrase (+)	actRepFrase	Repite la lectura de la última frase leída.
Salir (+)	actSalir	Termina la ejecución del sistema, previamente si hubiera habido cambios en algún texto pide la confirmación de grabar. Elimina todas las instancias de la clase TTexto y el enlace con la librería Lector.dll.
SelecTodo (+)	actSelecAll	Selecciona todo el texto.
TEditorWin (+) Constructor	FormCreate	Crea e inicializa los objetos de la clase TEditorWin, recupera los parámetros de configuración del sistema y además se encarga de crear e inicializar las instancias de las clases y datos de la librería Lector.dll que permite sintetizar los textos ingresados.
~TEditorWin (+) Destructor	FormDestroy	Elimina los objetos y datos de la clase TeditorWin y de la librería Lector.dll.
MuestraAbrev (+)	mmiAbrevClick	Crea una instancia de la clase TAbrev donde se listan las abreviaturas.
MuestraAmbiguo (+)	mmiAmbiguaClick	Crea una instancia de la clase TAmbiguo donde se listan las palabras que son ambiguas en su acentuación.

MuestraAtona (+)	mmiAtonaClick	Crea una instancia de la clase TAtona donde se listan las palabras que no se acentúan.
MuestraCatGram (+)	mmiCatGramClick	Crea una instancia de la clase TCatGram donde se listan las categorías gramaticales consideradas en el sistema.
Configurar (+)	mmiConfigClick	Permite realizar el mantenimiento de la configuración del sistema. Para lo cual crea una instancia de la clase TConfigurar.
MuestraDurBase Cons (+)	mmiConsonantesClick	Crea una instancia de la clase TDurBaseC donde se lista la duración base de las consonantes.
MuestraPalGram (+)	mmiExcepGramClick	Crea una instancia de la clase TPalGram donde se listan las palabras consideradas como excepciones gramaticales.
MasVeloc (+)	mmiMasVelocClick	Incrementa la velocidad de lectura en 10%.
MasVol (+)	mmiMasVolClick	Incrementa en 10% el volumen de lectura.
MenosVeloc (+)	mmiMenosVelocClick	Disminuye la velocidad de lectura en 10%.
MenosVol (+)	mmiMenosVolClick	Disminuye el volumen de lectura en un 10%.
MuestraPausas (+)	mmiPausasClick	Crea una instancia de la clase TPausa donde se listan las pausas y sus duraciones.
MuestraReglaDur Cons (+)	mmiRegDurConsClick	Crea una instancia de la clase TReglaDurCons donde se listan las reglas usadas para calcular la duración de algunas consonantes.
MuestraSigla (+)	mmiSiglaClick	Crea una instancia de la clase TSigla donde se listan las siglas usadas por el sistema.
MuestraTerminaciones (+)	mmiTermClick	Crea una instancia de la clase TArcTerm donde se listan las terminaciones y desinencias de palabras que permiten identificar a que categoría gramatical pudieran pertenecer.
MuestraDurBase Vocal (+)	mmiVocalesClick	Crea una instancia de la clase TDurBaseV donde se lista la duración base de las vocales.

TABLA 3.2.1.2.2 DICCIONARIO DE DATOS DE LA CLASE TTEXTO

Clase	Definición	
TTexto	Clase que representa al texto propiamente dicho que será leído por el sintetizador.	
Atributos		
Descripción	Alias	Definición
Arc_Leer (-) : string	-	Almacena la ruta y nombre del archivo de texto que se leerá.
Est_Ado (-) : UINT8	-	Indica el estado de la lectura: 0: Leyendo. 1: En pausa. 2: Repitiendo la última frase.

Tip_Lectura(-) : UINT8	-	Indica el tipo de lectura que se esta realizando. (0): leer al instante. (1): grabar en un archivo de sonido. (2): ambos.
Tex_ToWav (-) : string	-	Almacena el nombre del archivo WAVE a generar cuando el sistema recibe la orden de generar un archivo de sonido.
Wav_Temp(-) : string	-	Almacena el nombre del archivo de sonido temporal que se crea con cada lectura.
Fra_Fonemas (+) : Pchar	-	Frase en fonemas luego de su procesamiento.
Fra_Letras (+) : Pchar	-	Almacena la frase luego del proceso de síntesis ya que no siempre es igual a la frase inicial debido a que se puede haber eliminado una secuencia de caracteres no reconocidos por el sistema.
Fra_Se (+) : string	-	Representa a la unidad de trabajo (FRASE).
Fra_Silabas (+) : Pchar	-	Frase en sílabas luego de su procesamiento.
Fra_UnidBasic (+) : PChar	-	Frase en unidades básicas(Demisílabas) luego de su procesamiento.
Operaciones		
Descripción	Alias	Definición
BuscarAbrevSigla (-) Retorno: Boolean	-	Busca la abreviatura o sigla que recibe como parámetro en el archivo dado también como parámetro. Si lo encuentra devuelve verdadero caso contrario será falso.
EncontrarFraseWin (-) Retorno: string	-	Busca la siguiente frase desde la posición del cursor hasta encontrar un marcador de fin de frase en la ventana de texto actual.
ObtenerSignoMenor (-) Retorno: char	-	De dos signos ortográficos que recibe como parámetros devuelve el signo ortográfico más cercano al inicio de la frase.
ReproducirWav (-)	-	Reproduce el archivo de sonido tipo WAVE que recibe como parámetro.
UbicarFinFrase (-) Retorno: char	-	Encuentra la ubicación del signo que representa el fin de una frase en el texto.
GeneraWave (+)	-	Llama a la función de interfase Copiar2doWav de la librería "Lector.dll" que agrega el segundo archivo de sonido temporal al primero. Se usa para generar un archivo de sonido correspondiente al texto que se está procesando.
GetArcNom (+) Retorno: string	-	Devuelve el nombre del archivo de texto que se está leyendo o que se va a leer.
GetEstado (+) Retorno: UINT8	-	Devuelve el estado de la lectura del texto.
GetTipLectura (+) Retorno: UINT8	-	Devuelve el tipo de lectura que se está realizando.

LeerTextoWin (+)	-	Manda a leer el contenido del texto de acuerdo al parámetro que recibe, si pTipo es: 0: Lee al instante el texto. Si es 1: Genera un archivo de sonido con el resultado de la lectura. Si es 2: Lee al instante el texto, y además, genera un archivo de sonido.
LeerTextoWin (+)	-	Es el procedimiento principal que se encarga de manejar la lectura del texto. Es el encargado de segmentar el texto en frases, enviar estas al sintetizador para obtener su equivalente acústico, y de acuerdo a lo solicitado reproducir el sonido al instante o almacenarlo en un archivo de sonido para su posterior reproducción.
LeerUltWav (+)	-	Lee el último archivo WAVE temporal generado correspondiente a la última frase procesada antes de que se detuviera la lectura del texto.
Open (+)	-	Carga el contenido del archivo que recibe como parámetro en el editor de texto actual.
PutArcNom (+)	-	Asigna el nombre de archivo que recibe como parámetro al atributo Arc_Leer de la instancia de la clase TTexto.
PutEstado (+)	-	Asigna el valor del estado de la lectura.
PutTipLectura (+)	-	Asigna el tipo de lectura que se desea hacer con el texto.
TTexto (+) Constructor	FormCreate	Crea un objeto texto, reservando espacio de memoria necesaria para sus atributos.
~TTexto (+) Destructor	FormDestro y	Elimina una instancia de la clase TTexto.

TABLA 3.2.1.2.3 DICCIONARIO DE DATOS DE LA CLASE TDETFRASE

Clase		Definición
TDetFrase		Muestra el detalle de la frase luego de su procesamiento.
Atributos		
Nombre	Alias	Definición
Let_Ras (+) : PChar	memLetras	Almacena la frase en letras.
Sil_Abas (+) : PChar	memSilabas	Almacena la frase en sílabas.
Fon_Emas (+) : PChar	memFonemas	Almacena la frase en su representación en fonemas.
Unid_Basicas (+) : PChar	memUnidBasi	Almacena la frase en su representación en unidades básicas (demisílabas)
Operaciones		
Descripción	Alias	Definición
TDetFrase (+) Constructor.	FormCreate	Asigna los valores del detalle de la frase a los campos a visualizar.
Salir (+)	sbSalirClick	Sale de la ventana de detalles de la frase.

TABLA 3.2.1.2.4 DICCIONARIO DE DATOS DE LA CLASE TFRECFUND

Clase	Definición	
TFrecFund	Muestra el gráfico correspondiente al contorno de frecuencia fundamental calculado para la última frase procesada.	
Atributos		
Nombre	Alias	Definición
Gra_Fico (+) : TChart	chtChart	Atributo que guarda el gráfico del contorno de frecuencia fundamental.
Operaciones		
Descripción	Alias	Definición
TFrecFund (+) Constructor.	FormCreate	Abre el archivo de texto que contiene los números reales que representan la variación de la frecuencia fundamental de la última frase procesada con lo cual construye su gráfico.
Imprimir (+)	tbnImprimirClick	Imprime el gráfico del contorno de frecuencia fundamental.
Salir (+)	tbnSalirClick	Sale de la ventana del gráfico de F0.

TABLA 3.2.1.2.5 DICCIONARIO DE DATOS DE LA CLASE TCATGRAMOUT

Clase	Definición	
TCatGramOut	Es una subclase de la clase TMantBase. Muestra las categorías gramaticales asignadas a las palabras componentes de la frase.	
Atributos		
Nombre	Alias	Definición
-	-	Las hereda de su superclase TMantBase.
Operaciones		
Descripción	Alias	Definición
TCatGramOut (+) Constructor.	FormCreate	Carga el archivo de texto que contiene las palabras de la frase y las categorías gramaticales asignadas en la ventana.

TABLA 3.2.1.2.6 DICCIONARIO DE DATOS DE LA CLASE TDURACION

Clase	Definición	
TDuracion	Es una subclase de la clase TMantBase. Contiene la duración asignada a cada demisílaba de la frase.	
Atributos		
Nombre	Alias	Definición
Dur_Total	edDurTot	Contiene la duración total de la frase en ms.
Operaciones		
Descripción	Alias	Definición
TDuracion (+) Constructor.	FormCreate	Carga el archivo de texto que contiene las demisílabas y sus duraciones asignadas.

2. Clases de Interfase de Usuario No Visual: Esta vista contiene una única clase que permite al usuario interactuar con el sistema mediante la línea

de comandos (D.O.S). Con esta clase se crea el programa LectorLC.exe. Su diagrama se muestra en la Figura 3.2.1.2.2.

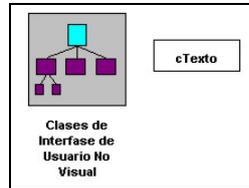


FIGURA 3.2.1.2.2 CLASES DE INTERFASE DE USUARIO NO VISUAL

Diccionario de Datos

TABLA 3.2.1.2.7 DICCIONARIO DE DATOS DE LA CLASE CTEXTO

Clase	Definición	
cTexto	Clase que abstrae el concepto de un archivo de texto (ASCII), que se va a leer. Agrupando los atributos y operaciones necesarios para cumplir con esta función.	
Atributos		
Nombre	Alias	Definición
Arc_Lpc (-) : UINT8[20]	-	Nombre del archivo de parámetros LPC.
Buf1 (-) : char *	-	Puntero que apunta a la primera mitad del buffer Centinela.
Buf2 (-) : char *	-	Puntero que apunta a la segunda mitad del buffer Centinela.
Cen_Tinela (-) : char *	-	Buffer donde se almacenan los caracteres leídos del archivo de texto con la finalidad de agilizar el reconocimiento de una cadena de caracteres (tipos de palabra) del texto, recibe este nombre debido a que se usa el algoritmo de Buffer con Centinela para su implementación.
Del_Antero (-) : INT16	-	Índice del buffer Centinela.
Des_File (-) : FILE *	-	Descriptor del archivo de texto a leer.
Dist_Demi (-) : TUINT32	-	Distancia entre Demisílabas.
Dist_Silaba (-) : TUINT32	-	Distancia entre sílabas.
Dist_Palabra (-) : TUINT32	-	Distancia entre palabras.
Fra_Demisílaba (-) : char *	-	Frase en Demisílabas.
Fra_Fonema (-) : char *	-	Frase en fonemas.
Fra_Letra (-) : char *	-	Frase luego de haber sido procesada.

Fra_Se (-) : char[1500]	-	Frase a procesar obtenida del texto.
Fra_Silaba (-) : char *	-	Frase en sílabas.
Int_DemiSil (-) : TUINT32	-	Tamaño de interpolación entre Demisílabas.
Int_Silaba (-) : TUINT32	-	Tamaño de interpolación entre sílabas.
Int_Palabra (-) : TUINT32	-	Tamaño de interpolación entre palabras.
Mod_Dur (-) : TUINT32	-	Parámetro de configuración del sistema que indica si se va a modificar la duración de las demisílabas al momento de sintetizarla.
Mod_F0 (-) : TUINT32	-	Parámetro de configuración del sistema que indica si se va a modificar el contorno de frecuencia fundamental al momento de sintetizar las frases del texto.
Path_Datos (-) : UINT8	-	Ruta donde se encuentran los datos del sintetizador.
Path_Salida (-) : UINT8	-	Directorio donde se guarda los archivos creados como resultado del procesamiento de las frases del texto.
Sin_Tetizador (-) : HANDLE *	-	Puntero a la dirección de memoria del sintetizador de la librería Lector.dll.
Tam_Frase (-) : INT16	-	Tamaño de la frase antes de procesarla.
Tex_ToWav (-) : char *	-	Nombre del archivo de sonido a generar como resultado de la lectura del texto.
Tip_Lectura (-) : UINT8	-	Indica el tipo de lectura que se esta realizando. (0): leer al instante. (1): grabar en un archivo de sonido. (2): ambos.
Vel_Ocidad (-) : TUINT32	-	Velocidad de lectura del sistema.
Vol_Umen (-) : TUINT32	-	Volumen de lectura del sistema.
Wav_Temp (-) : char[15]	-	Nombre del archivo de sonido temporal que se crea con cada lectura de una frase del texto.
Operaciones		
Descripción	Alias	Definición
Cargar_Configuración (-) Retorno: void	-	Recupera la configuración de parámetros del sistema.
EncontrarFraseCon (-) Retorno: bool	-	Encuentra la siguiente frase a procesar.
CapturaChar (-) Retorno: INT8	-	Devuelve un caracter del buffer de entrada Centinela. También, es la encargada llenar el buffer cuando se han leído todos los caracteres de este.

DevuelveChar (-) Retorno: void	-	Retrocede una posición el índice del buffer de entrada Centinela.
EsFinFrase (-) Retorno: bool	-	Verifica si el caracter que recibe como parámetro es un caracter de FIN DE FRASE.
BuscarAbrevSigla (-) Retorno: bool	-	Verifica si el parámetro pAbrev que recibe, se encuentra en los archivos de abreviaturas o siglas.
cTexto (+) Constructor.	-	Inicializa las variables a utilizar en el proceso de lectura, creando un enlace con el sintetizador, reservando el espacio de memoria necesario para almacenar los datos a usar por el sintetizador.
~cTexto (+) Destructor.	-	Elimina el enlace con el sintetizador y libera la memoria reservada por el programa.
LeerTextoCon (+) Retorno: void	-	Maneja la lectura del texto. Obtiene las frases incluidas en el texto y las envía a sintetizar.

3. Configuración y Mantenimiento: Agrupa a las clases que permiten ver y modificar la configuración del sistema, así como los datos que utiliza. Su diagrama se muestra en la Figura 3.2.1.2.3.

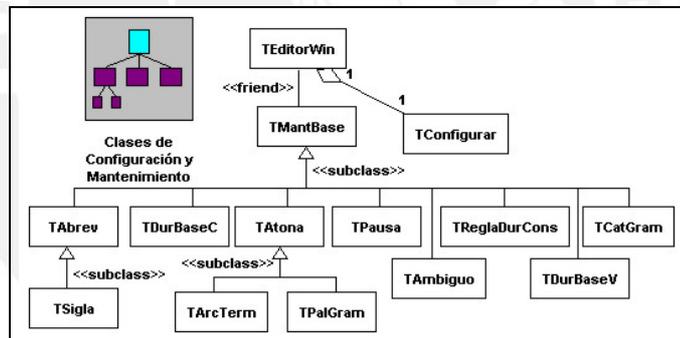


FIGURA 3.2.1.2.3. CLASES DE CONFIGURACIÓN Y MANTENIMIENTO

Diccionario de Datos

TABLA 3.2.1.2.8 DICCIONARIO DE DATOS DE LA CLASE TCONFIGURAR

Clase	Definición	
Tconfigurar	Es la clase que nos permitirá modificar los parámetros por omisión que tiene el sistema, como son: la velocidad de lectura, volumen de voz, tamaño de interpolación, distancia de interpolación, si se desea aplicar los cambios prosódicos (duración y contorno de F0) a la sintetización de la frase, etc.	
Atributos		
Nombre	Alias	Definición
Dis_DemiOmi (-) : TUINT32	-	Distancia entre Demisílabas por omisión.
Dis_PalaOmi (-) : TUINT32	-	Distancia entre palabras por omisión.

Dis_SilaOmi (-) : TUINT32	-	Distancia entre sílabas por omisión.
Mod_DurOmi (-) : Boolean	-	Indica por omisión si se va a modificar la duración de las demisílabas.
Mod_F0Omi (-) : Boolean	-	Indica por omisión si se va a modificar el contorno de frecuencia fundamental.
Num_InterPalaOmi (-) : TUINT32	-	Número de muestras a interpolar por omisión entre palabras.
Num_InterSilaOmi (-) : TUINT32	-	Número de muestras de interpolación por omisión entre sílabas.
Num_InterDemi Omi (-) : TUINT32	-	Número de muestras de interpolación por omisión entre Demisílabas.
Vel_Omisión (-) : TUINT32	-	Representa la velocidad por omisión que tiene el sintetizador de voz (Velocidad de lectura).
Vol_Omisión (-) : TUINT32	-	Representa el volumen por omisión que tiene el sintetizador de voz.
Mod_Dur (+) : Boolean	chbModDur	Indica si se va a modificar la duración de las demisílabas.
Mod_F0 (+) : Boolean	chbModF0	Indica si se va a modificar el contorno de frecuencia fundamental de las frases.
Vel_Ocidad (+) : TUINT32	tbrVelocidad	Indica la velocidad de lectura del sistema.
Vol_Umen (+) : TUINT32	tbrVolumen	Indica el volumen de lectura del sistema.
Num_MuesInter Demi (+) : TUINT32	tbrTamDemi	Indica el número de muestras que se tomará para realizar la interpolación entre demisílabas.
Num_MuesInter Sila (+) : TUINT32	tbrTamSila	Indica el número de muestras que se tomará para realizar la interpolación entre sílabas.
Num_MuesInter Pala (+) : TUINT32	tbrTamPala	Indica el número de muestras que se tomarán para realizar la interpolación entre palabras.
Dis_EntreDemi (+) : TUINT32	tbrDistDemi	Indica la distancia de interpolación entre demisílabas.
Dis_EntreSila (+) : TUINT32	tbrDistSila	Indica la distancia de interpolación entre sílabas.
Dis_EntrePala (+) : TUINT32	tbrDistPala	Indica la distancia de interpolación entre palabras.
Operaciones		
Nombre	Alias	Definición
CargarValOmision (-)	-	Cargar en memoria los valores por omisión de la configuración del sintetizador.
ModifConfig (-)	tbnGuardarClick	Guarda los cambios hechos en la configuración del sistema.
MostrarConfig (-)	-	Lee la configuración actual del sistema.
TConfigurar (+) Constructor.	FormCreate	Inicializa los objetos de la clase TConfigurar.

Salir (+)	tbnSalirClick	Sale del proceso de configuración del sistema.
RecuperaValOmi (+)	TbnValOmiClick	Restaura los valores predeterminados para los parámetros del sistema.

TABLA 3.2.1.2.9 DICCIONARIO DE DATOS DE LA CLASE TMANTBASE

Clase		Definición
TMantBase		Es una superclase que maneja las consultas, impresión y mantenimiento de las tablas de los archivos de datos del sistema conversor texto a voz. A partir de esta clase se derivan una serie de subclases con características particulares para hacer la consulta y mantenimiento de los distintos archivos de datos usados por el sistema.
Atributos		
Nombre	Alias	Definición
Cam_Bio (#) : Boolean	-	Indica si ha habido algún cambio en los datos del archivo mostrado.
Est_Ado (+) : string	lblEstado	Indica el modo en que se encuentra la ventana de mantenimiento (Consulta o Edición).
Nro_Reg (+) : UINT16	LblNroReg	Contiene el número de registros mostrados en la ventana de consulta.
Operaciones		
Descripción	Alias	Definición
ImprimeCabecera (-)	-	Imprime la cabecera del reporte a generar.
EscribeNroReg (#)	-	Muestra el número de registros o filas.
MensajeMantError (#)	-	Muestra el mensaje de error generado al abrir el archivo de texto que recibe como parámetro.
Consultar (+)	tbnConsultarClick	Coloca el mantenimiento en modo Consulta, es decir, no se puede hacer modificaciones.
Editar (+)	tbnEditarClick	Coloca el mantenimiento en modo de Edición, es decir, permite hacer cambios en la tabla.
Imprimir (+)	tbnImprimirClick	Permite imprimir el contenido de la tabla.
Salir (+)	tbnSalirClick	Sale de la ventana de consulta.

