

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



Tesis:
**“Silabeo Automático para la Generación de Vi-sílabas
en Español”**

Para Obtener el Grado de:
**Maestro en Ciencias en Ciencias de la
Computación**

Presenta:
I.S.C. José Antonio Radilla Ávila

Director:
Dra. María Lucila Morales Rodríguez



SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

SECRETARÍA DE
EDUCACIÓN PÚBLICA

Cd. Madero, Tamps; a **17 de Agosto de 2012.**

OFICIO No.: U5.160/12
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS

**C. ING. JOSÉ ANTONIO RADILLA ÁVILA
PRESENTE**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

“SILABEO AUTOMÁTICO PARA LA GENERACIÓN DE VI-SÍLABAS EN ESPAÑOL”

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE
“Por mi patria y por mi bien”

M. P. María Yolanda Chávez Cinco
M. P. MARÍA YOLANDA CHÁVEZ CINCO
JEFA DE LA DIVISIÓN



S.E.P.
DIVISIÓN DE ESTUDIOS
DE POSGRADO E
INVESTIGACIÓN
I.T.C.M.

c.c.p.- Archivo
Minuta

MYCHC 'NICO 'ayge'



Ave. 1º. de Mayo y Sor Juana I. de la Cruz, Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Tels. (833) 3 57 48 20, Fax: (833) 357 48 20, Ext. 1002, email: itcm@itcm.edu.mx
www.itcm.edu.mx



DECLARACIÓN DE ORIGINALIDAD

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones.

Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

28 de Agosto del 2012, Cd. Madero, Tamps.

Ing. José Antonio Radilla Ávila

DEDICATORIAS

*Gracias a Dios por darme la sabiduría y la inteligencia en la realización de esta Tesis.
Y por permitirme ser una persona de ejemplo a mis hermanos*

*En Memoria de mi Señor Padre Hermán:
Por haberme enseñado a luchar y a valorar las cosas que cuestan en la vida.
También por mostrarme a base de ejemplo y esfuerzo el anhelo de superación
y de no desistir jamás ante la calamidad*

*A mi madre: Ana Lilia
Por hacer de mí una gran persona a través de sus valiosos consejos,
virtudes, enseñanzas, educación y su gran amor.*

*A mi hermana: Mayola Berenice
Por su apoyo incondicional y por estar conmigo en la buenas
y en la malas. Y por creer en mí a pesar de las circunstancias.*

*A mi novia: Camila
Por su gran amor hacia a mi persona y su gran apoyo en los
momentos difíciles de mi vida. Gracias mi amor por este
gran regalo.*

*A mis Amigos: Patty y Roger
Por sus apoyos incondicionales ya que en cada momento
Me mostraron su amistad y su afecto.*

*Todo lo puedo en Cristo que me fortalece
Filipenses 4:13*

¡GRACIAS!

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a Dios por darme la salud y la sabiduría en la realización de esta Tesis.

Así mismo quiero expresar mi gratitud a la directora de esta tesis de maestría, la Dra. María Lucila Morales Rodríguez, por la dedicación, paciencia, tiempo y asesoría en la realización de este proyecto.

Agradezco al Dr. Arturo Hernández Ramírez, por su consejos y asesorías en la realización de este trabajo de Tesis, y a cada uno de los profesores que fueron parte del comité como es el M.C José Apolinar Ramírez Saldivar ya que sus consejos siempre fueron llevados a cabo y son parte fundamentales en mi persona.

Expreso mi gratitud al Dr. Juan Javier González Barbosa ya que fue la persona que me inspiró en estudiar un postgrado, y así mismo fue parte elemental en la revisión de este trabajo de Tesis.

Quiero retribuir mi agradecimiento a los Doctores Rodolfo Abraham Pazos Rangel y al Dr. José Antonio Martínez Flores ya que ellos me motivaron en la culminación de este proyecto de Tesis, brindándome consejos, asesorías y amonestaciones.

Gracias a la familia Blanco Hernández, por estar en los momentos delicados de mi vida y apoyarme con sus valiosos consejos, exhortaciones, amonestaciones y por creer en mí como persona, mi más sincero agradecimiento sin ustedes no hubiera llegado lejos.

Retribuyo mi agradecimiento al Sr. Juan Manuel Herrera Ruiz y a su apreciable esposa, ya que ellos fueron pieza fundamental, en la realización de este documento, y son las personas que me motivaron siempre a salir adelante.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT), a la Dirección General de Educación Superior Tecnológica (DGEST), y así mismo al Instituto Tecnológico de Ciudad Madero (ITCM), por todas las facilidades y apoyo otorgado para la realización de este trabajo de investigación.

RESUMEN

Uno de los campos de investigación más activo en el modelado y animación de mundos virtuales, es la animación facial de modelos humanoides e imaginarios que habitan éstos mundos, los cuales pueden ser desarrollados en 2D y 3D.

La generación de vi-sílabas como técnica de sincronización de labios en personajes virtuales permitirá aumentar la credibilidad de la animación facial. En esta investigación, se presenta la creación de vi-sílabas de frases escritas en el idioma español aplicando fundamentos lingüísticos para seleccionar uno ó dos visemas y asociarlo a una sílaba (vi-sílaba).