TABLA 3.2.1.2.10 DICCIONARIO DE DATOS DE LA CLASE TABREV

Clase		Definición
Tabrev		Es una subclase de la clase TMantBase. Permite realizar la consulta, mantenimiento e impresión de las abreviaturas usadas en el sistema.
Atributos		
Nombre	Alias	Definición
Arc_Nom (#) : string	-	Almacena la ruta y el nombre del archivo de datos que contiene las abreviaturas.
Operaciones		
Descripción	Alias	Definición

LlenaGrilla (#)	-	Lee el archivo que recibe como parámetro y lo muestra en la ventana de consulta (Grilla).
TAbrev (+) Constructor.	FormCrea e	Crea la ventana de consulta y carga los datos del archivo de datos.
Guardar (+)	TbnGuarda r Click	Guarda en el archivo de datos de las abreviaturas los cambios hechos por el usuario.

TABLA 3.2.1.2.11 DICCIONARIO DE DATOS DE LA CLASE TSIGLA

Clase		Definición
TSigla		Es una subclase de la clase TAbrev. Permite realizar la consulta, mantenimiento e impresión del archivo de siglas usadas en el sistema.
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
-	-	No tiene operaciones propias.

TABLA 3.2.1.2.12 DICCIONARIO DE DATOS DE LA CLASE TDURBASEC

Clase		Definición
TDurBaseC		Es una subclase de la clase TMantBase. Permite realizar la consulta, mantenimiento e impresión de las duraciones base de las consonantes.
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TDurBaseC (+) Constructor.	FormCrea e	Crea la ventana de consulta y carga los datos del archivo de datos de las duraciones base.
Guardar (+)	tbnGuardar Click	Guarda en el archivo de datos de las duraciones base de las consonantes los cambios hechos por el usuario.

TABLA 3.2.1.2.13 DICCIONARIO DE DATOS DE LA CLASE TATONA

Clase		Definición
TAtona		Es una subclase de la clase TMantBase. Permite realizar la consulta e impresión del diccionario léxico que contiene las palabras átonas consideradas en el sistema.
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TDurBaseC (+) Constructor.	FormCrea e	Crea la ventana de consulta y carga los datos del archivo de datos de las palabras átonas.

TABLA 3.2.1.2.14 DICCIONARIO DE DATOS DE LA CLASE TARCTERM

Clase	Definición	
TArcTerm	Es una subclase de la clase TAtona. Permite realizar la consulta e impresión del diccionario léxico que contiene las terminaciones y desinencias comunes mediante las cuales se puede establecer a que categoría gramatical pertenece una palabra.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
-	-	No tiene operaciones propias.

TABLA 3.2.1.2.15 DICCIONARIO DE DATOS DE LA CLASE TPALGRAM

Clase	Definición	
TPalGram	Es una subclase de la clase TAtona. Permite realizar la consulta e impresión del diccionario léxico que contiene las palabras consideradas como excepciones gramaticales en el sistema (conocidas también como palabras gramaticales), es decir, que no pueden ser identificadas aplicando las reglas gramaticales ni usando el diccionario de terminaciones y desinencias.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
-	-	No tiene operaciones propias.

TABLA 3.2.1.2.16 DICCIONARIO DE DATOS DE LA CLASE TPAUSA

Clase	Definición	
TPausa	Es una subclase de la clase TMantBase. Permite realizar la consulta, mantenimiento e impresión de las pausas y sus duraciones usadas en el sistema.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TPausa (+) Constructor.	FormCreate	Crea la ventana de consulta y carga los datos del archivo de pausas.
Guardar (+)	tbnGuardar Click	Guarda en el archivo de pausas los cambios hechos por el usuario.

TABLA 3.2.1.2.17 DICCIONARIO DE DATOS DE LA CLASE TAMBIGUO

Clase	Definición	
TAmbiguo	Es una subclase de la clase TMantBase. Permite realizar la consulta e impresión del diccionario léxico que contiene las palabras ambiguas en su acentuación usadas en el sistema.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TAmbiguo (+) Constructor.	FormCreate	Crea la ventana de consulta y carga los datos del archivo de palabras ambiguas.

TABLA 3.2.1.2.18 DICCIONARIO DE DATOS DE LA CLASE TREGLADURCONS

Clase	Definición	
TReglaDurCons	Es una subclase de la clase TMantBase. Permite realizar la consulta, mantenimiento e impresión de las duraciones de las consonantes bajo los siguientes criterios: Si está en sílaba acentuada, si está al inicio de una sílaba y si está en una sílaba antes de pausa.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TReglaDurCons (+) Constructor.	FormCreate	Crea la ventana de consulta y carga las duraciones de las consonantes de acuerdo a los criterios mencionados.
Guardar (+)	tbnGuardar Click	Guarda en el archivo de reglas de duraciones de consonantes los cambios hechos.

TABLA 3.2.1.2.19 DICCIONARIO DE DATOS DE LA CLASE TDURBASEV

Clase	Definición	
TDurBaseV	Es una subclase de la clase TMantBase. Permite realizar la consulta, mantenimiento e impresión de las duraciones base de las vocales.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TDurBaseV (+) Constructor.	FormCreate	Crea la ventana de consulta y carga las duraciones base de las vocales.
Guardar (+)	tbnGuardar Click	Guarda en el archivo de duraciones base de las vocales los cambios hechos por el usuario.

TABLA 3.2.1.2.20 DICCIONARIO DE DATOS DE LA CLASE TCATGRAM

Clase	Definición	
TCatGram	Es una subclase de la clase TMantBase. Permite realizar la consulta de las categorías gramaticales.	
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos propios.
Operaciones		
Descripción	Alias	Definición
TCatGram (+) Constructor	FormCreate	Crea la ventana de consulta y carga las categorías gramaticales.

4. Interfase del Lector: Agrupa las clases y funciones que sirven de enlace entre la parte externa del sistema, clases de los paquetes “Interfase de Usuario” y “Configuración y Mantenimiento” usados en los programas “Lector.exe” y “LectorLC.exe” con la librería “Lector.dll”, donde están implementadas las clases de los paquetes “Proceso Lingüístico-Prosódico” y “Procesamiento Digital de Señales”. El diagrama de clases de este paquete se muestra en la Figura 3.2.1.2.4.

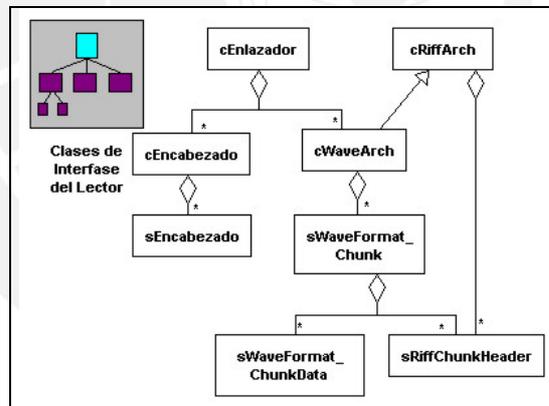


FIGURA 3.2.1.2.4 DIAGRAMA DE CLASES DE INTERFASE DEL LECTOR

Diccionario de datos:

TABLA 3.2.1.2.21 DICCIONARIO DE DATOS DE LA CLASE CENLAZADOR

Clase	Definición
cEnlazador	Esta clase controla la generación de las voces correspondientes a las frases que recibe de los programas (Lector.exe y LectorLC.exe). Para lo cual utiliza las clases llamadas de nivel alto (lingüística-prosódica) y las clases de nivel bajo (procesamiento digital de señales) de la librería Lector.dll.
Atributos	

Nombre	Alias	Definición
Nom_PathDatos (-) : char[1000]	-	Directorio donde se encuentran los archivos de datos usados por el sistema.
Nom_PathSalida (-) : char[1000]	-	Directorio donde se almacenan los archivos generados por la ejecución del sistema.
Arc_Wav (-) : cWaveArch *	-	Maneja los atributos y operaciones necesarios para poder crear, escribir y leer un archivo de sonido de formato WAVE.
Fra_Se (-) : cFrase *	-	Declaración de una instancia de la clase cFrase.
Mod_Dur (-) : UINT8	-	Indica si se va a modificar la duración de las demisílabas de la frase.
Mod_F0 (-) : UINT8	-	Indica si se va a modificar el contorno de frecuencia fundamental de la frase.
Operaciones		
Descripción	Alias	Definición
cEnlazador (+) Constructor	-	Adicionalmente a lo especificado en [Mateo 2000] crea un objeto cFrase que se usa para realizar el análisis lingüístico prosódico de las frases a sintetizar.
~cEnlazador (+) Destructor	-	Destruye los objetos cEncabezado y cFrase.
ObtenerDatos Símbolo (#) : Retorno: bool	-	Se agregó la funcionalidad de reconocer pausas incluidas dentro de la cadena de unidades básicas. Las pausas en la cadena de demisílabas tiene el siguiente formato “[c]” donde “c” es el carácter ASCII que hace de código de la pausa.
ProcesarUnidBasi (#) : Retorno: void	-	Controla la generación de las voces correspondientes a la cadena de demisílabas que recibe como parámetro.
UbicarDatos Símbolo (#) : Retorno: bool	-	Se agregó el manejo de las pausas como parte de la cadena de unidades básicas.
Sintetizar (+) : Retorno: void	-	Sintetiza y genera la voz correspondiente a la frase que recibe como parámetro.
EscribirPausaWav (#) :Retorno: void	-	Se encarga de generar las muestras necesarias para la pausa correspondiente al tiempo que recibe como parámetro.
ObtenerDir Eliminación (#) : Retorno: UINT8	-	Obtiene la dirección de eliminación de las muestras de señales de voz de la unidad básica que recibe como parámetro.
AgregarArcWav (+) : Retorno: void	-	Concatena los dos archivos de sonido en formato WAV que recibe como parámetro.

5. Funciones de Interfase de Lector: En el siguiente cuadro se describen las funciones del paquete de “Interfase del Lector” que sirven de interfaz de la

librería “Lector.dll”, es decir, sólo mediante estas funciones se puede acceder a la funcionalidad de la librería.

TABLA 3.2.1.2.22 DICCIONARIO DE DATOS DE LAS FUNCIONES DE INTERFASE

Nombre	Definición
IniciarSintetizador Retorno: long	Crea un objeto de la clase cEnlazador, cargando los archivos de datos del sistema en memoria para hacer más rápido su acceso.
LeerCadena Retorno: long	Genera la voz correspondiente a la cadena de caracteres que recibe como parámetro.
TerminarSintetizador Retorno: long	Elimina un objeto de la clase cEnlazador, liberando la memoria reservada para los datos.
Copiar2doWav Retorno: long	Concatena dos archivos de sonido de formato WAVE. Agregando el segundo archivo que recibe como parámetro en el primero.
CreaDemiSilabas Retorno: long	Permite generar el archivo de parámetros de predicción lineal y la señal de excitación multipulso (Archivo lpc).

- Clases de Proceso Lingüístico-Prosódico: Agrupa las clases que sirven para realizar el procesado lingüístico-prosódico de una frase para producir una secuencia de habla más natural. Estas clases permiten generar la secuencia de demisílabas de la frase, así como la duración que deben tener estas demisílabas en el contexto en que se presentan y el contorno de frecuencia fundamental que le corresponde a la frase. Su diagrama se muestra en la Figura 3.2.1.2.5.

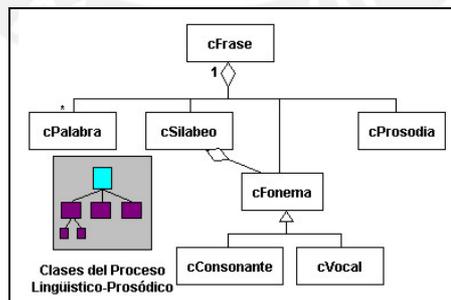


FIGURA 3.2.1.2.5 DIAGRAMA DE CLASES DEL PROCESO LINGÜÍSTICO-PROSÓDICO

Diccionario de datos

TABLA 3.2.1.2.23 DICCIONARIO DE DATOS DE LA CLASE CFRASE

Clase	Definición
cFrase	Es la abstracción de la “Frase” , que para nuestro caso se define como una serie de palabras limitadas por un delimitador llamado de fin de frase. Es la clase principal del sistema. Tiene como clase amiga a la clase cEnlazador.

Atributos		
Nombre	Alias	Definición
Nom_PathDatos (-) : char[1000]	-	Ruta donde se encuentran los archivos de datos.
Nom_PathSalida (-) : char[1000]	-	Ruta donde se encuentran los archivos de salida.
Arr_Abrev (-) : sRegAbrev *	-	Arreglo que contiene las abreviaturas y sus equivalentes en palabras expandidas.
Arr_Sigla (-) : sRegAbrev *	-	Arreglo que contiene las Siglas y sus equivalentes en palabras expandidas.
Arr_Atona (-) : sRegPalGram *	-	Arreglo que contiene palabras átonas y la categoría gramatical a la que pertenecen.
Arr_Ambiguo (-) : sRegAmbiguo *	-	Arreglo de palabras ambiguas en su acentuación.
Arr_CatGram (-) : sRegPalGram *	-	Arreglo que contiene las palabras y sus categorías gramaticales consideradas como excepciones gramaticales.
Arr_ReglaGra (-) : sReglaGra *	-	Arreglo que contiene las reglas gramaticales de contexto que se usan para hacer la desambiguación gramatical de las palabras no categorizadas definitivamente.
Arr_Term (-) : sRegPalGram *	-	Arreglo que contiene las terminaciones o desinencias tipo que identifican gramaticalmente a una palabra como perteneciente a cierta categoría gramatical.
Arr_Dur (-) : UINT16 *	-	Arreglo que guarda la duración calculada de cada demisílaba de acuerdo al contexto en que se encuentra dentro de la frase.
Arr_F0 (-) : TFLOAT *	-	Arreglo que contiene los valores numéricos que definen el contorno de F0 de la frase.
Tam_Abrev (-) : UINT16	-	Número de elementos del arreglo de abreviaturas.
Tam_Sigla (-) : UINT16	-	Número de elementos del arreglo de siglas.
Tam_Atona (-) : UINT16	-	Número de elementos del arreglo de palabras átonas.
Tam_Ambiguo (-) : UINT16	-	Número de elementos del arreglo de palabras ambiguas en su acentuación.
Tam_CatGram (-) : UINT16	-	Número de elementos del arreglo de excepciones gramaticales.
Tam_ReglaGra (-) : UINT16	-	Número de elementos del arreglo de reglas gramaticales.
Tam_Term (-) : UINT16	-	Número de elementos del arreglo de terminaciones o desinencias.
Dur_Total (-) : TUINT32	-	Duración total de la "Frase" en milisegundos.

Tam_F0 (-) : UINT16	-	Número de elementos del arreglo del contorno de frecuencia fundamental.
Fra_Letras (-) : char *	-	Representación de la frase procesada en grafemas.
Fra_Silabas (-) : char *	-	División de la frase procesada en sílabas.
Fra_Fonemas (-) : char *	-	Representación de la frase en fonemas.
Fra_Demisílabas (-) : char *	-	Representación de la frase en demisílabas.
Tip_Frase (-) : eTipoFrase	-	Indica el tipo de frase que se está procesando puede ser: enunciación, interrogativa, exclamativa y sus variantes.
Sig_NoFrase (-) : eTip_Palabra	-	Almacena el signo ortográfico final de la frase.
Num_Palabras (-) : UINT8	-	Cantidad de palabras que tiene la frase.
Num_Silabas (-) : UINT8	-	Número de sílabas que tiene la frase.
Num_DemiSil (-) : UINT16	-	Cantidad de demisílabas que tiene la frase.
Num_GrpFonico (-) : :UINT8	-	Cantidad de grupos fónicos en la que se divide la frase.
Con_Sonante (-) : cFonema *	-	Declaración de una instancia de la clase cFonema, usada para realizar el análisis de los grafemas de tipo "Consonante" de la frase.
Voc_AI (-) : cFonema *	-	Declaración de una instancia de la clase cFonema, usada para realizar el análisis de los grafemas de tipo "Vocal" de la frase.
Pro_Sodia (-) : cProsodia *	-	Objeto que agrupa los atributos y operaciones para calcular los parámetros prosódicos de la frase.
Sil_Abeo (-) : cSilabeo *	-	Objeto que agrupa los atributos y operaciones que permiten silabear la frase, y luego obtener su representación en demisílabas.
Pal_Abras (+) : cPalabra *	-	Conjunto de palabras de la que está compuesta la frase, está implementada como una lista doblemente enlazada.
Operaciones		
Descripción	Alias	Definición
Llenar_Datos (-) : Retorno: void	-	Carga el diccionario de datos del sistema.
Normalizar (-) : Retorno: void	-	Hace la normalización de la Frase, para lo cual divide la cadena Fra_Se y almacena las palabras en una lista doblemente enlazada de instancias de la clase cPalabra.

PreProceso (-) : Retorno: void	-	Expande las palabras especiales de la que está compuesta la frase. Hace la silabificación y acentuación fonética de la frase.
Categorizar (-) : Retorno: void	-	Hace la categorización gramatical de las palabras que componen la frase.
InsertarPausa (-) : Retorno: void	-	Inserta pausa en el texto que se encuentra entre dos pausas marcadas ortográficamente, cuando detecta que el texto podría producir una secuencia de habla demasiado larga que pueda causar una sensación de ahogo en el oyente.
AnalizarFrase (-) : Retorno: void	-	Analiza la frase para intentar descubrir el tipo o patrón melódico que le corresponde.
UbicarPalPausa (-) : Retorno: UINT8	-	Ubica la primera pausa a partir de la primera palabra que recibe como parámetro, en su segundo parámetro devuelve el puntero a la palabra con pausa.
PosInsPausa (-) : Retorno: cPalabra *	-	Devuelve la posición donde se debe colocar una pausa no marcada ortográficamente. Además, devuelve el código de la pausa a insertar.
UbicaConjCoord (-) : Retorno: cPalabra *	-	Trata de ubicar una conjunción coordinante entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.
UbicaConjSub (-) : Retorno: cPalabra *	-	Trata de ubicar una conjunción subordinante entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.
UbicaPronRel (-) : Retorno: cPalabra *	-	Trata de ubicar un pronombre relativo entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.
UbicaVerbo (-) : Retorno: cPalabra *	-	Trata de ubicar un verbo entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.
UbicaPreposicion (-) : Retorno: cPalabra *	-	Trata de ubicar una preposición entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.

UbicaPalFuncion (-) : Retorno: cPalabra *	-	Trata de ubicar una palabra función entre las dos palabras que recibe como parámetros, si lo encuentra verifica si cumple las condiciones para poder insertar una pausa. Si cumple devuelve un puntero a esa palabra, caso contrario devuelve un puntero a NULO.
CorregirAcentuacion (-) :Retorno: void	-	Corrige la acentuación fonética de las palabras de acuerdo a criterios prosódicos.
CargarPalabras (-) : Retorno: void	-	Se encarga de crear y añadir elementos a la lista de palabras de la que esta compuesta la Frase.
ReducirComplejidad (-) :Retorno: void	-	Llama una serie de métodos de la clase cPalabra para reducir la variabilidad de la frase.
PasarASilabas (-) : Retorno: void	-	Silabea las palabras de la frase.
PasarAUnidBasic (-) : Retorno: void	-	Primero obtiene la representación alofónica del texto de la frase, y en una segunda etapa obtiene su representación en Demisílabas.
PasarAFone (-) : Retorno: void	-	Pasa las palabras de la frase a su representación en fonemas (alófonos).
PasarADemisilabas (-) : Retorno: void	-	Pasa la frase que está en fonemas a su representación en demisílabas.
CalcularNumDemi (-) :Retorno: UINT16	-	Cuenta y devuelve el número de demisílabas que tiene la frase.
BuscarSilaba (-) : Retorno: bool	-	Busca la posición donde termina una sílaba y devuelve el separador con que termina.
AplicarFonSintactica (-) :Retorno: void	-	Aplica las reglas de desambiguación gramatical en las palabras que tienen más de una categoría gramatical asignada o no tienen ninguna.
GenDuracion (-) : Retorno: void	-	Genera la duración de las demisílabas de la frase de acuerdo a la información lingüística obtenida y al contexto en que se encuentra.
GenEntonacion (-) : Retorno: void	-	Genera el arreglo de contorno de frecuencia fundamental de la frase de acuerdo a la información lingüística obtenida.
Scanner (-) : Retorno: eTip_Palabra	-	Analizador léxico que tiene como función obtener e identificar las palabras de la cadena Fra_Se. Se implementa mediante un autómata finito no determinístico.
getc (-) : Retorno: INT8	-	Obtiene el siguiente caracter de la frase que se esta procesando en el analizador léxico "Scanner".
ungetc (-) : Retorno: void	-	Retrocede un carácter en la Fra_Se.
FinFrase (-) : Retorno: bool	-	Verifica si llegamos al final de la Frase.
fallo (-) : Retorno: INT8	-	Función de recuperación del analizador léxico "Scanner", le permite al Scanner continuar trabajando después de un fallo en la identificación de un lexema.

AvanzaIndice (-) : Retorno: void	-	Avanza el índice de las palabras de la frase tantas veces como se le indica en su parámetro.
NroGrpFonicos (-) : Retorno: UINT8	-	Devuelve el número de grupos fónicos de la que está compuesta la frase.
HayPronInterrogativo (-) : Retorno: bool	-	Verifica si existe un pronombre interrogativo en la frase.
EsEnumerativa (-) : Retorno: UINT8	-	Verifica si la frase enunciativa es de tipo enumerativa.
Imprime (-) : Retorno: void	-	Genera los archivos de resultados del sistema.
ImprimeDuracion (-) : Retorno: void	-	Escribe en el archivo que recibe las demisílabas de la frase y su duración.
ImprimeF0 (-) : Retorno: void	-	Escribe los valores del arreglo del contorno de frecuencia fundamental de la frase en un archivo.
EliminarNodo (-) : Retorno: void	-	Elimina una palabra de la lista de palabras de la frase.
IniFinUltSilaba (-) : Retorno: void	-	Encontrar las posiciones de inicio y final de la última sílaba según las posiciones de sus demisílabas en la Frase.
cFrase (+) : Constructor	-	Inicializa los arreglos y datos necesarios para realizar el procesado lingüístico-prosódico de la frase.
~cFrase (+) : Destructor	-	Destructor de la clase cFrase, elimina o libera espacio de memoria de los datos que tiene cargadas y el arreglo de palabras de la Frase.
Proc_Lingüístico (+) : Retorno: void	-	Realiza el análisis lingüístico-prosódico de la frase, con la finalidad de obtener la secuencia de demisílabas, la duración de estas y el Contorno de Frecuencia Fundamental(F0) de la frase.
EstablecerFrase (+) : Retorno: void	-	Establece la cadena de caracteres que recibe como parámetro como la unidad de trabajo. Copia el parámetro en el atributo Fra_Letras.
LimpiarFrase (+) : Retorno: void	-	Terminado de sintetizar la frase libera la memoria reservada para esta y sus palabras.
ObtenerTipoFrase (+) : Retorno: eTipoFrase	-	Devuelve el tipo de frase.

TABLA 3.2.1.2.24 DICCIONARIO DE DATOS DE LA CLASE CPALABRA

Clase	Definición
cPalabra	Agrupación de caracteres de la frase separados entre si por una sucesión de blancos, tabulados y nuevas líneas. No siempre son palabras propiamente dichas ya que podemos encontrarnos con signos ortográficos, abreviaturas, etc. Se representa mediante una "lista doblemente enlazada".

Atributos		
Nombre	Alias	Definición
Pal_Letras (-) : char *	-	Representación de la palabra en grafemas.
Pal_Silaba (-) : char *	-	Cadena de sílabas de la palabra.
Tip_Palabra (-) : eTip_Palabra	-	Tipo de palabra, puede ser: Palabra simple, Fecha, Hora, Moneda, Número, Nombre Propio, Abreviatura, Sigla y los signo ortográficos.
Pos_EnFrase (-) : UINT16	-	Posición de la palabra en la frase.
Mod_Escritura (-) : UINT8	-	Modo de escritura que tenía la palabra originalmente. (0): Todo en mayúscula. (1): inicial en mayúscula. (2): todo en minúscula. (3): variable.
Num_CatGram (-) : UINT8	-	Cantidad de categorías gramaticales asignadas a la palabra antes de su desambiguación.
Num_Silaba (-) : INT8	-	Cantidad de sílabas en la palabra.
Pos_Acento1 (-) : UINT8	-	Posición de la primera sílaba acentuada en la palabra.
Pos_Acento2 (-) : UINT8	-	Posición de la segunda sílaba acentuada en la palabra.
Tip_Acento (-) : UINT8	-	Tipo de acentuación que tiene la palabra. (0): Acento Ortográfico. (1): Acento Prosódico. (2): Átona. (3): Acento Ambiguo.
Est_Acento (-) : UINT8	-	Estado de la acentuación fonética de la palabra hecha por el sistema. (0): acentuación no definida. (1): definitivamente inacentuada. (2): definitivamente acentuada.
Num_Acentos (-) : UINT8	-	Número de acentos que tiene la palabra (0, 1 o 2).
Cod_Pausa (-) : UINT8	-	Código de pausa que precede a la palabra si es que no es un signo ortográfico, caso contrario, es el correspondiente al signo ortográfico.
Pal_Sig (+) : cPalabra *	-	Puntero a la siguiente palabra de la frase.
Pal_Ant (+) : cPalabra *	-	Puntero a la palabra anterior de la frase.
Cat_Gram (+) : eCat_Gramatica[5]		Arreglo con las categorías gramaticales asignadas a la palabra.
Operaciones		
Descripción	Alias	Definición
PasarAMinusculas (-) :Retorno: void	-	Pasa la palabra a minúscula y asigna el modo de escritura que tenía la palabra(Letras en mayúsculas, inicial en mayúscula, variable, etc.)

CreaSubLista (-) : Retorno: UINT8	-	Crea una lista de palabras con las palabras de la cadena pFrase que recibe. Retorna el número de elementos creados.
CreaNodo (-) : Retorno: void	-	Crea un nodo (elemento) de la lista de palabras de la frase.
InsertaSubLista (-): Retorno: void	-	Inserta una sublista de palabras a partir de la palabra actual dada como parámetro.
CatGramDeAtona (-) : Retorno: UINT8	-	Busca la categoría gramatical de la palabra átona en el diccionario léxico de palabras átonas.
CatGramDePal Gram (-) : Retorno: UINT8	-	Busca la categoría gramatical de la palabra que recibe como parámetro en el diccionario léxico de excepciones gramaticales.
CatGramDeTerm (-) : Retorno: UINT8	-	Verifica si la palabra que recibe como parámetro tiene alguna terminación definida en la tabla de terminaciones o desinencias. Si lo tiene devuelve la categoría gramatical que le corresponde.
DesambiguaCat Gram (-) : Retorno: void	-	Si la palabra tiene más de una categoría gramatical asignada trata de desambiguarla usando las reglas gramaticales de contexto.
UbicarRegla (-) : Retorno: bool	-	Ubica las reglas correspondientes a la categoría gramatical actual que se intenta desambiguar.
CumpleContexto (-): Retorno: bool	-	Verifica si la categoría gramatical actual asignada a la palabra cumple con el contexto, es decir, si cumple las reglas gramaticales de contexto.
CatGram Excepcion (-): Retorno: void	-	Maneja algunas excepciones a las reglas de categorización gramatical.
cPalabra (+) : Constructor	-	Inicializa los atributos de la palabra.
~cPalabra (+) : Destructor	-	Libera la memoria reservada al crear el objeto palabra.
ExpandeAbrev (+) : Retorno: UINT8	-	Expande la palabra tipo abreviatura o sigla que recibe como parámetro y con el resultado crea una sublista el cual inserta en el lugar de la abreviatura. Devuelve la cantidad de palabras en la que se expandió la abreviatura o sigla.
FechaALetras (+) : Retorno: UINT8	-	Si la palabra analizada es una fecha se pasa a su equivalente en letras, devuelve la cantidad de palabras en que se expandió la fecha.
HoraALetras (+) : Retorno: UINT8	-	Si la palabra es una hora se pasa a su equivalente en letras. Devuelve el número de palabras en que se expandió la hora.
DividePalComp (+) : Retorno: UINT8	-	Divide una palabra compuesta separada por un guión.
MonALetras (+) : Retorno: UINT8	-	Si la palabra es de tipo moneda lo pasa a su representación en palabras. Devuelve la cantidad de palabras en que se expandió.

NumALetras (+) : Retorno: UINT8	-	Si la palabra actual es un número se pasa a su equivalente en letras. Devuelve la cantidad de palabras en que se expandió el número.
SilabeaAcentua (+) :Retorno: UINT8	-	Silabea la palabra y luego la acentúa fonéticamente.
Acentuar (+) : Retorno: void	-	Aplica las reglas de acentuación fonética a la palabra.
CatGramX Diccionario (+) : Retorno: void	-	Intenta categorizar la palabra usando el diccionario de las palabras átonas, palabras gramaticales y el de las terminaciones.
CatGramXRegla (+) :Retorno: void	-	Hace la categorización de la palabra por reglas de contexto.
AsignarCatGram (+) :Retorno: void	-	Trata de asignar una categoría gramatical válida a la palabra usando las reglas de contexto.
ColocaPalTexto (+) :Retorno: void	-	Coloca la cadena de caracteres que recibe como parámetro como texto de la palabra.
ColocaTipoPalabra (+) :Retorno: void	-	Asigna el tipo de palabra.
ColocaPausa (+) : Retorno: void	-	Asigna el código de pausa que debe de ir antes que la palabra actual.
ColocaPosEnFrase (+) :Retorno: void	-	Asigna la posición de la palabra en la frase.
ColocaModo Escritura (+) : Retorno: void	-	Asigna el modo de escritura de la palabra.
ObtieneModo Escritura (+) : Retorno: UINT8	-	Devuelve el modo de escritura de la palabra.
ObtieneTipoPalabra (+) :Retorno: eTip_Palabra	-	Devuelve el tipo de palabra.
ObtienePalTexto (+) :Retorno: char *	-	Devuelve un puntero al texto de la palabra.
ObtienePalSilaba (+) :Retorno: char *	-	Devuelve un puntero a la cadena de palabra en sílabas.
ObtienePosEnFras e (+):Retorno: UINT8	-	Devuelve la posición que ocupa la palabra en la frase.
ObtieneNumCat Gram (+) : Retorno: UINT8	-	Devuelve la cantidad de categorías gramaticales asignada a la palabra.
ObtieneCodPausa (+) : Retorno: UINT8	-	Devuelve el código de la pausa que precede a la palabra.
ObtieneNumSilaba (+) :Retorno: UINT8	-	Devuelve el número de sílabas de la palabra.
ObtieneEstAcento (+) :Retorno: UINT8	-	Devuelve el estado de la acentuación de la palabra.

ObtieneNum Acentos (+): Retorno: UINT8	-	Devuelve el número de acentos que tiene la palabra.
ObtienePosAcento1 (+): Retorno: UINT8	-	Devuelve la posición del primer acento de la palabra.
ObtienePosAcento2 (+): Retorno: UINT8	-	Devuelve la posición del segundo acento de la palabra.

TABLA 3.2.1.2.25 DICCIONARIO DE DATOS DE LA CLASE CSILABEO

Clase		Definición
cSilabeo		Esta clase agrupa los atributos y métodos utilizados en la silabación y acentuación de las palabras de la frase, además, permite obtener la cadena de demisílabas (unidades básicas) usadas en la sintetización de la frase.
Atributos		
Nombre	Alias	Definición
Voc_AI (-) : cFonema *	-	Instancia de la clase abstracta cFonema que tiene toda la información necesaria para trabajar con las vocales.
Con_Sonante (-) : cFonema *	-	Instancia de la clase abstracta cFonema que tiene toda la información necesaria para trabajar con las consonantes.
Arr_Atona (-): sRegPalGram *	-	Puntero a la dirección de memoria del diccionario léxico de las palabras átonas.
Arr_Ambiguo (-) : sRegAmbiguo *	-	Puntero a la dirección de memoria del diccionario léxico de las palabras ambiguas.
Tam_Atona (-) : UINT16	-	Cantidad de elementos del diccionario léxico de las palabras átonas.
Tam_Ambiguo (-) : UINT16	-	Cantidad de elementos del diccionario léxico de las palabras ambiguas en su acentuación.
Operaciones		
Descripción	Alias	Definición
UbicarConsonante (-): Retorno: bool	-	Devuelve la ubicación de la primera consonante que encuentra en la cadena que recibe como parámetro.
UbicarVocal (-): Retorno: bool	-	Devuelve la ubicación de la primera vocal que encuentra en la cadena que recibe como parámetro.
UbicarUltVocal (-) : Retorno: void	-	Devuelve la ubicación de la última vocal que encuentra en la cadena que recibe como parámetro.
HayAcento (-): Retorno: bool	-	Verifica si en la cadena dada como argumento existe alguna sílaba acentuada.
SilabeaV (-) : Retorno: void	-	Devuelve una sílaba que comienza con una vocal(V).
SilabeaCV (-) : Retorno: void	-	Devuelve una sílaba que empieza con una consonante seguida de una vocal (CV)

SilabeaCC (-): Retorno: void	-	Devuelve una sílaba que empieza con dos consonantes consecutivas (CC).
V01 .. V08 (-): Retorno: bool	-	Son ocho funciones que verifican si se trata de sílabas de los tipos V, VC, VV, VVC, VhV, VCC, VVV, VVVC respectivamente. Son sílabas que inician con una vocal. (V: Vocal, C: Consonante, h: Consonante "h").
CV01 .. CV10 (-): Retorno: bool	-	Son diez funciones que verifican si se trata de sílabas de los tipos CV, CVC, CVCC, CVV, CVVC, CVVV, CVVVC, quV, quVC, quVVC respectivamente. Son sílabas que inician con una consonante y vocal. (V: Vocal, C: Consonante, q: Consonante "q", u: Vocal "u").
CC01 .. CC07 (-) Retorno: bool	-	Son siete funciones que verifican si se trata de sílabas de los tipos CCV, CCVC, CCVCC, CCVV, CCVVC, CCVhV, CCVVVC respectivamente. Son sílabas que inician con dos consonantes. (V: Vocal, C: Consonante, h: Consonante "h").
BuscarTilde (-): Retorno: UINT8	-	Devuelve la posición de la sílaba con tilde en la palabra que recibe como parámetro.
EsAcentoAmbiguo (-): Retorno: bool	-	Verifica si la palabra está en el diccionario léxico de palabras ambiguas.
EsAtona (-): Retorno: bool	-	Verifica si la palabra se encuentra en el diccionario léxico de palabras átonas.
ColocaTilde (-): Retorno: bool	-	Tilda la vocal acentuada de la palabra que recibe como parámetro de acuerdo a las reglas de acentuación del español.
cSilabeo (+): Constructor	-	Con los parámetros que recibe inicializa los sus atributos.
SilabeaFrase (+): Retorno: UINT8	-	Silabea la cadena (frase) que recibe como parámetro.
Silabear (+): Retorno: UINT8	-	Silabea un palabra.
DemiSilabear (+): Retorno: UINT8	-	Divide una cadena silabeada y transcrita a alófonos en demisílabas (unidades básicas).
Acentuar (+): Retorno: void	-	Coloca una tilde en las palabras con acentuación prosódica que no la tienen.
AplicarRegla Acento (+): Retorno: void	-	Utilizando las reglas de acentuación del castellano coloca una tilde en la sílaba apropiada de la palabra.

TABLA 3.2.1.2.26 DICCIONARIO DE DATOS DE LA CLASE CPROSODIA

Clase	Definición
cProsodia	Agrupar los atributos y métodos necesarios para calcular la prosodia correspondiente a la frase, se consideran dos variables: la duración segmental (alófonos y demisílabas) y el contorno de frecuencia fundamental.
Atributos	

Nombre	Alias	Definición
Vocal_Fuerte (-): char *	-	Cadena de caracteres que contiene los alófonos correspondientes a las vocales fuertes.
Vocal_Debil (-): char *	-	Cadena de caracteres que contiene los alófonos correspondientes a las vocales débiles.
Cons_Sorda (-): char *	-	Cadena de caracteres que contiene los alófonos correspondientes a las consonantes sordas.
Arr_VocalBase (-): sRegVocal *	-	Arreglo de duraciones base (iniciales) de las vocales.
Num_VocalB (-): UINT8	-	Número de elementos del arreglo de duraciones base de las vocales.
Arr_ConsBase (-): sRegDurCons *	-	Arreglo de duraciones base (iniciales) de los alófonos de las consonantes.
Num_ConsB (-): UINT8	-	Número de elementos del arreglo de duraciones base de los alófonos de las consonantes.
Arr_Pausa (-): sRegPausa *	-	Arreglo de duraciones correspondiente a las pausas.
Num_Pausas (-): UINT8	-	Número de elementos del arreglo de duraciones de las pausas.
Arr_ReglaCons (-): sRegReglaCons *	-	Arreglo de reglas de modificación de la duración consonántica.
Num_ReglaCons (-): UINT8	-	Número de elementos del arreglo de reglas de la duración consonántica.
Operaciones		
Descripción	Alias	Definición
Cargar_BaseVocal (-): Retorno: void	-	Carga en memoria las duraciones base de las vocales.
Cargar_BaseCons (-): Retorno: void	-	Carga en memoria las duraciones base de las consonantes.
Cargar_Pausa (-): Retorno: void	-	Carga en memoria las duraciones de las pausas.
Cargar_ReglaCons (-): Retorno: void	-	Carga en memoria las reglas de modificación de las duraciones consonánticas.
DuracionBaseVocal (-): Retorno: UINT16	-	Devuelve la duración base correspondiente a la vocal que recibe como parámetro.
DuracionBaseCons (-): Retorno: UINT16	-	Devuelve la duración base correspondiente a la consonante que recibe como parámetro.
DuracionSemiLarga (-): Retorno: UINT16	-	Devuelve la duración correspondiente a una vocal semilarga.
SilabaFinVocal (-): Retorno: bool	-	Verifica si la vocal esta en posición final de sílaba.
EsVocalFuerte Anterior (-): Retorno: bool	-	Verifica si el alófono anterior es una vocal fuerte.

EsVocalFuerte Siguiente (-): Retorno: bool	-	Verifica si el alófono siguiente es una vocal fuerte.
EsSordoAnterior (-): Retorno: bool	-	Verifica si el alófono anterior es una consonante sorda.
EsVocalAnterior (-): Retorno: bool	-	Verifica si el alófono anterior es una vocal.
EsVocalSiguiente(-): Retorno: bool	-	Verifica si el alófono siguiente es una vocal.
EstaAnteriorEn (-): Retorno: bool	-	Verifica si el alófono anterior esta dentro de la cadena pFuente que recibe como parámetro.
EstaSiguienteEn (-): Retorno: bool	-	Verifica si el alófono siguiente está dentro de la cadena pFuente que recibe como parámetro.
CopiaAlofSiguiente (-): Retorno: void	-	Devuelve el alófono siguiente de la cadena de alófonos de la frase.
CopiaAlofAnterior (-): Retorno: void	-	Devuelve el alófono anterior de la cadena de alófonos de la frase.
FormaGrpCons (-): Retorno: bool	-	Verifica si la cadena de fonemas que recibe forma un grupo consonántico.
PrePausal (-): Retorno: bool	-	Verifica si el alófono se encuentra en posición inmediatamente anterior a una pausa.
UbicarPrimerAcento (-): Retorno: TINT32	-	Devuelve la posición de la primera sílaba acentuada en la frase.
UbicarUltAcento (-): Retorno: TINT32	-	Devuelve la posición de la última sílaba acentuada en la frase.
UbicarPausaFin (-): Retorno: TINT32	-	Devuelve la posición de la demisílaba que representa a la pausa de fin de grupo fónico.
UbicarPos Parentesis (-): Retorno: UINT8	-	Devuelve la ubicación del grupo parentético (texto entre paréntesis) en relación a los otros grupos fónicos de la frase.
LlenaPausaF0 (-): Retorno: void	-	Llena de ceros el arreglo de F0 en la parte correspondiente a una pausa.
BorraHastaPausa (-): Retorno: void	-	Elimina las demisílabas de la cadena de demisílabas hasta encontrar una pausa.
PatronF01 (-): Retorno: void	-	Impone el contorno de frecuencia fundamental (F0) correspondiente al patrón o modelo 1.
PatronFOA .. PatronF0F (-): Retorno: void	-	Son seis funciones que imponen el contorno de F0 correspondiente a los patrones o modelos A, B, C, D, E, F respectivamente.
InterpolaF0 (-): Retorno: void	-	Interpola los puntos marcados al definir los patrones melódicos para generar el contorno de F0 con un muestreo uniforme.
InterpolaPuntos (-): Retorno: void	-	Une dos puntos del contorno de frecuencia fundamental.

UnePuntos (-): Retorno: void	-	Une dos puntos del contorno de frecuencia fundamental utilizando el algoritmo para unir dos puntos de una recta.
EstaAcentuado (-): Retorno: bool	-	Verifica si la sílaba es acentuada.
InicioSilaba (-): Retorno: bool	-	Verifica si el alófono que recibe como parámetro esta al inicio de una sílaba.
EstaDosVocales (-): Retorno: bool	-	Verifica si el alófono que recibe como parámetro esta entre dos vocales.
cProsodia (+): Constructor	-	Inicializa los arreglos y atributo a usar por la instancia de la clase.
~cProsodia (+): Destructor	-	Libera la memoria reservada por la instancia de la clase.
F0 (+): Retorno: void	-	Genera el patrón del contorno de frecuencia fundamental correspondiente a la frase.
DuracionVocal (+): Retorno: UINT16	-	Devuelve la duración que le corresponde a una vocal en la frase.
Duracion Consonante (+): Retorno: UINT16	-	Devuelve la duración que le corresponde a una consonante en la frase.
DuracionPausa (+): Retorno: UINT16	-	Devuelve la duración de la pausa que recibe como parámetro.

TABLA 3.2.1.2.27 DICCIONARIO DE DATOS DE LA CLASE CFONEMA

Clase		Definición
cFonema		Representa a los sonidos que se pueden producir con los órganos articulatorios y que cada idioma tiene seleccionado para su uso en el habla. Es una clase abstracta que se usa como superclase de cVocal y cConsonante.
Atributos		
Nombre	Alias	Definición
-	-	No tiene atributos
Operaciones		
Descripción	Alias	Definición
-	-	No tiene operaciones

TABLA 3.2.1.2.28 DICCIONARIO DE DATOS DE LA CLASE CCONSONANTE

Clase		Definición
cConsonante		Representa a las consonantes y agrupa los atributos y métodos necesarios para su manejo en el sistema. Es una subclase de la clase cFonema.
Atributos		
Nombre	Tipo	Definición
Con_Sonante (-): char *	-	Lista de consonantes considerados en el sistema.
Grp_Consoni (-): char *	-	Lista de los grupos consonánticos iniciales.

Grp_ConsFin (-): char *	-	Lista de los grupos consonánticos finales.
Alo_Fono (-): sAlofono *	-	Reglas de transcripción alofónica.
Num_Alofono (-): UINT16	-	Número de reglas de transcripción alofónica.
Fon_Sintac (-): sAlofono *	-	Reglas para la fonetización sintáctica.
Num_FonSintac (-): UINT16	-	Número de reglas de fonetización sintáctica.
Operaciones		
Descripción	Alias	Definición
BuscarCaracter (-): Retorno: bool	-	Busca un caracter en una cadena y devuelve su posición en la cadena.
RevisarGrafema (-): Retorno: bool	-	Revisa si el grafema examinado es igual al del alófono correspondiente a la regla actual.
RevisarAnterior (-): Retorno: bool	-	Verifica si los grafemas anteriores al grafema actual cumplen con las condiciones necesarias para la aplicación del alófono.
RevisarPosterior (-): Retorno: bool	-	Verifica si los grafemas posteriores al grafema actual cumplen con las condiciones necesarias para la aplicación del alófono.
ProbarCondicion (-): Retorno: bool	-	Comprueba si los caracteres anteriores o posteriores cumplan con la condición dada.
EvaluaORGrafema (-):Retorno: bool	-	Según el formato de las reglas los elementos de las condiciones en paréntesis si están separadas por espacios en blanco están unidas por el operador AND, y en cada elemento si hay más de una condición están se consideran unidas por el operador lógico OR. Este método analiza si se cumplen las condiciones OR de la regla cuando se trata de grafemas.
EvaluaORAlofono (-):Retorno: bool	-	Tiene la misma funcionalidad que la función anterior, pero para el caso de los alófonos.
ObtieneCond Compuesta (-) : Retorno: bool	-	En las reglas las condiciones compuesta, que son grafemas con más de una letra, se separan con el caracter , este método devuelve ese grafema compuesto.
CargarReglas (-): Retorno: bool	-	Carga en los arreglos respectivos las reglas de alófonos y fonetización.
cConsonante (+): Constructor		Inicializa los atributos de su instancia y carga los arreglos de las reglas de fonetización.
cConsonante (+): Constructor	-	Inicializa los atributos de la instancia de la clase cConsonante, pero sin llenar los arreglos de las reglas de fonetización.
~cConsonante (+): Destructor	-	Libera la memoria reserva para los arreglos de las reglas de fonetización.

AplicarRegla (+): Retorno: bool	-	Verifica si el alófono candidato a reemplazar al grafema actual del texto cumple con las condiciones necesarias para este fin. Si cumple devuelve el símbolo fonético que le corresponde al grafema, caso contrario, devuelve nulo.
EsGrpConslni (+): Retorno: bool	-	Verifica si los caracteres que recibe como parámetro pertenecen a la lista de consonantes que forman un grupo consonántico inicial.
EsGrpConsFin (+): Retorno: bool	-	Verifica si los caracteres que recibe como parámetro pertenecen a la lista de consonantes que forman un grupo consonántico final.
EsConsonante (+): Retorno: bool	-	Verifica si el caracter que recibe es una consonante.

TABLA 3.2.1.2.29 DICCIONARIO DE DATOS DE LA CLASE CVOCAL

Clase		Definición
cVocal		Representa a las vocales. Es una subclase de la clase abstracta cFonema.
Atributos		
Nombre	Alias	Definición
Voca_Noac (-): char *	-	Lista de vocales no acentuadas.
Voca_Acen (-): char *	-	Lista de vocales acentuadas.
Voca_Fuer (-): char *	-	Lista de vocales fuertes (a, e, o, á, é, ó, í, ú).
Voca_Debi (-): char *	-	Lista de vocales débiles(i, u, ü).
Dip_Tongo (-): char *	-	Lista de diptongos.
Trip_Tongo (-): char *	-	Lista de triptongos.
Operaciones		
Descripción	Alias	Definición
cVocal (+): Constructor	-	Inicializa los atributos que son propiedades de esta clase.
Acentuar (+): Retorno: void	-	Recibe una vocal no acentuada y la reemplaza por su equivalente acentuada.
EsVocal (+): Retorno: bool	-	Verifica si el caracter que recibe como argumento es una vocal.
EsVocalAcentuada (+): Retorno: bool	-	Verifica si el caracter que recibe como argumento es una vocal acentuada.
EsVocalDebil (+): Retorno: bool	-	Verifica si el caracter que recibe es una vocal débil.
EsVocalFuerte (+): Retorno: bool	-	Verifica si el carácter que recibe es una vocal fuerte.

EsDiptongo (+): Retorno: bool	-	Verifica si la cadena que recibe es un diptongo.
EsTriptongo (+): Retorno: bool	-	Verifica si la cadena que recibe es un triptongo.
ObtenerSimbolo (+): Retorno: void	-	Devuelve el alófono que le corresponde a una vocal dependiendo si es acentuada o inacentuada.

7. Clases del Procesamiento Digital de Señales: Agrupa las clases que se usan en la codificación de las muestras de voz (demisílabas) y en la síntesis de voz (lectura de frases). Estas clases son descritas en detalle en [Mateo 2000], en este trabajo se describirán las funciones añadidas y modificadas para soportar la nueva información prosódica y lingüística descritas en los paquetes anteriores. Su diagrama se muestra en la Figura 3.2.1.2.8.

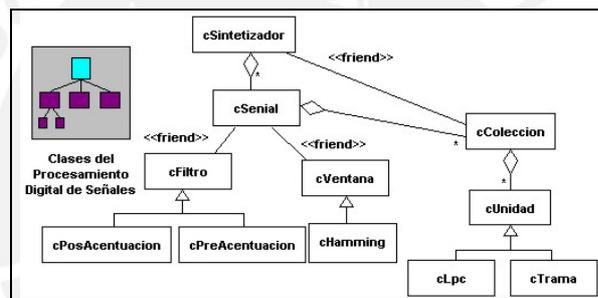


FIGURA 3.2.1.2.8 DIAGRAMA DE CLASES DE CODIFICACIÓN Y SÍNTESIS DE VOZ

Diccionario de datos

TABLA 3.2.1.2.30 DICCIONARIO DE DATOS DE LA CLASE CSINTETIZADOR

Clase	Definición	
cSintetizador	Es la clase interfaz del paquete de procesamiento digital de señales. Maneja el análisis(codificación) de las muestras de voz y la síntesis de las frases. Tiene como clase amiga a la clase cEnlazador.	
Atributos		
Nombre	Alias	Definición
Sen_lal (-): cSenial *	-	Declaración de una instancia de la clase cSenial usada en los procesos de análisis y síntesis de VOZ.
Mod_Dur (-): UINT8	-	Indica si va a modificar la duración de las demisílabas según el arreglo de duraciones del módulo prosódico.
Mod_F0 (-): UINT8	-	Indica si se va a modificar la variación de la frecuencia fundamental de la frase usando el arreglo de contorno de frecuencia fundamental.
Operaciones		
Descripción	Alias	Definición

OrdenaPulsos (-): Retorno: void	-	Ordena el arreglo de ubicaciones de la señal de excitación calculada en el análisis (codificación) de voz antes de almacenarla en el archivo lpc.
FormarSerial (-): Retorno: void	-	Obtiene en base a la información del archivo lpc todos los datos de la señal y las tramas. En este procedimiento se llama a la función que modifica la distancia entre pulsos de la señal de excitación para modificar la frecuencia fundamental.
ModificarF0 (-): Retorno: void	-	Modifica la distancia de las ubicaciones de los pulsos de la señal de excitación multipulso.
cSintetizador (+) : Constructor	-	Inicializa los atributos de la instancia.
Sintetizar (+): Retorno: void	-	Obtiene la señal de voz correspondiente a una unidad básica de la frase procesada a partir de los datos del filtro predictor almacenados en el archivo lpc. Modifica la duración de la señal de la unidad básica de acuerdo a la duración calculada en el módulo prosódico y manda aplicar el contorno de F0 que le corresponde.

TABLA 3.2.1.2.31 DICCIONARIO DE DATOS DE LA CLASE CSENIAL

Clase		Definición
cSenial		Abstrae el concepto de una señal digitalizada, formada por un conjunto de muestras que están divididas en tramas. Una señal , que puede ser una palabra, o una unidad básica, debidamente digitalizada, esta formado por varias tramas ,en nuestro caso de 8ms con overlap de 4ms. Tiene como clases amigas a cVentana, cSintetizador y cEnlazador.
Atributos		
Nombre	Alias	Definición
-	-	No se agregaron atributos nuevos.
Operaciones		
Descripción	Alias	Definición
Decimar (-): Retorno: void	-	Realiza la decimación (eliminación) de muestras de la señal de voz de la unidad básica.
Interpolar (-): Retorno: void	-	Usando interpolación lineal incrementa el número de muestras de la señal de voz de la unidad básica.
AjustaSerial (-): Retorno: void	-	Elimina las muestras de la señal de voz de la unidad básica de su inicio y final, según el parámetro que indica la dirección de eliminación.

3.3.1.3 Diagrama de Estados

Este diagrama muestra los estados por los cuales pasa la “Frase” en la generación de su señal acústica (voz).

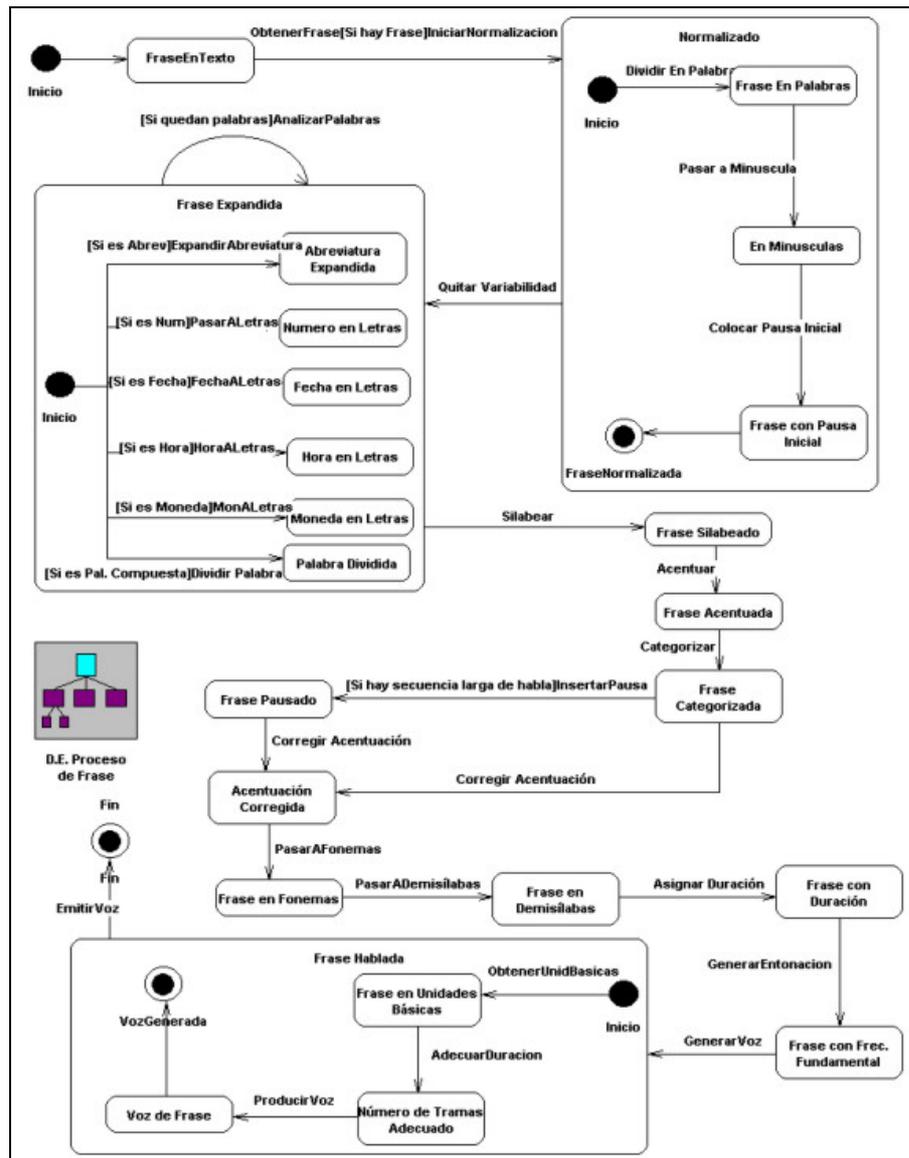


FIGURA 3.2.1.3.1 DIAGRAMA DE ESTADOS DE LA FRASE

3.3.2 Diseño Técnico

3.3.2.1 Diagramas de Secuencia

Estos diagramas muestran la interacción de los objetos del sistema conversor texto-voz y los mensajes que se intercambian entre ellos.

1. Configurar sistema: Muestra el intercambio de mensajes que se produce para ver y/o modificar la configuración del sistema.

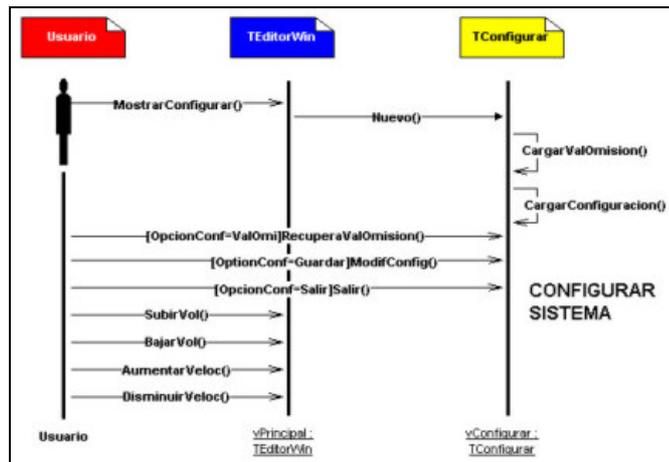


FIGURA 3.2.2.2.1 DIAGRAMA DE SECUENCIA CONFIGURAR SISTEMA

2. Manipular lectura: Muestra el intercambio de mensajes que se dan entre los objetos del sistema cuando se usa las opciones del sistema.

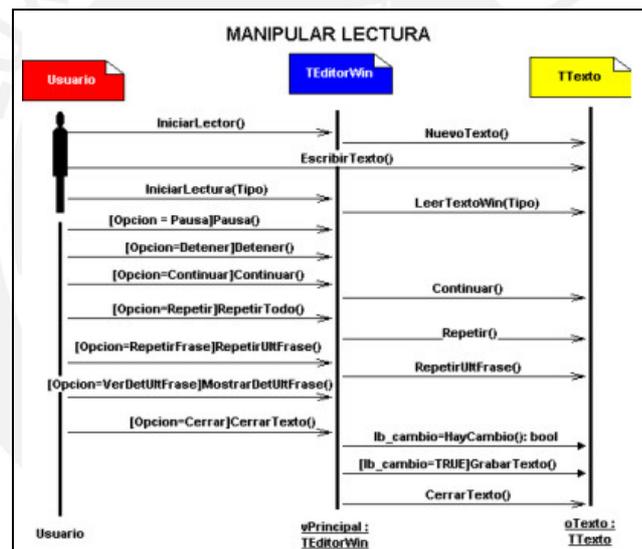


FIGURA 3.2.2.2.2 DIAGRAMA DE SECUENCIA MANIPULAR LECTURA

3. Leer Texto: Se muestran tres diagramas en los que se observa toda la interacción del usuario y el sistema para generar la voz de un texto.

➤ Interface de Usuario Entorno Windows:

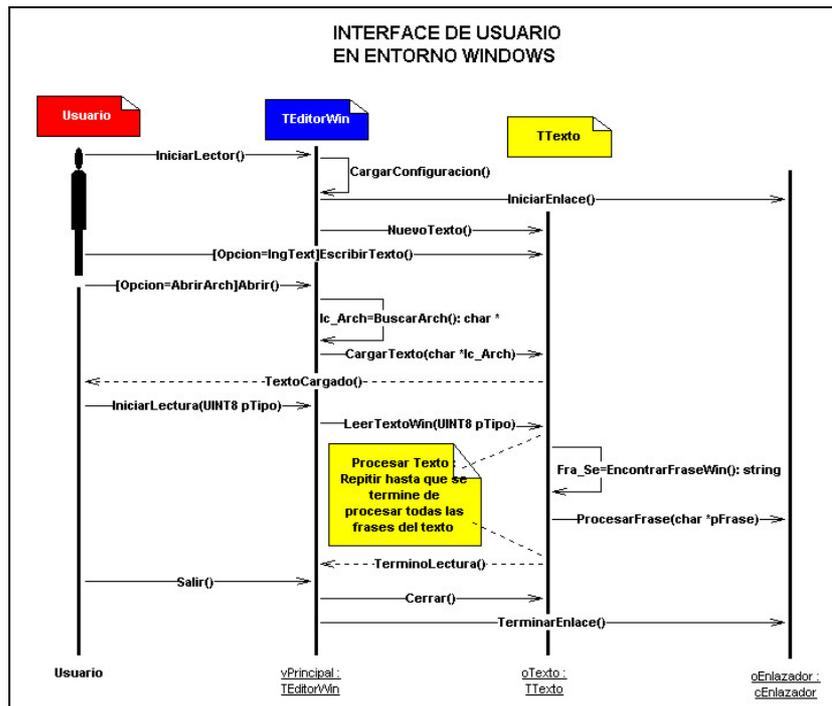


FIGURA 3.2.2.3 DIAGRAMA DE SECUENCIA INTERFACE DE USUARIO ENTORNO WINDOWS

➤ Interface de usuario por línea de comandos:

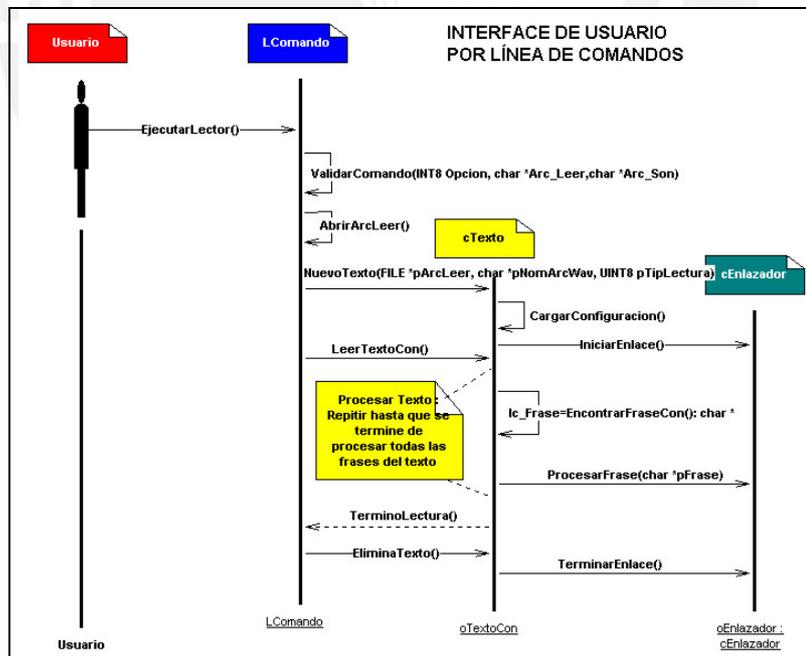


FIGURA 3.2.2.4 DIAGRAMA DE SECUENCIA INTERFACE DE USUARIO POR LÍNEA DE COMANDOS

➤ Proceso Lingüístico-Prosódico: Muestra la secuencia de mensajes entre objetos del sistema para realizar el proceso lingüístico-prosódico de la frase que recibe, y a partir de este análisis generar la secuencia

de demisílabas y los parámetros prosódicos que se usarán al momento de generar las señales de voz de la frase.

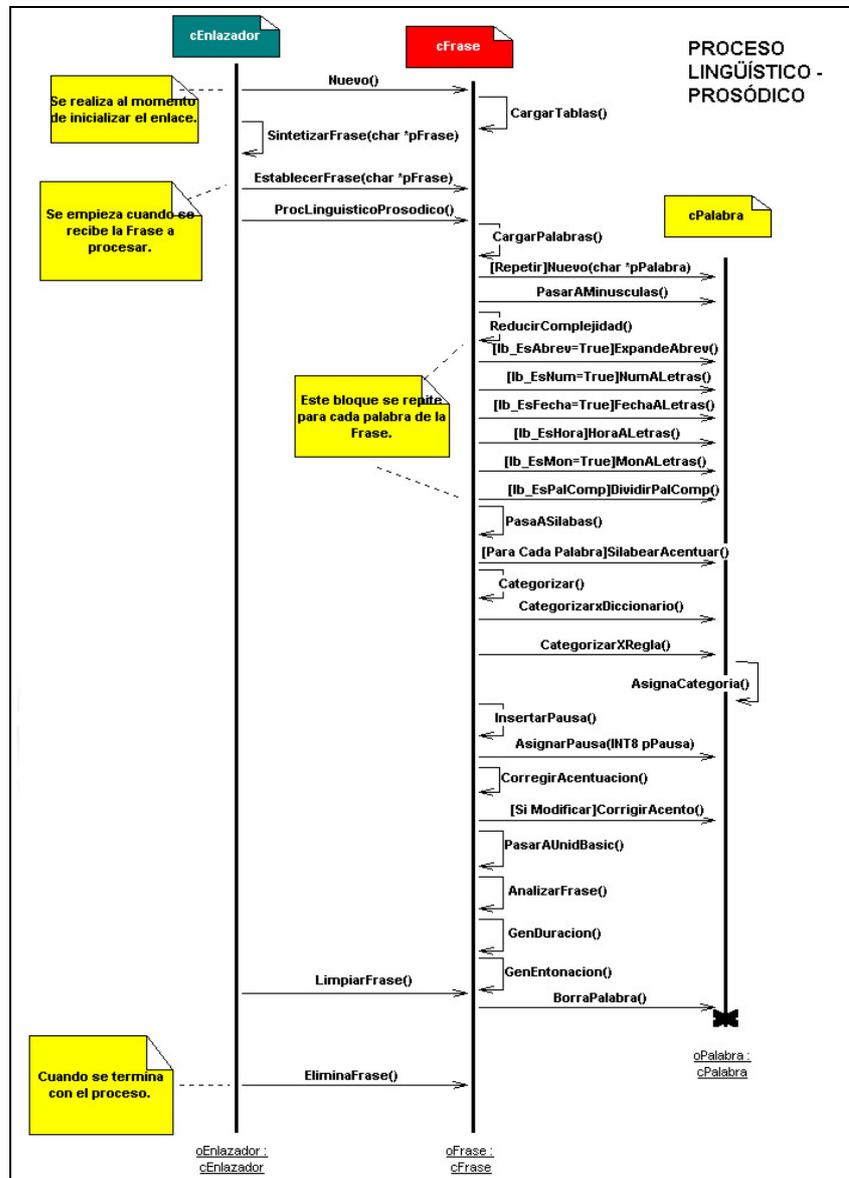


FIGURA 3.2.2.5 DIAGRAMA DE SECUENCIA DEL PROCESO LINGÜÍSTICO-PROSÓDICO

- Sintetización de Voz: Este diagrama muestra la interacción entre las instancias de las clases usadas en la sintetización de la voz de las frases y cuya funcionalidad se agregó o modificó para poder aplicar las modificaciones prosódicas (duración de demisílabas y contorno de frecuencia fundamental de la frase).

4

Conclusiones, Observaciones y Recomendaciones

4.1 Conclusiones y Observaciones

1. De acuerdo a las pruebas de audición realizadas cuyos resultados se muestran en el Anexo I y de las observaciones dadas por las personas que participaron en estas, se puede concluir que se obtuvo un sistema conversor de texto a voz que genera voz sintética con una inteligibilidad y naturalidad buenas. Y con el acento de un hablante limeño.
2. Se concluye que el análisis lingüístico-prosódico juega un papel importante en la generación de una voz sintética más natural. Debido a que enriquece el texto con información relevante que nos permite conocer la forma de onda de la señal acústica resultante y la duración de sus segmentos.
3. Así mismo, se concluye que tanto el procesado lingüístico de un texto como el procesado acústico dentro del desarrollo de un sistema conversor texto voz, deben en la medida de lo posible desarrollarse en paralelo, debido a que ambas están muy interrelacionadas. Además, de esta manera se puede sacar mayor provecho a la interdisciplinariedad que presenta la Tecnología del Habla.
4. Del análisis prosódico realizado, se concluye que este es muy dependiente del locutor donante de la voz y que no se puede tomar los estudios encontrados en la literatura referente al tema más que como una base o guía para llevar a cabo un estudio propio para el sistema de acuerdo a la persona donante de la voz.
5. El desarrollo orientado a objetos desde el análisis hasta la programación usada en el presente trabajo, resultó ser el adecuado ya que nos permitió estructurar el sistema en una forma ordenada y clara, utilizando sus

componentes básicos, como son la encapsulación, la herencia y el polimorfismo, así como el uso de interfaces para el acceso a los métodos de las clases. Con lo que se facilita las futuras modificaciones al sistema, como era nuestro segundo objetivo.

4.2 Recomendaciones

1. En este trabajo se ha abordado todas las tareas mínimas de las cuales debe estar compuesta un sistema conversor texto-voz, pero debido a lo limitado de los recursos con que se disponía para desarrollar el trabajo, principalmente el tiempo, no se ha realizado un estudio detallado de cada tarea. Con la finalidad de mejorar la funcionalidad de cada tarea e incluso buscar algoritmos más eficientes a los usados, se recomienda agrupar las tareas en subgrupos afines y presentarlos como futuros trabajos de Tesis, además, se recomienda tratar de formar grupos con integrantes de diferentes especialidades, ya que como se ha mencionado la Tecnología del Habla es multidisciplinar donde confluyen conocimientos de diferentes ramas del saber.
2. La librería (lector.dll) que contiene clases que realizan el estudio lingüístico-prosódico y el procesado acústico, puede ser utilizada por otras aplicaciones desarrolladas en sistemas operativos Windows usando las funciones de interfase definidas.
3. Basado en este trabajo se puede intentar desarrollar sistemas que emitan voz sintética que expresen elementos suprasegmentales de un texto, como son las emociones (tristeza, alegría, dolor, etc.) para lo cual se debe tener un módulo semántico que nos pueda indicar el sentido que tienen las frases.
4. Se recomienda el estudio de nuevos modelos de análisis y síntesis de voz que nos permitan tener mayor control sobre la forma del espectro de la señal acústica que nos permitan aplicar los parámetros prosódicos provenientes del módulo prosódico con mayor facilidad y flexibilidad.
5. La calidad de las palabras grabadas usadas para obtener las demisílabas juegan un papel importante en la calidad de la voz generada por los sistemas

conversores de texto-voz. En este trabajo se uso la voz del tesista pero lo ideal sería contar con una persona que tenga buena pronunciación como un locutor profesional, pero como la obtención de estas voces puede tomar bastante tiempo (varias sesiones), debido a que hay que hacer pruebas y adaptar el sistema a las características de la voz del donante, se recomienda para próximos trabajos contar con un presupuesto para este fin.



BIBLIOGRAFÍA

- [Aarcos] “Gramática de la lengua española”. Aarcos Llorach, Emilio.
- [Cabral 1978] “La fuerza de las palabras”. Manuel Cabral. Ed. Selecciones del Reader’s Digest. 1978.
- [Díaz 1970] “Lengua española 6”. Dra. Delia Díaz de Villar. Ed. Edime Organización Gráfica. 1970.
- [Hernández 1996] “Gramática funcional del español”. Hernández Alonso, César. 1996.
- [Laura 1999] “La duración consonántica en Castellano”. Laura del Barrio Estévez y Sergio Tornel Castells. Estudios de Lingüística de la Universidad Autónoma de Barcelona, Pág. 9-35 vol. 13. 1999.
- [López 1993] “Estudio de técnicas de procesado lingüístico y acústico para sistemas de conversión texto-voz en español basados en concatenación de unidades”. E. López Gonzalo. Tesis Doctoral. Universidad Politécnica de Madrid. Julio 1993.
- [Mateo 2000] “Conversión de texto a voz mediante la concatenación de demisílabas usando predicción lineal con entrada multipulso”. Gamál Mateo Cárdenas. “Tesis de titulación”.Junio 2000.
- [Mora 1998] “Entonación”. Elsa Mora. Revista de Español Vivo. Español Actual Pág. 43-49 vol. 69. 1998.
- [Navarro 1948] “Manual de entonación española”. T. Navarro Tomás. Segunda edición, corregida. Hispanic Institute. Nueva York 1948.
- [Navarro 1972] “Manual de pronunciación española”. T Navarro Tomás. Decimoséptima edición. Madrid, 1972.
- [Ríos 1999] “La transcripción fonética automática del diccionario electrónico de formas simples flexivas del español: Estudio fonológico en el léxico”. Antonio Ríos Mestre. Estudios de lingüística española. Volumen 4. 1999.
- [Torrejón 1999] “Introducción a la fonética y fonología españolas”. Alfredo Torrejón. 1999.
- [Charte 1998] “Programación con Delphi 4”. Francisco Charte. 1998.
- [Larman 1999] “UML y Patrones”. Craig Larman. Prentice Hall, Mexico 1999.
- [Schildt 1995] “Programación en C y C++ en Windows 95”. Herbert Schildt. McGraw-Hill. 1995.
- [Figuroa 1998] “Tesis de licenciatura: Síntesis de Voz en español, un enfoque silábico”. Morelia, Mich. 1998.

Anexo A

Diagrama de Transiciones del Automata Finito Usado en la Normalización de la Frase

En este anexo se muestran los diagramas de transiciones usados por el sistema para el reconocimiento de las tramas de los componentes léxicos(tokens) dentro de lo que hemos llamado la unidad de trabajo (Frase). Es decir, todos los tipos de palabras considerados.

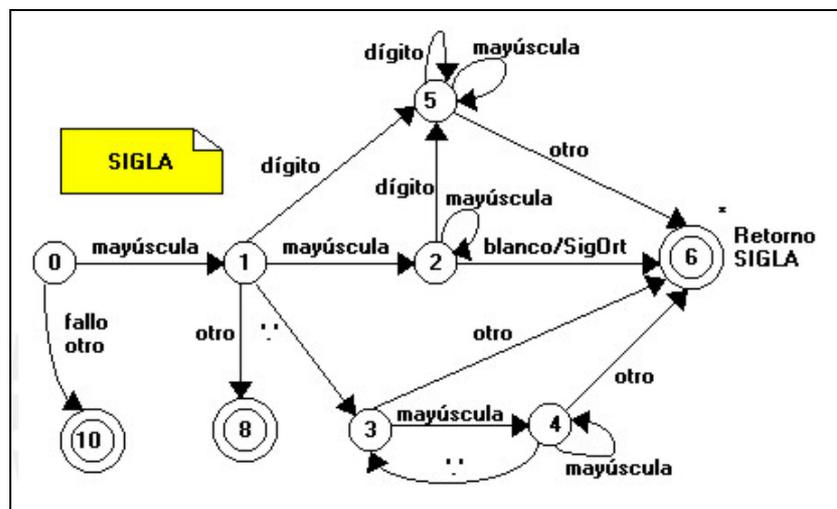


FIGURA A.1 PARA EL RECONOCIMIENTO DE SIGLAS

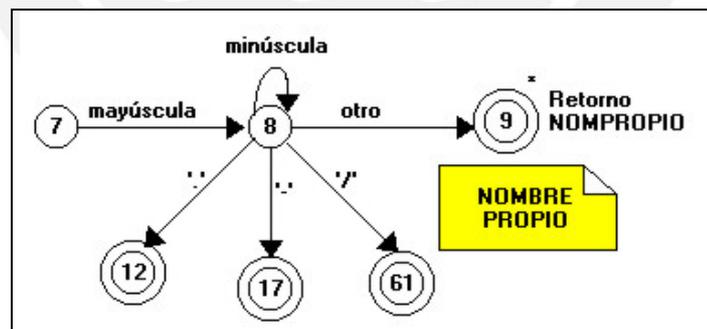


FIGURA A.2 PARA EL RECONOCIMIENTO DE NOMBRES PROPIOS

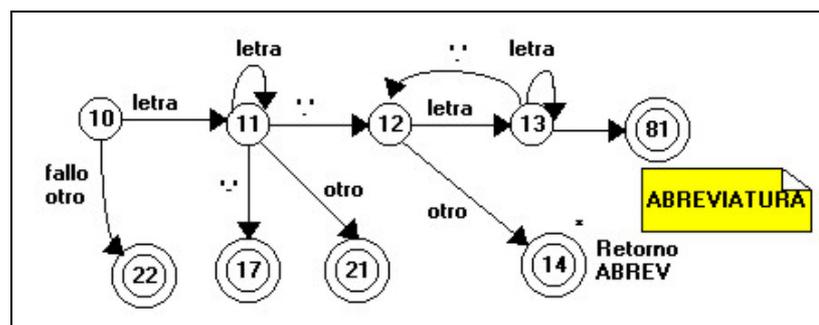


FIGURA A.3 PARA EL RECONOCIMIENTO DE ABREVIATURAS

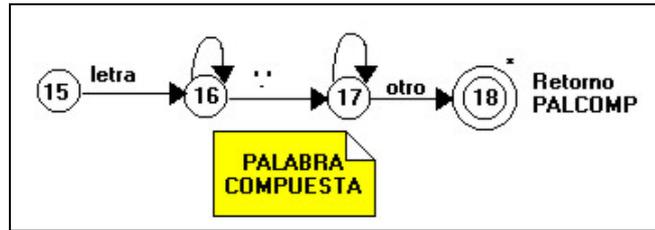


FIGURA A.4 PARA EL RECONOCIMIENTO DE PALABRAS COMPUESTAS

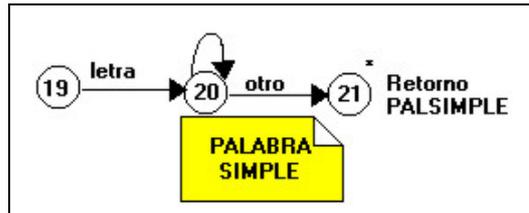


FIGURA A.5 PARA EL RECONOCIMIENTO DE PALABRAS SIMPLES

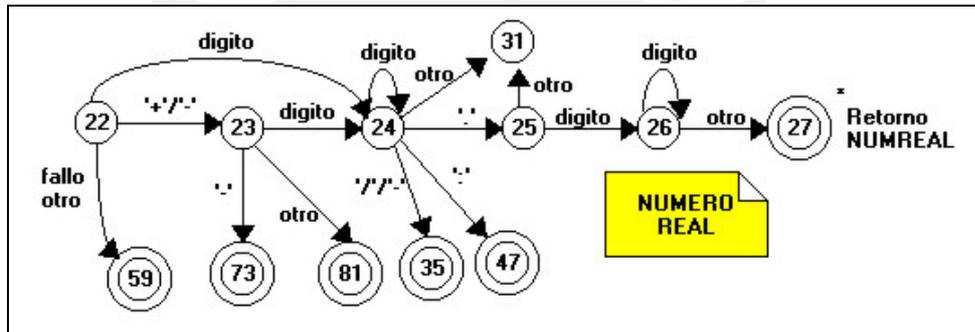


FIGURA A.6 PARA EL RECONOCIMIENTO DE NÚMEROS REALES CON SIGNO

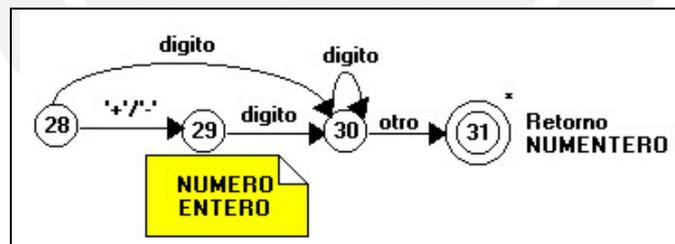


FIGURA A.7 PARA EL RECONOCIMIENTO DE NÚMEROS ENTEROS CON SIGNO

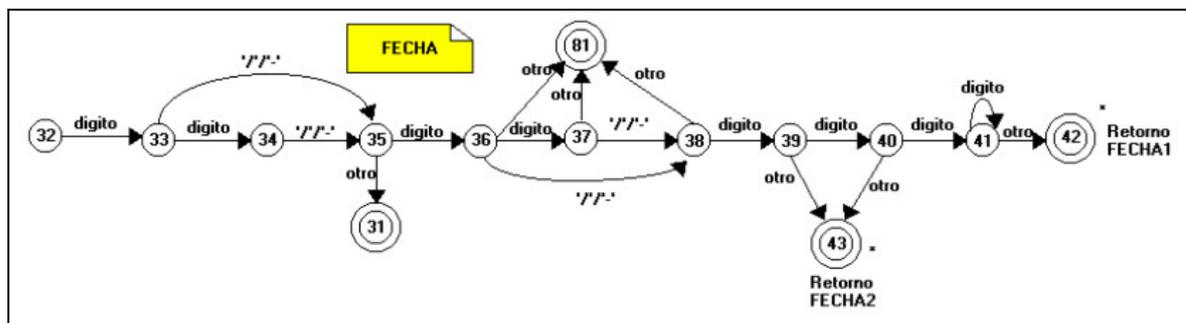


FIGURA A.8 PARA EL RECONOCIMIENTO DE FECHAS

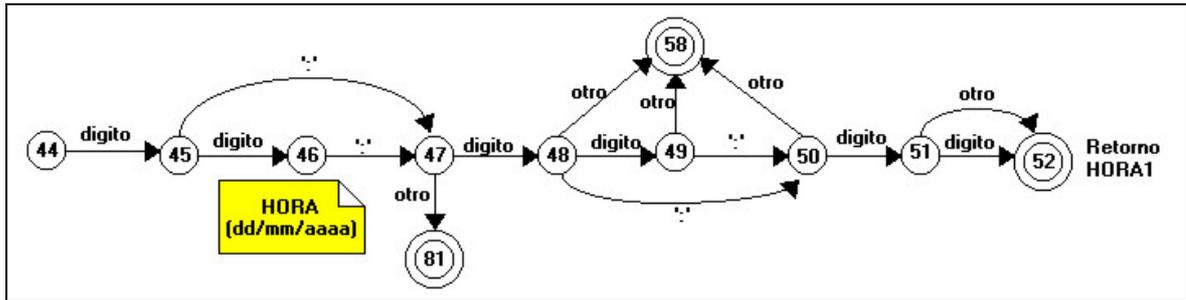


FIGURA A.9 PARA EL RECONOCIMIENTO DE HORAS CON FORMATO DD/MM/AAAA

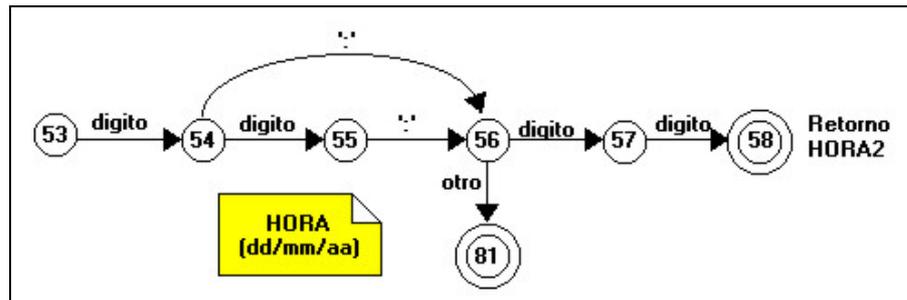


FIGURA A.10 PARA EL RECONOCIMIENTO DE HORAS FORMATO DD/MM/AA

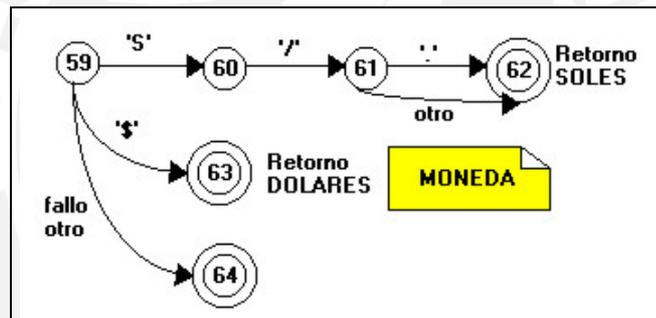


FIGURA A.11 PARA EL RECONOCIMIENTO DE MONEDAS (SOLES Y DÓLARES)

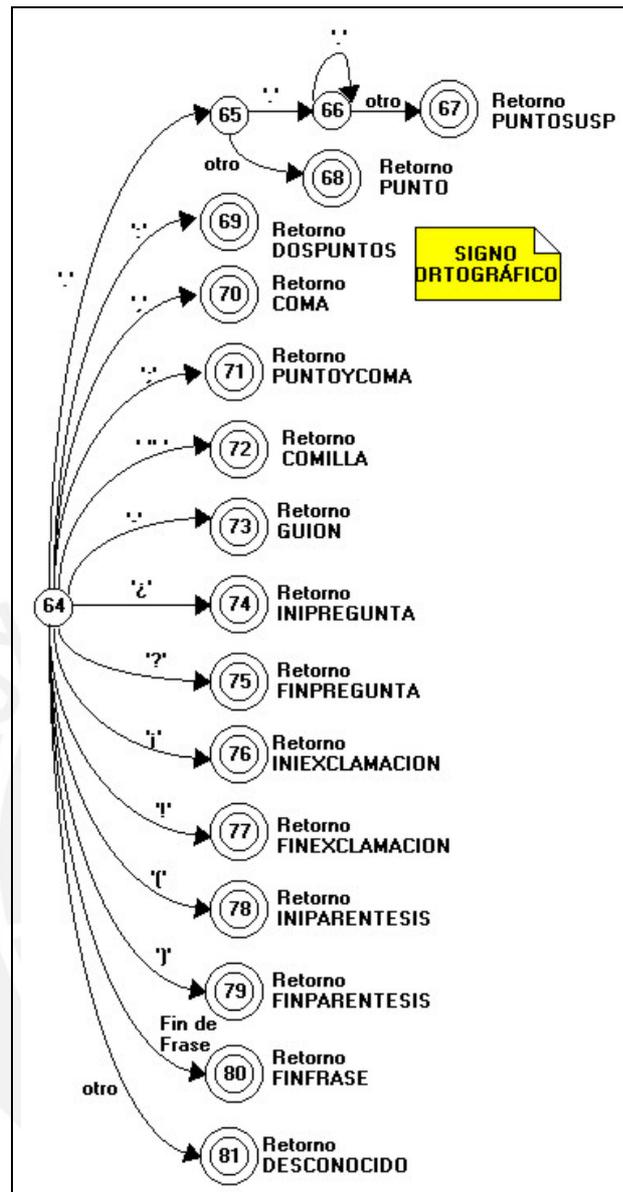


FIGURA A.12 PARA EL RECONOCIMIENTO DE SIGNOS ORTOGRÁFICOS

Anexo B

Algoritmo de Reglas Para la Separación de las Sílabas de una Palabra

De la bibliografía se concluye que todas las posibles combinaciones de sílabas en el idioma Castellano, se pueden separar en tres grupos principales de acuerdo a los grafemas que forman el inicio de sílaba:

1. Inicio de sílaba: Vocal.
2. Inicio de sílaba: Consonante + Vocal.
3. Inicio de sílaba: Consonante + Consonante.

Mediante el siguiente pseudocódigo se describe el algoritmo utilizado para la separación de las sílabas de una palabra:

Función Silabear(**Entrada:** Palabra; **Salida:** PalabraSilabas,
NumeroSilabas)

Inicio

NumeroSilabas = 0

PalabraSilabas = Vacío

Mientras (no sea final de palabra) Hacer

Inicio

L1 es la primera letra de Palabra

L2 es la segunda letra de Palabra

Si (L1 es Vocal) Entonces

Sílaba = SilabeaV - Caso 1 – Inicio de sílaba Vocal

Caso Contrario Si (L2 es Vocal ó L2 es 'y') Entonces

Sílaba = SilabeaCV - Caso 2 – Inicio de sílaba Consonante + Vocal

Caso Contrario

Sílaba = SilabeaCC - Caso 3 – Inicio de sílaba Consonante +
Consonante

Si (Longitud de Sílaba > 0) Entonces

Inicio

Agregar Sílaba a PalabraSilabas

Eliminar letras de Sílabas encontradas de Palabra

NumeroSilabas = NumeroSilabas + 1

Fin Si

Caso Contrario

Eliminar primera letra Palabra

Fin Mientras

Fin Función

En la tabla B.1 se describe la notación utilizada en esta sección.

TABLA B.1 NOTACIÓN UTILIZADA

Símbolo	Descripción
⊠	Fin de palabra, o carácter diferente a vocal o consonante (',', ':', '1', etc.)
\$	Cualquier carácter
[CV]	Las siguientes dos letras son CV
[CCV]	Las siguientes dos letras pertenecen al grupo de consonantes inseparables
V*	Vocal acentuada
	Utilizada para establecer posibilidades alternativas
Li	l-ésima letra de la sílaba o palabra

B-1 Algoritmo para el caso de inicio de sílaba: Vocal

Según la estructura de las sílabas presentadas en [Mateo 2000], las sílabas que empiezan con vocal presentan los casos de terminación mostrados en la tabla B.1.1, la tabla también muestra la sílaba resultante y la referencia a la función que comprueba la validez de la regla.

TABLA B.1.1 POSIBLES CASOS DE TERMINACIÓN DE UNA SÍLABA QUE INICIA CON UNA VOCAL

Referencia	Sílaba	Posibles casos de terminación de la sílaba
V01	V	⊠ [CV] [CCV] V*
V02	VC	⊠ [CV] [CCV]
V03	VV	⊠ [CV] [CCV]
V04	VVC	⊠ [CV] [CCV]
V05	VhV	⊠ [CV] [CCV]
V06	VCC	⊠ [CV] [CCV]

B-2 Algoritmo para el caso de inicio de sílaba: Consonante + Vocal

En la tabla B.2.1, se describen los posibles casos de terminación de sílaba, del caso de inicio de sílaba Consonante + Vocal.

TABLA B.2.1 POSIBLES CASOS DE TERMINACIÓN DE UNA SÍLABA QUE INICIA CON CONSONANTE + VOCAL

Referencia	Sílaba	Posibles casos de terminación de la sílaba
CV01	CV	∅ [CV] [CCV]
CV02	CVC	∅ [CV] [CCV]
CV03	CVCC	∅ [CV]
CV04	CVV	∅ [CV] [CCV]
CV05	CVVC	∅ [CV] [CCV]
CV06	CVVV	∅ [CV] [CCV]
CV07	CVVVC	∅ [CV] [CCV]
CV08	quV guV	∅ [CV] [CCV] V
CV09	quVC guVC	∅ [CV] [CCV]
CV10	QuVVC guVVC	∅ [CV] [CCV]

B-3 Algoritmo para el caso de inicio de sílaba: Consonante + Consonante

Para el tercer caso se describen las reglas en la tabla B.3.1.

TABLA B.3.1 POSIBLES CASOS DE TERMINACIÓN DE UNA SÍLABA QUE INICIA CON CONSONANTE + CONSONANTE

Referencia	Sílaba	Posibles casos de terminación de la sílaba
CC01	CCV	∅ [CV] [CCV]
CC02	CCVC	∅ [CV] [CCV]
CC03	CCVCC	∅ [CV]
CC04	CCVV	∅ [CV] [CCV]
CC05	CCVVC	∅ [CV] [CCV]
CC06	CCVhV	\$
CC07	CCVVVC	∅ [CV] [CCV]

Anexo C

Listado del Diccionario Léxico y la Tabla de Terminaciones

C-1 Palabras Átonas

TABLA C.1. PALABRAS CONSIDERADAS COMO EXCEPCIONES GRAMATICALES

Categoría	Palabra	Categoría	Palabra
ADJETIVO	don	PREPOSICION	ante
	doña		cabe
	frey		contra
	san		desde
	sor		entre
ADVESPECIFICADOR	muy		hacia
	tan		hasta
ARTICULO	al		mediante
	del		para
	el		por
ARTINDETERMINADO	un	so	
	una	tras	
	unas	a	
	unos	con	
ARTPREV	la	PREPVACIA	de
	las		en
	lo		sin
CONJUNCIONCOORD	los	PRONPOSESIVO	mi
	e		su
	mas		sus
	ni	PRONPREVERBAL	le
	o		les
	pero		me
	sino		nos
	u		os
y	se		
CONJUNCIONSUB	ahora	PRONRELATIVO	te
	aunque		cuando
	como		cuanto
	porque		cuyo
	pues		donde
si	que		
NOMBREADJETIVO	fray		quien
PREPOSICION	allende		tu

C-2 Palabras Ambiguas en su Acentuación

TABLA C.2.1 PALABRAS AMBIGUAS CONSIDERADAS EN EL SISTEMA

Palabra
bajo
cual
sobre
solo

C-3 Excepciones Gramaticales

TABLA C.3.1 LISTADO DE EXCEPCIONES GRAMATICALES

Categoría	Palabra	Categoría	Palabra
ADJETIVO	algún	PRONDETERMINATE	otro
	anterior	PRONINTERROGATIVO	cómo
	bueno		cuál
	cualquier		cuán
	exterior		cuándo
	extremo		cuánto
	grande		dónde
	inferior		qué
	ínfimo		quién
	inteligente		quieres
	interior		PRONPERSONAL
	íntimo	consigo	
	máximo	contigo	
	medio	él	
	mínimo	ella	
	muchos	ellas	
	ningún	ello	
	óptimo	ellos	
	oscuro	mí	
	pésimo	nosotras	
	pocos	nosotros	
	posterior	poco	
	postremo	sí	
	puro	ti	
	sumo	tú	
	superior	usted	
	supremo	ustedes	
	todos	vos	
	último	vosotras	
	un	vosotros	
	una	yo	
	unas	PRONPOSESIVO	mía
uno	mías		

	unos		mío
	varios		míos
	vehemente		nuestra
ADVERBIO	abajo		nuestras
	acá		nuestro
	acaso		nuestros
	adelante		suya
	adentro		suyas
	adrede		suyo
	afuera		suyos
	ahí		tuya
	ahora		tuyas
	alrededor		tuyo
	allá		tuyos
	allí		vuestra
	anoche		vuestras
	antaño		vuestro
	anteanoche		vuestros
	anteayer	PRONRELATIVO	cuales
	antes	VERBO	animo
	aprisa		animó
	aquí		busco
	arriba		comí
así	compró		
aún	dé		
ayer	decían		
bien	dicho		
cerca	esperen		
ciertamente	hablemos		
cierto	miento		
claro	termino		
cuan	terminó		
cuanto	viniste		
delante	VERBOESTAR		está
dentro		estaba	
deprisa		estabais	
despacio		estábamos	
después		estaban	
encima		estabas	
enseguida		estad	
entonces		estado	
fuera		estáis	
harto		estamos	
hoy		están	
jamás		estando	
lejos		estar	
lento		estará	

	luego		estarán
	mal		estarás
	mañana		estaré
	medio		estaréis
	mejor		estaremos
	mientras		estaría
	mucho		estaríais
	nada		estaríamos
	no		estarían
	nunca		estarías
	otrora		estás
	peor		esté
	poco		estéis
	presto		estemos
	pronto		estén
	puro		estés
	quedo		estoy
	quizá		estuve
	quizás		estuviera
	recién		estuvierais
	regular		estuviéramos
	seguramente		estuvieran
	sí		estuvieras
	siempre		estuviere
	sobre		estuviereis
	solamente		estuviéremos
	sólo		estuvieren
	también		estuvieres
	tampoco		estuvieron
	tanto		estuviese
	tarde		estuvieseis
	temprano		estuviésemos
	todavía		estuviesen
	todo		estuvieses
	ya		estuvimos
	apenas		estuviste
	bastante		estuvisteis
	casi		estuvo
ADVESPECIFICADOR	demasiado	VERBOHABER	ha
	más		habed
	menos		habéis
	tan		haber
CONJUNCIONSUB	apenas		había
	conque		habíais
	luego		habíamos
	mientras		habían
INTERJECCION	ah		habido

	ay		habiendo
	bah		habrá
	caramba		habrán
	eh		habrás
	zas		habré
NOMBREADJETIVO	bajo		habréis
	demás		habremos
NOMCOMUN	algo		habría
	alguien		habríaís
	alguno		habríamos
	algunos		habrían
	ánimo		habrías
	aquél		han
	aquella		has
	aquellas		hay
	aquello		haya
	aquéllos		haya
	baba		hayáis
	babas		hayamos
	cable		hayan
	cables		hayas
	café		he
	competición		hemos
	cual		hube
	cualquiera		hubiera
	demente		hubierais
	ésa		hubiéramos
	ésas		hubieran
	ése		hubieras
	eso		hubiere
	esos		hubiereis
	especimen		hubiéremos
	especímenes		hubieren
	ésta		hubieres
	estado		hubieron
	ésta		hubiese
	éste		hubieseis
	esto		hubiésemos
	éstos		hubiesen
	estudio		hubieses
	germen		hubimos
	gérmenes		hubiste
	leño		hubisteis
	libro		hubo
	manos	VERBOSER	era
	mayoría		erais
	mesa		éramos

	estas		somos
	este		son
	estos		soy
	ninguna		

C-4 Terminaciones y Desinencias

TABLA C.4.1 TABLA DE TERMINACIONES Y DESINENCIAS

Terminación	Categoría	Terminación	Categoría
a	VERBO	idad	NOMCOMUN
a	NOMBREADJETIVO	ido	VERBO
á	VERBO	iendo	VERBO
á	NOMBREADJETIVO	iera	VERBO
aba	VERBO	ierais	VERBO
abais	VERBO	iéramos	VERBO
ábamos	VERBO	ieran	VERBO
aban	VERBO	ieras	VERBO
abas	VERBO	iere	VERBO
ad	VERBO	iereis	VERBO
ado	VERBO	iéremos	VERBO
ado	NOMBREADJETIVO	ieren	VERBO
ais	VERBO	ieres	VERBO
áis	VERBO	ieron	VERBO
amos	VERBO	iese	VERBO
an	NOMCOMUN	ieseis	VERBO
an	VERBO	iésemos	VERBO
án	NOMCOMUN	iesen	VERBO
án	VERBO	ieses	VERBO
ando	VERBO	imos	ADJETIVO
ar	VERBO	imos	VERBO
ar	NOMBREADJETIVO	ió	VERBO
ara	VERBO	ión	NOMCOMUN
ará	VERBO	ir	VERBO
arais	VERBO	irá	VERBO
áramos	VERBO	irán	VERBO
aran	VERBO	irás	VERBO
arán	VERBO	iré	VERBO
aras	VERBO	iréis	VERBO
arás	VERBO	iremos	VERBO
are	VERBO	iría	VERBO
aré	VERBO	iríais	VERBO
areis	VERBO	iríamos	VERBO
aréis	VERBO	irían	VERBO
aremos	VERBO	irías	VERBO
áremos	VERBO	is	NOMBREADJETIVO
aren	VERBO	ís	VERBO
ares	VERBO	ísimo	ADJETIVO

aría	VERBO	ísimos	ADJETIVO
aríaís	VERBO	ismo	NOMCOMUN
aríamos	VERBO	iste	VERBO
arían	VERBO	isteis	VERBO
arías	VERBO	ita	NOMBREADJETIVO
aron	VERBO	itas	ADJETIVO
as	VERBO	ito	NOMBREADJETIVO
as	NOMBREADJETIVO	itos	NOMCOMUN
ás	VERBO	la	VERBO
ase	VERBO	la	NOMBREADJETIVO
aseís	VERBO	las	VERBO
ásemos	VERBO	las	NOMBREADJETIVO
asen	VERBO	les	VERBO
ases	VERBO	les	NOMBREADJETIVO
aste	VERBO	lo	VERBO
asteís	VERBO	lo	NOMBREADJETIVO
aya	VERBO	los	VERBO
ayáis	VERBO	los	NOMBREADJETIVO
ayamos	VERBO	me	VERBO
ayan	VERBO	mela	VERBO
ayas	VERBO	melas	VERBO
az	VERBO	mele	VERBO
aza	NOMBREADJETIVO	meles	VERBO
azas	NOMBREADJETIVO	melo	VERBO
azo	NOMBREADJETIVO	melos	VERBO
azos	NOMBREADJETIVO	mente	ADVERBIO
ble	NOMBREADJETIVO	nos	VERBO
bles	NOMBREADJETIVO	nos	NOMBREADJETIVO
e	VERBO	nosla	VERBO
e	NOMBREADJETIVO	noslas	VERBO
é	VERBO	nosle	VERBO
é	NOMBREADJETIVO	nosles	VERBO
ed	VERBO	noslo	VERBO
eís	VERBO	noslos	VERBO
éis	VERBO	ño	NOMBREADJETIVO
emos	VERBO	ños	NOMBREADJETIVO
en	NOMCOMUN	o	VERBO
en	VERBO	o	NOMBREADJETIVO
én	NOMCOMUN	ó	VERBO
én	VERBO	ó	NOMBREADJETIVO
er	VERBO	ois	VERBO
er	NOMBREADJETIVO	óloga	NOMBREADJETIVO
era	VERBO	ólogas	NOMBREADJETIVO
era	NOMBREADJETIVO	ólogo	NOMBREADJETIVO
erá	VERBO	ólogos	NOMBREADJETIVO
erais	VERBO	oma	NOMBREADJETIVO
éramos	VERBO	omas	ADJETIVO

eran	VERBO	omos	VERBO
erán	VERBO	on	VERBO
eras	VERBO	os	VERBO
erás	VERBO	os	NOMBREADJETIVO
ere	VERBO	osa	NOMBREADJETIVO
eré	VERBO	oy	VERBO
ereis	VERBO	rio	NOMBREADJETIVO
eréis	VERBO	rió	NOMBREADJETIVO
eremos	VERBO	se	VERBO
éremos	VERBO	sela	VERBO
eren	VERBO	selas	VERBO
eres	NOMCOMUN	sele	VERBO
eres	VERBO	seles	VERBO
ería	VERBO	seles	VERBO
eríais	VERBO	selo	VERBO
eríamos	VERBO	selos	VERBO
erían	VERBO	seme	VERBO
erías	VERBO	semela	VERBO
erías	NOMBREADJETIVO	semelas	VERBO
ero	ADJETIVO	semele	VERBO
eron	VERBO	semeles	VERBO
érrimo	ADJETIVO	semelo	VERBO
érrimos	ADJETIVO	semelos	VERBO
es	VERBO	senos	VERBO
es	NOMBREADJETIVO	seoa	VERBO
és	VERBO	sete	VERBO
ese	VERBO	setela	VERBO
eseis	VERBO	setelas	VERBO
ésemos	VERBO	setele	VERBO
esen	VERBO	seteles	VERBO
eses	VERBO	setelo	VERBO
i	VERBO	setelos	VERBO
i	NOMBREADJETIVO	te	VERBO
í	VERBO	tela	VERBO
í	NOMBREADJETIVO	telas	VERBO
ía	VERBO	tele	VERBO
íais	VERBO	teles	VERBO
íamos	VERBO	telo	VERBO
ían	VERBO	telos	VERBO
ías	VERBO	z	NOMBREADJETIVO
ld	VERBO		

Anexo D

Reglas de Transcripción Grafema-Alófono y Fonetización Sintáctica

Este anexo consta de dos partes: en la primera se presentan las reglas usadas en la transcripción grafema alófono, y en la segunda, se listan las reglas utilizadas en la fonetización sintáctica la cual permite corregir la transcripción hecha con las reglas anteriores, modelando los fenómenos de modificación alofónica que se produce al leer un texto en forma natural.

D-1 Reglas de Transcripción Grafema-Alófono

TABLA D.1.1 REGLAS DE TRANSCRIPCIÓN GRAFEMA-ALÓFONO

Nro.	Foco	Contexto Izquierdo	Contexto Derecho	cambio
1	p		- cs	b3
2	p			p1
3	b	0mn	aeiouáéíóúÁÉÍÓÚ	b1
4	b	#- mn	aeiouáéíóúÁÉÍÓÚ	b1
5	v	0mn	aeiouáéíóúÁÉÍÓÚ	b1
6	v	#- mn	aeiouáéíóúÁÉÍÓÚ	b1
7	t			t1
8	d	0nl	aeiouáéíóúÁÉÍÓÚ	d1
9	d	#- nl	aeiouáéíóúÁÉÍÓÚ	d1
10	qu		eiéíÉÍ	k1
11	c		aouáóúÁÓÚ	k1
12	k		- bvdymñllrrl	g1
13	k			k1
14	gu	0n	eiéíÉÍ	g1
15	gu	#- n	eiéíÉÍ	g1
16	g	0n	aouáóúÁÓÚü	g1
17	g	#- n	aouáóúÁÓÚü	g1
18	b		aeiouáéíóúÁÉÍÓÚ	b2
19	v		aeiouáéíóúÁÉÍÓÚ	b2
20	f			f1
21	c		eiéíÉÍ	s1
22	z			s1
23	d		aeiouáéíóúÁÉÍÓÚ	d2
24	s			s1
25	hi	0#-aeiouáéíóúÁÉÍÓÚ	aeou	y1
26	hi	0#-aeiou	áéóúÁÉÍÓÚ	y1
27	y		aeiouáéíóúÁÉÍÓÚ	y1
28	g		eiéíÉÍ	x1
29	j		#0	g3
30	j			x1
31	gu		eiéíÉÍ	g2
32	g		aouáóúÁÓÚü	g2
33	ch			c1
34	m		# aeiouáéíóúÁÉÍÓÚ	n1

35	m		0	n1
36	m		aeiouáéíóúÁÉÍÓÚ	m1
37	n		#- pbv	n2
38	n		#- vm	n2
39	n		#- f	n2
40	n		- f	n2
41	n		# f	n2
42	n		aeiouáéíóúÁÉÍÓÚ	n1
43	m			n2
44	n			n2
45	ñ			n3
46	ll		-#0	l1
47	ll			l2
48	l			l1
49	rr			r2
50	r	- !0#nls		r1
51	r		0	r1
52	r	!0#-		r1
53	r		!r	r2
54	g			g3
55	xc		eiéíÉí	c2s1
56	x			c2s1
57	-x			c2-s1
58	c-c			g3-s1
59	c-n			g3-n1
60	c			c2
61	b			b3
62	d			d3
63	y			i1
64	w			u1

Leyenda:

- 0: Inicio de sílaba después de pausa o inicio de frase.
- : Separador de sílabas
- #: Separador de palabras
- !: Niega la condición
- |: Agrupa grafemas compuestos en las condiciones de contexto.

D-2 Reglas Para la Fonetización Sintáctica

TABLA D.2.1 REGLAS PARA LA FONETIZACIÓN SINTÁCTICA

Nro	Foco	Contexto Izquierdo	Contexto Derecho	cam bio
1	b3#		a3a4e3e4i3i4o3o4u3u4	#b2
2	d3#		a3a4e3e4i3i4o3o4u3u4	#d2
3	g3#		e3e4i3i4	#g1
4	g3#		a3a4o3o4u3u4	#g2
5	p1#		a3a4e3e4i3i4o3o4u3u4	#p1
6	T1#		a3a4e3e4i3i4o3o4u3u4	#t1
7	c2#		a3a4e3e4i3i4o3o4u3u4	#c2
8	k1#		a3a4e3e4i3i4o3o4u3u4	#k1
9	c1#		a3a4e3e4i3i4o3o4u3u4	#c1
10	f1#		a3a4e3e4i3i4o3o4u3u4	#f1
11	z1#		a3a4e3e4i3i4o3o4u3u4	#z1
12	s1#		a3a4e3e4i3i4o3o4u3u4	#s1
13	x1#		a3a4e3e4i3i4o3o4u3u4	#x1
14	l1#		a3a4e3e4i3i4o3o4u3u4	#l1
15	r1#		a3a4e3e4i3i4o3o4u3u4	#r1
16	p1#		d1g1y1m1n1n3l1l2r2	b3#
17	t1#		b1g1y1m1n1n3l1l2r2	d3#
18	k1#		b1d1y1m1n1n3l1l2r2	g3#
19	p1#		p1b1b2	#
20	t1#		t1d1d2b1b2	#
21	k1#		k1g1g2	#
22	c2#		k1g1g2	#
23	f1#		f1	#
24	z1#		z1	#
25	s1#		s1	#
26	x1#		x1	#
27	r1#		r2	#
28	g3#		p1t1k1c1f1z1s1x1	c2#
29	#i3	a2e2o2 !a1a2e1e2o1o2u1u2	!#	#i1
30	#i3	a1e1o1 !a1a2e1e2o1o2u1u2i1i2	!#	#i1
31	#u3	a2e2o2 !a1a2e1e2o1o2u1u2	!#	#u1
32	#u3	a1e1o1 !a1a2e1e2o1o2u1u2i1i2	!#	#u1
33	#i3	a2e2o2 !a1a2e1e2o1o2u1u2	# p1t1k1b1d1g1y1c1f1z1s1x1m1n1n3l1l2r2	#i1

Anexo E

Duración Consonántica

TABLA E.1 DURACIÓN CONSONÁNTICA SEGÚN EL ACENTO, LA POSICIÓN EN LA SÍLABA Y LA POSICIÓN EN EL GRUPO FÓNICO

Acento		+				-				Promedio
Inicio de sílaba		+		-		+		-		
Prepausal		+	-	+	-	+	-	+	-	
r1	ms	27	29	81	52	33	27	68	48	46
r2	ms	103	86	*	*	111	84	*	*	96
m1	ms	80	68	*	76	79	66	*	78	74
n1	ms	66	62	148	73	58	59	120	71	78
n3	ms	102	71	*	*	105	89	*	*	90
l1	ms	53	44	103	52	81	42	84	47	59
l2	ms	78	62	*	*	70	61	*	*	67
f1	ms	125	101	*	*	139	93	*	*	115
z1	ms	105	106	131	91	120	86	*	*	104
s1	ms	83	96	133	63	108	103	118	69	99
z	ms	*	*	*	63	*	*	*	63	63
y1	ms	59	57	*	*	78	55	*	*	60
x1	ms	128	100	145	94	125	90	*	*	111
c1	ms	132	107	*	*	155	95	*	*	117

Anexo F

Estructura de Archivos de Datos y de la Configuración del Sistema

TABLA F.1. ESTRUCTURA Y DEFINICIÓN DE LOS ARCHIVOS DE DATOS DEL SISTEMA

Nombre	Descripción	Estructura
Abrev.dat	Es un archivo binario que almacena los pares de abreviaturas y su equivalente de palabra extendida.	sRegAbrev: char Abrev[10]; char Expand[100];
Alofono.dat	Almacena las reglas de conversión grafema alófono, usada en el proceso de transcripción grafema a unidades básicas. Tiene cuatro campos: grafema, condición de contexto anterior, condición de contexto posterior, y el símbolo de alófono correspondiente al grafema.	sAlofono: char Graf[7]; char Ante[60]; char Post[60]; char Simb[7];
Ambiguo.dat	Almacena las palabras ambiguas en su acentuación. Tiene un solo campo, la palabra misma.	sRegAmbiguo: char Ambiguo[10];
ArcTerm.dat	Almacena desinencias y sufijos que por su forma podrían pertenecer a cierta categoría gramatical. Tiene dos campos: la terminación de palabra y el código de la categoría gramatical a la que pertenece.	sRegPalGram: char PalGram[15]; INT16 CatGram;
Atona.dat	Contiene una lista de palabras consideradas átonas por el sistema. Tiene dos campos: La palabra misma, y el código de la categoría gramatical a la que pertenece.	sRegPalGram: char PalGram[15]; INT16 CatGram;
CatGram.dat	Almacena todas las categorías gramaticales que usa el sistema. Tiene tres campos: código, nombre de la categoría gramatical y su descripción.	sCatGram: UINT8 Cod; char Nombre[20]; char Desc[60];
DurCons.dat	Contiene las reglas para la modificación de la duración de las consonantes. Tiene cinco campos: El alófono de la consonante, indicador de si está en sílaba acentuada, indicador si está al inicio de una sílaba, indicador si está en posición prepausal y la duración que le corresponde si cumple las tres condiciones anteriores.	sRegReglaCons: char Alof[3]; UINT8 Acento; UINT8 IniSil; UINT8 PrePausa; UINT16 Duracion;
FonSintac.dat	Contiene las reglas de contexto que se usan en la fonetización sintáctica. Tiene cuatro campos: símbolo(s) candidato(s) a modificarse, condición de contexto anterior, condición de contexto posterior, y el nuevo símbolo(s) a colocarse en reemplazo del candidato.	sAlofono: char Graf[7]; char Ante[60]; char Post[60]; char Simb[7];

PalGram.dat	Contiene una lista de palabras que se consideran como excepciones gramaticales. Tiene dos campos, la palabra y la categoría gramatical a la cual pertenece.	sRegPalGram: char PalGram[15]; INT16 CatGram;
Pausa.dat	Guarda la información de las pausas consideradas en el sistema. Tiene tres campos: Código de pausa, descripción y la duración expresada en milisegundos.	sRegPausa: UINT8 Cod; char Desc[45]; UINT16 Duracion;
ReglaGra.dat	Almacena las reglas gramaticales de contexto utilizadas en la desambiguación gramatical. Tiene tres campos: condiciones del contexto izquierdo, condiciones del contexto derecho, y la clave, que representa a la categoría gramatical que se está intentando desambiguar.	sReglaGra: char ContextIzq[12]; char ContextDer[12]; UINT8 clave;
Sigla.dat	Almacena las siglas y su equivalente expandida como se desea que se pronuncie. Tiene la misma estructura que las abreviaturas.	sRegAbrev: char Abrev[10]; char Expand[100];

TABLA F.2. ESTRUCTURA DE LA CONFIGURACIÓN DEL SISTEMA

HKEY_LOCAL_MACHINE			
SOFTWARE			
CTV_Lector			
	Arc_Lpc	"fonemas.lpc"	Archivo LPC.
	Path_Datos	"D:\CTV_Lector\Datos"	Directorio de datos.
	Path_Exe	"D:\CTV_Lector"	Directorio del programa.
	Path_Salida	"D:\CTV_Lector\Salida"	Directorio de resultados.
	DISTANCIA		
	DistDemi	1	Dist. entre demisílabas.
	DistDemiOmi	1	Dist. entre demisílabas por omisión.
	DistPalabra	10	Dist. entre palabras.
	DistPalOmi	10	Dist. entre palabras por omisión.
	DistSilaba	1	Distancia entre sílabas.
	DistSilOmi	1	Dist. entre sílabas por omisión.
	GENERALES		
	Velocidad	16000	Velocidad de lectura.
	VelOmi	16000	Vel. De lectura por omisión.
	VolOmi	1	Vol. Por omisión.
	Volumen	1	Volumen de lectura por omisión.
	INTERPOLACIÓN		

	InterDemiOmi	20	Tam. interpolación de demisílabas por omisión
	InterDemiSil	20	Tam. interpolación de demisílabas.
	InterPalabra	1	Tam. de interpolación de palabras.
	InterPalOmi	1	Tam. de interpolación de palabras por omisión.
	InterSilaba	20	Tam. interpolación de sílabas.
	InterSilOmi	20	Tam. interpolación de sílabas por omisión.
	PROSODIA		
	ModDur	1	Indicador de modificación de la duración de las unidades básicas.
	ModDurOmi	1	Indicador de modificación por omisión de la duración de las unidades básicas.
	ModF0	1	Indicador de modificación del contorno de frecuencia fundamental de la frase.
	ModF0Omi	0	Indicador de modificación por omisión del contorno de frecuencia fundamental de la frase.

Anexo G

Estándares de Programación

G-1 Estándares de Pantallas

1. **Estándares para nombre de archivos:** Para nombrar a los archivos fuente, se seguirá el siguiente formato: **Descripción + Mnemónico**

TABLA G.1.1.1 ESTÁNDARES PARA NOMBRE DE ARCHIVOS

Archivo	Mnemónico	Ejemplo
Form (Formulario)	Frm	TextoFrm
Múltiple Document Interfase	Mdi	EditorWinMdi
Unit Forms (Unidades de formulario)	Frm	TextoFrm
Unit (Unidades de propósito general)	Unt	TiposUnt

2. **Estándares para controles:** Se definieron los siguientes estándares de controles:

- El nombre de los objetos será constituido por un prefijo en minúsculas y un mnemotécnico que trate de explicar el uso del objeto cuya primera letra estará en mayúscula: Por ejemplo, el botón Aceptar se llamará *btnAceptar*. El tipo de letra a utilizar en todos los controles es MS Sans Serif tamaño 8. En la Tabla 3.1.3.1.2 se muestran los objetos Delphi usados con su respectivo prefijo:

TABLA G.1.2.1 ESTÁNDARES PARA CONTROLES

Control	Prefijo	Control	Prefijo
ActionList	Al	ImageList	il
Button	btn	RichEdit	re
Panel	pnl	SpeedButton	sb
Label	Lbl	MainMenu	mm
Edit	edt	MainMenuItem	mmi
ListBox	Lb	StringGrid	sg
CheckBox	Cb	TrackBar	tbr
RadioButton	Rb	ToolBar	tib
ScrollBar	scb	ToolBarButton	tbn

- Los botones de la barra de herramientas tienen de alto 22 y de ancho 23 Unidades.
3. **Plantillas:** Con la finalidad agilizar el desarrollo y aprovechando la herencia de la programación orientada a objetos, se definió el diseño y la funcionalidad de algunos botones y pantallas a nivel general, para que

sean usadas vía la herencia para generar otras pantallas con los mínimos cambios posibles.

- Plantilla de Botones:

TABLA G.1.3.1 PLANTILLA DE BOTONES

Nombre	Icono	Funcionalidad
Nuevo		Crea un documento nuevo
Abrir		Abre un documento de texto.
Guardar		Guarda los cambios hechos en un documento.
Salir		Cierra la ventana activa.
Cortar		Corta el texto seleccionado.
Copiar		Copia el texto seleccionado.
Pegar		Pega el texto previamente seleccionado.
Eliminar		Elimina el texto seleccionado.
Buscar		Busca una palabra o frase en el texto.
Consultar		Coloca los datos visualizados en modo "consulta".
Editar		Coloca los datos visualizados en modo "Edición"
Imprimir		Manda imprimir los reportes y gráficos del sistema.

- Plantilla de la pantalla de Mantenimiento de datos: La siguiente pantalla tiene la funcionalidad general de todas las pantallas de consulta y mantenimiento usadas en el sistema y mediante la herencia pasa estas funcionalidades a las ventanas derivadas de esta.

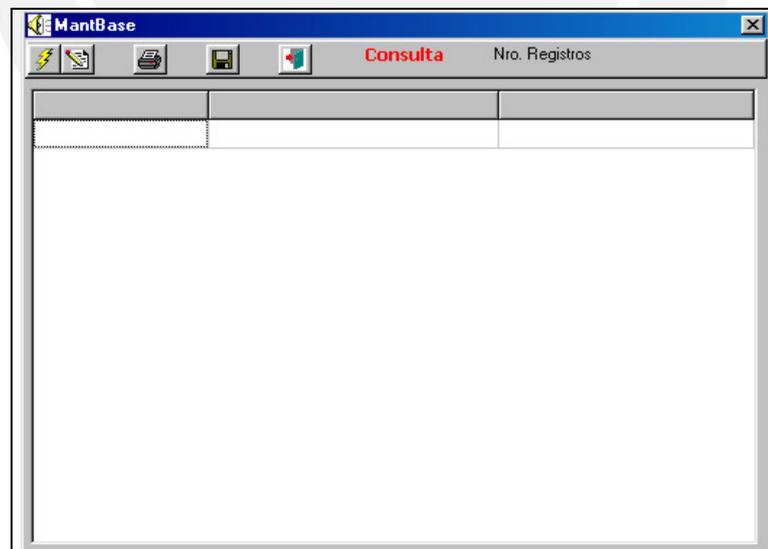


FIGURA G.1.3.1 PLANTILLA DE MANTENIMIENTO DE DATOS

G-2 Estándares de Reportes

Se usará el tipo de letra "Courier New", con las siguientes variantes:

Característica	Letra	Alineación
Título del reporte	Negrita, tamaño 14	Centrado
Fecha y nro. de página	Normal, tamaño 8	Superior derecha
Título de datos	Negrita, tamaño 10	
Detalle de datos	Normal, tamaño 9	

G-3 Estándares de Programación

Si no se especifica los estándares definidos son válidos para ambos entornos de desarrollo, Delphi y Visual C++.

1. **Variables:** La nomenclatura de las variables será la siguiente: **Alcance + TipoDato + NombreDeVariable**. Donde Alcance puede ser:

Alcance	Prefijo
Global	g
Local al procedimiento o función	l
Parámetro	p

2. **Estándares de clases:**

- Las clases se nombrarán de la siguiente manera:
 1. En Visual C++: c + NombreClase. Ejemplo: cTexto.
 2. En Delphi: T + NombreClase. Ejemplo: TTexto.
- Para nombrar a los atributos se usarán nombres descriptivos. El nombre se divide en dos grupos separados por un guión. Las primeras letras de cada grupo se escriben en mayúsculas. Ejemplo: Est_Ado.
- Para nombrar a las operaciones, se usarán nombres descriptivos, en este caso no se separan por un guión. Ejemplo: InsertarPausa.
- Para nombrar a las variables objeto de clases, se omitirá la primera letra del nombre de la clase. Ejemplo: Texto.

Anexo H

Diseño y Funcionalidad de Menús, Programas y Reportes

H-1 Diseño de Menús

El siguiente cuadro muestra la estructura del menú del sistema conversor de texto-voz(CTV):

Descripción	Icono	Tecla de Acceso
<u>A</u> rchivo		
<u>N</u> uevo		Ctrl+N
<u>A</u> brir		Ctrl+A
<u>C</u> errar		

<u>G</u> uardar		Ctrl+G
<u>G</u> uardar como...		

<u>S</u> alir		
<u>E</u> dición		
<u>C</u> ortar		Ctrl+X
<u>C</u> opiar		Ctrl+C
<u>P</u> egar		Ctrl+V
<u>B</u> orrar		

<u>S</u> eleccionar todo		Ctrl+E
<u>F</u> echa y <u>H</u> ora		F6

<u>B</u> uscar		Ctrl+B
<u>B</u> uscar siguiente		F3
<u>L</u> ectura		
<u>L</u> eer		Ctrl+L
<u>G</u> enerar Wav		Ctrl+W
<u>L</u> eer y <u>G</u> enerar Wav		

<u>C</u> ontinuar		Ctrl+S
<u>P</u> ausar		Ctrl+P

<u>D</u> etener		Ctrl+D

<u>R</u> epetir		Ctrl+T
<u>R</u> epetir <u>F</u> rase		Ctrl+R
<u>V</u> er		
<u>D</u> etalle de Frase		Ctrl+F

<u>C</u> ategorización de Frase		
<u>D</u> uración		
<u>F</u> rec. Fundamental		

Herramientas		
<u>C</u> onfiguración		F2

+ <u>V</u> olumen		F4
- <u>V</u> olumen		F5

+ <u>V</u> elocidad		F7
- <u>V</u> elocidad		F8

<u>T</u> ablas		
<u>A</u> breviaturas		
<u>S</u> iglas		
<u>P</u> ausas		

<u>C</u> at. Gramaticales		
<u>E</u> xcep. Gramaticales		
<u>T</u> erminaciones Gram.		
<u>A</u> tonas		
<u>A</u> mbigua		
Ventana		
<u>M</u> osaico		
<u>C</u> ascada		
<u>O</u> rganizar Iconos		

Lista de Documentos Abiertos		

H-2 Diseño de Programas

El sistema conversor texto-voz(CTV) está compuesto de dos módulos principales: el módulo “cliente” que viene a ser la parte ejecutable del sistema, y una librería que contiene las clases que permiten hacer el análisis lingüístico-prosódico y la generación de voz. En este apartado describiremos los programas de los que está compuesto el módulo cliente (ejecutable) del sistema. En la siguiente tabla se muestran los programas de este módulo:

Programa	Descripción
Lector.exe	Programa principal del sistema. Basado en ventanas (entorno Windows), la tarea principal que realiza es leer un texto cargado en el editor del sistema. Permite seguir la secuencia de la lectura, y además, permite manipular la lectura, brindando opciones para detenerla, repetir la última frase, volver a leer todo, editar el texto a leer, crear un nuevo documento, etc. Tiene subprocesos que permiten hacer la consulta y mantenimiento de la configuración y base de datos del sistema. Finalmente, se cuenta con consultas, gráficos y reportes que permiten visualizar los resultados del procesamiento sobre la última frase. Desarrollado con Borland Delphi 5.0.

LectorLC.exe	Programa que se ejecuta desde la línea de comandos de Windows, no tiene ventanas y la única función que cumple es leer el texto que se le da como parámetro. Al igual que el programa "Lector.exe", permite indicar la opción de lectura, que puede ser, leer al instante, generar un archivo de sonido con el resultado de la lectura, y leer y generar archivo de sonido. Desarrollado con Visual C++ 6.0.
--------------	--

1. **Lector.exe:**

➤ **Jerarquía de Ventanas**

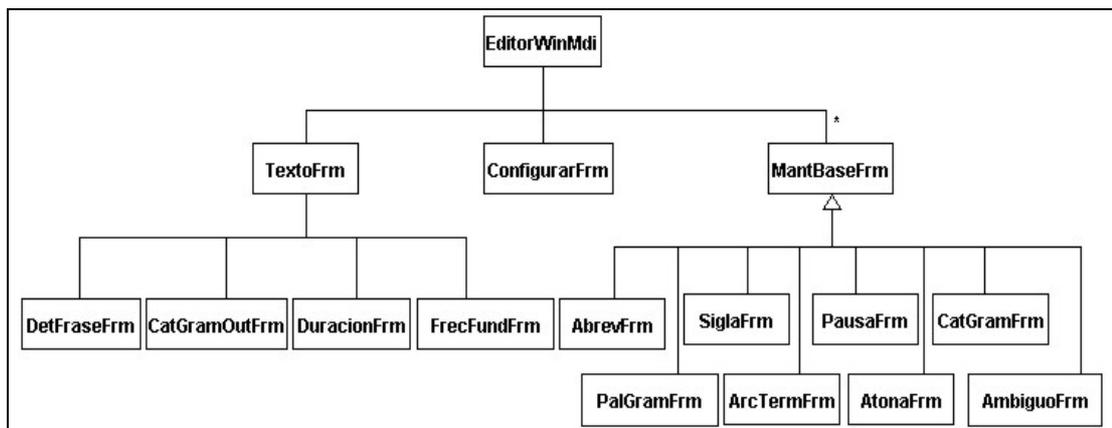


FIGURA G.1 JERARQUÍA DE VENTANAS DEL PROGRAMA LECTOR.EXE

➤ **Clases Utilizadas:** TEditorWin, TConfigurar, TTexto, TMantBase.

A continuación se describen cada una de las ventanas que tiene el programa:

- **EditorWinMdi:** Es la ventana principal del sistema, por ser de tipo MDI (Multiple Document Interface) contiene al menú principal y a todas las demás ventanas del sistema.

1. Diseño:

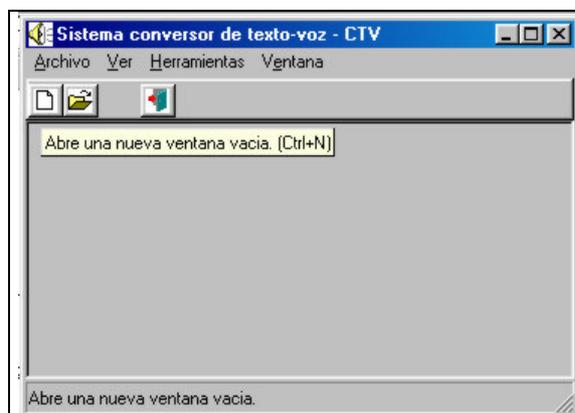


FIGURA G.2 DISEÑO DE LA VENTANA EDITORWINMDI

2. Operación:

- Para crear un nuevo documento, se deberá presionar el botón de nuevo .
 - Para abrir un archivo de texto preexistente pulsar el botón de abrir .
 - Para salir del sistema presionar el botón salir .
 - Las tres opciones anteriores se encuentran también dentro del menú "Archivo".
 - Para acceder a la configuración del sistema, ir al menú "Herramientas" y elegir la opción "Configuración" o presione "F2".
 - Para consultar las tablas de base de datos, ir al menú "Herramientas", al submenú "Tablas" y elegir la tabla que se desea ver.
- **TextoFrm:** Ventana hija de EditorWinMdi. Tiene un objeto TRichEdit que recibe el texto a leer. Tiene las siguientes funcionalidades: permite manejar la edición del texto, su lectura y visualizar los resultados del procesamiento de una frase. Para realizar la última tarea se hace uso de otras ventanas como se muestra en el diagrama de la Figura 3.2.2.4.1. Esta ventana cuenta con su propio menú Edición, Lectura y Ver que se inserta en el menú de la ventana EditorWinMdi.

1. Diseño

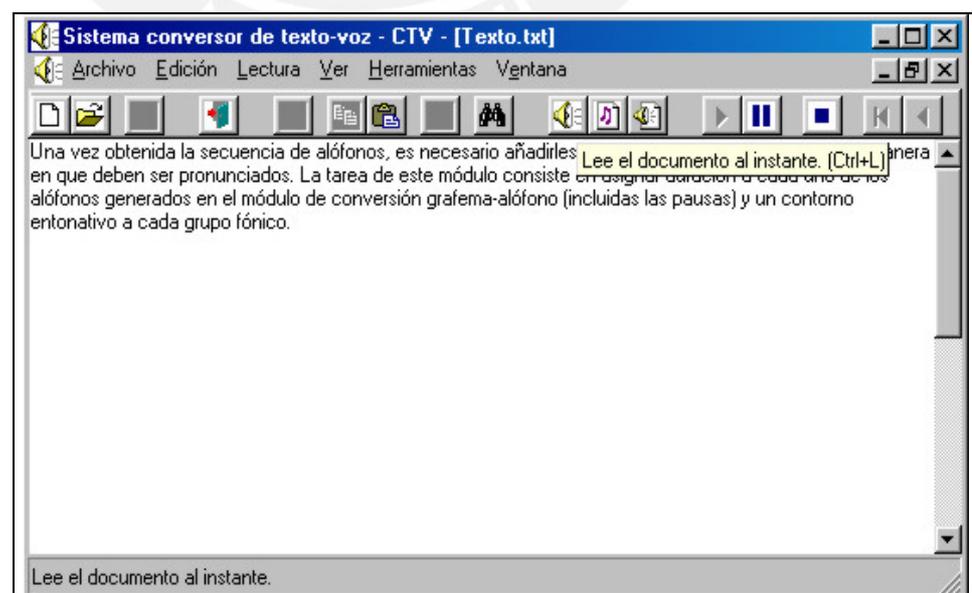


FIGURA G.3 DISEÑO DE LA VENTANA TEXTOFRM

2. Operación:

- Para iniciar la lectura de un texto se tiene tres opciones: Para escuchar la lectura del texto al instante se deberá presionar el botón de *leer* , para guardar el resultado de la lectura en un archivo de sonido se deberá presionar el botón *generar wav* , para escuchar y grabar se deberá presionar el botón *leer y generar wav* .
 - Para pausar la lectura del texto deberá presionar el botón *pausar* .
 - Para continuar con la lectura del texto deberá presionar el botón *continuar* .
 - Para detener la lectura del texto deberá presionar el botón *detener* .
 - Para repetir la lectura de la última frase procesada deberá presionar el botón *repetir frase* .
 - Para repetir la lectura de todo el documento se deberá presionar el botón *repetir* .
 - Las opciones anteriores también son accesibles desde el menú Lectura.
 - Para visualizar los resultados del procesamiento de la última frase, deberá ir al menú Ver y elegir la opción deseada.
- **ConfigurarFrm:** Esta ventana permite dar mantenimiento a la configuración del sistema. Es de tipo diálogo, y es llamado desde la ventana EditorWinMdi, como se lo indica el diagrama de la Figura 3.2.2.4.1.

1. Diseño:

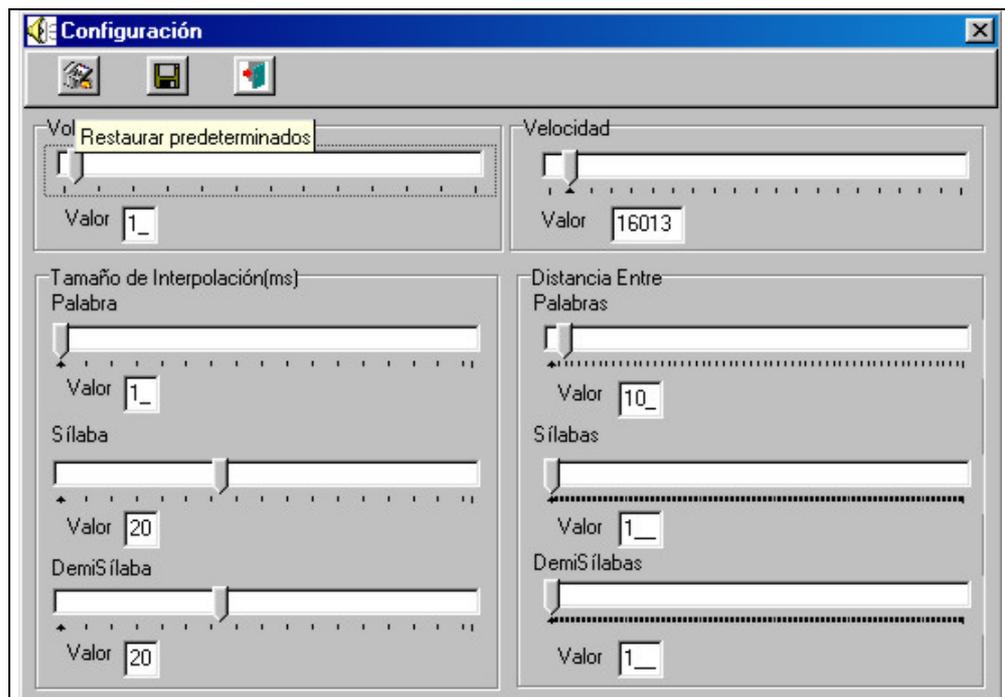
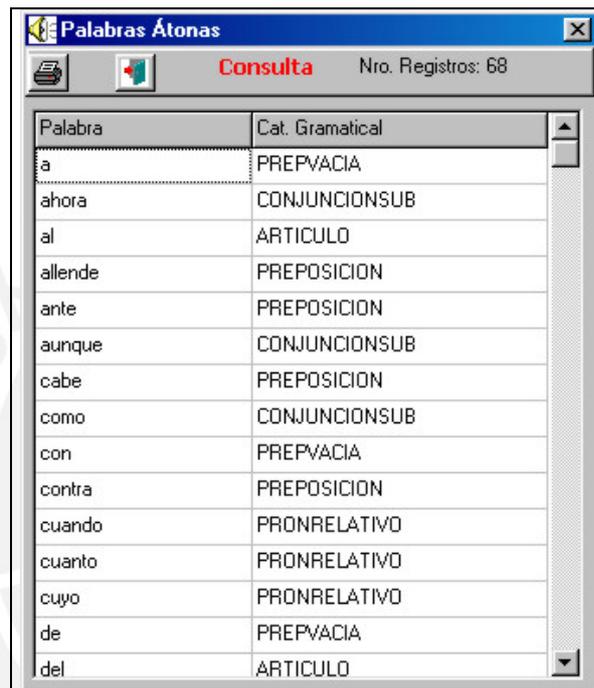


FIGURA G.4 DISEÑO DE LA VENTANA CONFIGURARFRM

2. Operación:

- Para modificar los valores de la configuración del sistema, deberá arrastrar el puntero de cada barra o escribir directamente el valor deseado en los cuadros de texto.
- Para recuperar los valores por omisión de la configuración del sistema presione sobre el botón *restaurar predeterminados*  de la barra de herramientas.
- Para guardar los cambios presionar sobre el botón *guardar* , caso contrario, para salir sin grabar presionar el botón salir .
- **MantBaseFrm:** Esta ventana es la plantilla de donde todas las pantallas de mantenimiento heredan sus propiedades y comportamiento, su diseño se muestra en la Figura 3.1.3.1.1 de la sección estándares de programación. Ahí también se especifica su operación. Las ventanas que lo heredan se muestran con un enlace de herencia en el diagrama de la Figura 3.2.2.4.1. Adicionalmente a pesar de que no está indicado en el diagrama, las ventanas CatGramOutFrm (Resultado de la categorización gramatical de una frase) y DuracionFrm (Listado de la duración de cada demisílaba) también son derivadas de la ventana MantbaseFrm. Debido a que el diseño y la funcionalidad de sus ventanas derivadas no varían sustancialmente no se muestra el

diseño ni la operación de todas las ventanas, sólo a manera de ejemplo, mostramos el diseño de la ventana AtonaFrm, como puede observarse debido a que la tabla de palabras átonas se le presenta al usuario en modo de sólo lectura, y este no puede ser modificado, se le a quitado los botones que ponían la tabla en modo consulta o de edición, y el botón grabar.



Palabra	Cat. Gramatical
a	PREPVACIA
ahora	CONJUNCIONSUB
al	ARTICULO
allende	PREPOSICION
ante	PREPOSICION
aunque	CONJUNCIONSUB
cabe	PREPOSICION
como	CONJUNCIONSUB
con	PREPVACIA
contra	PREPOSICION
cuando	PRONRELATIVO
cuanto	PRONRELATIVO
cuyo	PRONRELATIVO
de	PREPVACIA
del	ARTICULO

FIGURA G.5 DISEÑO DE LA VENTANA ATONAFRM

- **DetFraseFrm:** Esta ventana muestra el detalle de la última frase procesada, presenta cuatro cajas de edición donde muestra la frase luego de la expansión en letras de cualquier unidad especial (abreviatura, números, fechas, etc.), la frase en sílabas, en fonemas, y en demisílabas.

1. Diseño

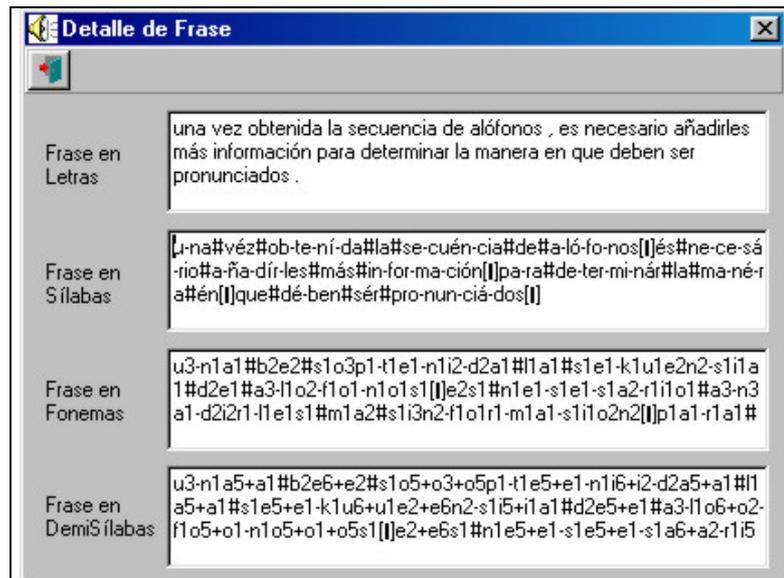


FIGURA G.6 DISEÑO DE LA VENTANA DETFRASEFRM

- **FrecFundFrm:** Esta pantalla permite visualizar el gráfico del contorno de frecuencia fundamental que se le asignó a la frase en el modelado prosódico.

1. Diseño

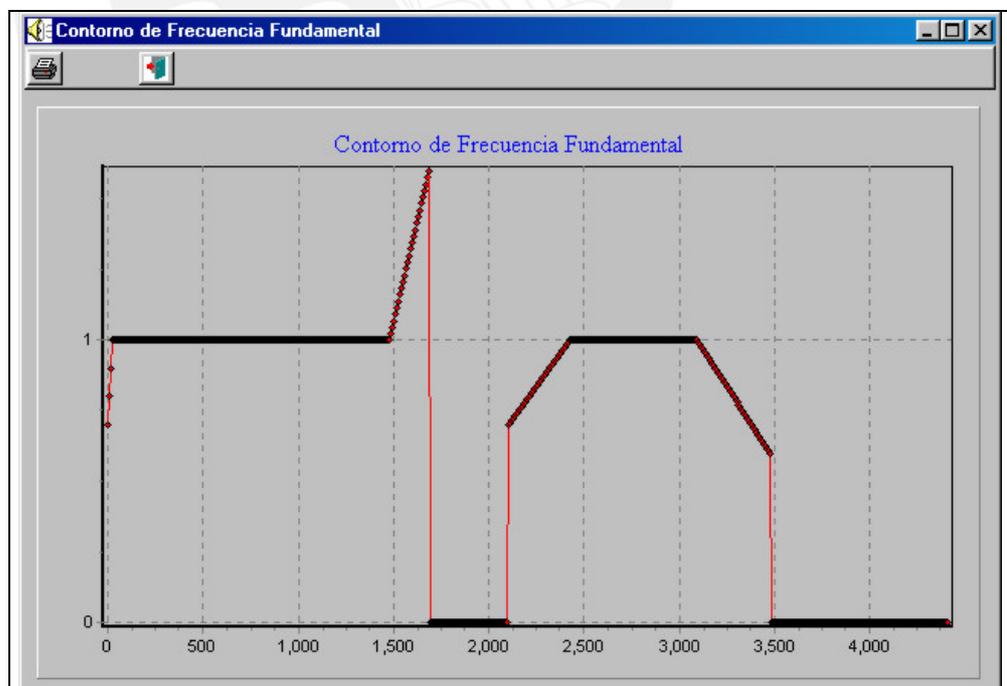


FIGURA G.7 DISEÑO DE LA VENTANA FRECFUNDFRM

2. **LectorLC.exe:**

- **Jerarquía de Ventanas:** Este programa hace uso de ventanas para su funcionamiento.
- **Clases Utilizadas:** cTexto.

H-3 Diseño de Reportes

El sistema maneja dos tipos de reportes: el primero es usado para imprimir el gráfico de contorno de frecuencia fundamental, el diseño es idéntico al presentado en la ventana FrecFundFrm de la Figura 3.2.2.4.7. El segundo tipo es usado para listar la información de los archivos de la base de datos del sistema, el diseño se corresponde también a las ventanas derivadas de MantBaseFrm. También aquí, a manera de ejemplo presentamos el diseño reporte correspondiente al archivo de palabras átonas. Los demás son similares, sólo varía el número de columnas del reporte.

1. Diseño

Palabras Átonas		Fecha: 20/11/2001
Palabra	Cat. Gramatical	Página: 1
a	PREPOSICION	
ahora	CONJUNCIONSUB	
al	ARTICULO	
allende	PREPOSICION	
ante	PREPOSICION	
aunque	CONJUNCIONSUB	
cabe	PREPOSICION	
como	PREPOSICION	

FIGURA G.8 DISEÑO DE REPORTE DE PALABRAS ÁTONAS

Anexo I

Resultados Experimentales

Para probar el sistema se llevaron a cabo cinco sesiones de audición a un total de diez personas. Los resultados de la pruebas de audición fueron las siguientes:

I-1 Números

Orden	Número	Entendieron
1	658	10/10
2	73	9/10 (*)
3	5591	10/10
4	1020	10/10
5	4357	10/10
6	8	10/10
7	7373	10/10
8	2158	10/10
9	354943	8/10 (*)
10	2	10/10
11	400	10/10
12	2345875	10/10
13	949	9/10 (*)
14	15	10/10

En la columna “Entendieron” de la tabla anterior, se muestra el “número de personas que entendieron el número” entre el “Número total de personas que escucharon”.

Los números con (*) no fueron comprendidos en su totalidad.

I-2 Fechas y Horas

Orden	Fecha / Hora	Entendieron
1	16/05/2003	10/10
2	24-09-2002	10/10
3	05:04	10/10
4	10:30:25	10/10

Todas las personas entendieron completamente las fechas y horas.

I-3 Palabras Complicadas

Orden	Palabra	Entendieron
1	Pala	8/10
2	Bala	9/10
3	Tala	7/10
4	Dala	8/10
5	Cala	8/10
6	Pila	9/10
7	Pula	10/10
8	Tila	5/10
9	Tula	10/10
10	Quila (*)	3/10
11	Prala	7/10
12	Trala	7/10
13	Crala	7/10
14	Brala	7/10
15	Drala	7/10
16	Lava	10/10
17	Labi (*)	5/10
18	Labu	8/10
19	Lada (*)	5/10
20	Ladi (*)	1/10

Las cuatro palabras que están marcadas con (*) no fueron comprendidas por la mayoría de las personas.

I-4 Oraciones

Orden	Oración
1	Sólo hizo una llamada por teléfono.
2	Esa ciudad tiene muchos charcos en el suelo.
3	Guarda bien tu llave del armario.
4	Soñó que vivía en el paraíso.
5	Me gusta el adjetivo indigente.
6	El hombre inspiraba confianza.
7	El coche se paró en el llano.
8	El chocolate estaba riquísimo.
9	No le gusta el lomo en los bocadillos.
10	El atleta pudo con su adversario.
11	Es un proyecto de cooperación internacional.
12	Detrás de una paz larga, llegó una guerra.
13	“Mientras la ciudad duerme” es un buen libro.
14	El colibrí es un ave diurna y el búho nocturna.
15	El niño gritaba y lloraba antes de comer.
16	Los reyes inspiraban confianza.
17	El alfarero giraba el torno admirablemente.
18	Le prohibió ingerir alimentos.
19	El rey jamás volverá.
20	El adjunto a dirección salió a la calle.

Sólo las oraciones 1, 9 fueron entendidas parcialmente por la mayoría de las personas, las combinación “Sólo hizo” de la oración 1 y “el lomo” en la oración 9 fueron las que no fueron entendidas por la mayoría de las personas.

Las palabras “adjetivo indigente” de la oración 5 no fue comprendida por dos personas.

El resto de las oraciones fue comprendida en su totalidad.

I-5 Párrafos

1. Servicio Municipal de Bomberos al habla. Estamos tratando de localizar una llamada urgente. Colgaron sin dar ningún detalle personal. Al parecer nos llamaron desde esta ciudad, al número 762 584. Agradeceríamos cualquier cosa que nos ayudara a identificarle puesto que parecía desesperado.
2. Por favor, quisiera pedir un taxi para mañana a primera hora. El Sr. Rodríguez, que vive en la avenida del Gral. Martínez Campos, quiere estar en la terminal internacional de Barajas alrededor de las seis y cuarto de la mañana. Su vuelo no sale hasta las ocho menos diez, pero tiene que facturar mucho equipaje. Por favor, que no se les olvide, y sobretodo sean puntuales. El Sr. Rodríguez estará esperando el taxi a las cinco y media.
3. Envíen una ambulancia al número dieciséis del Pasaje de la Concepción lo más rápido que puedan. Un anciano ha resbalado, se ha roto la pierna y no se puede mover. Al pasaje sólo se puede llegar por la calle de la Iglesia. La entrada por la Avenida del Parque está cerrada por obras. Ya hemos avisado al Hospital General y esperan al herido.
4. Para ir a Úbeda, el mejor itinerario es el siguiente: se coge la Nacional 340 por la costa, pasando por Tarragona y Castellón, hasta el cruce con la Nacional 3 en Valencia. Desde Valencia, se sigue por la Nacional 3 hasta Requena. En Requena hay un desvío hasta la Nacional 322, que llega a Úbeda pasando por Albacete. Se tarda unas 8 horas.
5. Parte nocturno del guardia municipal 3652. Durante el turno se recibieron seis llamadas. Dos allanamientos de morada, tres escándalos por borrachera y

una agresión. Uno de los borrachos fue detenido, y los otros dos sólo fueron amonestados. El agresor fue arrestado y puesto a disposición judicial.

6. Inventario del departamento de caballeros y jóvenes del 6 de marzo de 1989, sección de camisetas y géneros de punto. Camisetas de manga larga, tallas 42 a 54: tres de cada talla, en blanco, celeste y crema. Chaquetas de lana, tallas 36 a 46: cuatro de cada color de cada talla, menos de las 38 y 40, de las que sólo quedan en gris, marengo y crema.
7. Estado de las carreteras del centro de la Península correspondiente al sábado 12 de agosto. Tráfico de fin de semana denso y lento en los tres carriles dirección sur de la Nacional 4 desde Ocaña hasta la intersección con la Nacional 420. Un accidente ha bloqueado la calzada derecha de la Nacional 40 al este de Griñón. Tráfico pesado en todas las salidas de Aranjuez. Se aconseja a los conductores que eviten la zona después de las 8 de la tarde.
8. Anoche, abrí la puerta del jardín para sacar al gato. Hacía una noche tan buena que pensé en dar un paseo y respirar el aire fresco. De repente, se me cerró la puerta. Me quedé en la calle, sin llaves. Para rematarlo, me arrestaron cuando trataba de forzar la puerta para entrar.
9. A mi casa se puede ir por un atajo cruzando el bosque. Los vecinos dicen que el bosque está embrujado. Nadie quiere pasar por allí cuando oscurece. Yo, por supuesto, no soy supersticioso. A mí me encanta ir por ese camino tan pintoresco.
10. Mañana mi hija va a hacer un pastel. Insiste en mezclarlo todo a mano. Está convencida de que está más bueno de esta forma. Esto ya lo veremos cuando se haya enfriado. La verdad, no tendrá tiempo de probarlo porque sus hijos se lo habrán comido antes.

Se entendieron todos los párrafos, sólo una persona indicó que entendió el párrafo 4 a un 50% aproximadamente.

I-6 Observaciones de las personas que escucharon el sistema

Los comentarios de las personas que participaron en las audiciones fueron:

1. La naturalidad: En general, los participantes estuvieron de acuerdo en que la naturalidad era buena, pero para que se asemeje al habla humana habría que hacer mejoras.
2. El manejo de pausas entre cada palabra y oración es muy buena.
3. El nivel de comprensión es muy bueno, y sólo en algunos casos puntuales se notan uniones bruscas entre sílabas que no permiten entender completamente algunas palabras.
4. La acentuación de las palabras es muy buena, se nota en todos los casos donde va la mayor fuerza de voz.
5. Se nota el acento de un hablante limeño.
6. Se debe disminuir el eco en algunos casos.
7. Se hace dificultoso el entendimiento de las letras “rr” y “ll”.