Una vi-sílaba muestra a los visemas más representativos de una sílaba para así reducir los movimientos labiales. Esta reducción, permitirá crear una sincronización labial más creíble, dado que un visema es la reproducción visual de los fonemas, y que una sílaba esta compuesta de dos ó más fonemas.

En esta tesis se desarrolló la implementación de las reglas de fonemización propuesta por Ríos Mestre [Ríos Mestre, 1999] en el lenguaje de programación Python. Estas reglas tienen la finalidad de dividir una frase ó palabra en forma de sílabas. Se aplicó minería de datos para seleccionar que punto de articulación es el que representa el fonema visual (visema) a una sílaba.

Se utilizó el algoritmo de clustering K-Means, con la finalidad de agrupar las sílabas en once grupos que representan los once puntos de articulación que existen. Posteriormente se aplicó el clasificador C4.5, para representar que animación es la correcta y que punto de articulación debe mostrarse en la consola de PANDA3D, se utilizó las librerías orange para implementar los métodos de minería de datos. Para sincronizar la voz con la animación de las vi-sílabas, se utilizó la librería pyTTS de Python.

ABSTRACT

One of the most active areas of research in modeling and animation of virtual worlds is the facial animation of imaginary and humanoid models, which can be developed in 2D and 3D.

The generation of vi-syllables as lip synchronization technique in virtual characters will increase the credibility of facial animation. In this research we create vi-syllable related to phrases in the Spanish language using linguistic fundamentals to select one or two visemes and associate it to one syllable (vi-syllable).

A vi-syllable associates visemes choosing only the most representative of a syllable in order to reduce the lip movements. This reduction will create more believable lip synchronization, since a syllable is composed of two or more phonemes.

This thesis implementation of the rules propuses by Ríos Mestre [Ríos Mestre, 1999] in the Python programming language. Data mining was used to select which is the pivot point that represents the visual phoneme (viseme) to a syllable.

Panda3D was used to animate the visemes for each syllable. These animations were previously processed with the use of Orange, a data mining libraries of Python. The K-Means and C4.5 algorithms were applied to choose the point of articulation that is display in the Panda3D console. In order to synchronize the voice with the animation of vi-syllables with used a text to speech library called pyTTS.

Tabla de Contenido

DECLARACIÓN DE ORIGINALIDAD	ii
DEDICATORIAS	iv
AGRADECIMIENTOS	v
RESUMEN	vi
ABSTRACT	vii
Capítulo 1. Introducción	1
1.1 Definición del problema	1
1.2 Justificación	2
1.3 Objetivos	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos	2
1.4 Alcances y Limitaciones	2
Capítulo 2. Animación de vi-sílabas en personajes virtuales	4
2.1 Animación de personajes virtuales	4
2.1.1 Personajes Virtuales Creíbles	4
2.1.2 Sincronización Labial	5
2.1.3 Visemas	6
2.2 Fundamentos de Lingüística	6
2.2.1 Fonología y Fonética	6
2.2.2 Grafemas y Alfabeto fonético Internacional	7
2.2.3 Fonemas, Alófonos y Archifonemas	7
2.3 Procesos de Fonemización	8
2.3.1 Proceso de Presilabeo	10
2.3.2 Proceso de conversión grafema a fonema	10
2.3.3 Proceso de Silabeo	12
2.3.3.1 Sílabas	12
2.3.3.2 Reglas de Silabeo	12
2.3.4 Proceso de Ajuste Silábico	14
2.3.5 Proceso de Punto de Articulación	14
Capítulo 3. Estado del Arte de la Sincronización Labial	21
3.1 Trabajos relacionados con la Sincronización Labial	21
3.2 Trabajos relacionados con la Selección de Visemas	23
3.3 Trabajos relacionados con la creación de vi-sílabas	27
3.4 Comparación de Trabajos relacionados	29
Capítulo 4. Arquitectura de Silabeo Automático para la generación de Vi-sílabas en español	32
4.1 Arquitectura General	32

4.2 Modelo de Silabeo Automático	33
4.3 Transcripción por Punto de Articulación.....	34
4.4 Modelo de Selección de Vi-sílabas	35
4.5 Modelo de Sincronización Labial.....	36
Capítulo 5 Evaluación y Análisis de Resultados	38
5.1 Asignación de un punto de Articulación a una sílaba	38
5.2 Selección de Visemas que se asociarán a una sílaba	45
Capítulo 6 Conclusiones y Trabajos Futuros	49
Anexos A. Procesos de Fonemización	51
A.1 Implementación del Algoritmo de Presilabeo	51
A.2. Implementación del Algoritmo de Transcripción Fonológica.....	53
A.3. Implementación del Algoritmo de Silabeo	64
A.4. Implementación del Algoritmo de Ajuste Silábico	70
Anexos B	77
B.1 Implementación del Algoritmo por Punto de Articulación	77
Anexos C	79
C.1. Implementación del Algoritmo de Selección de Visemas (Panda3D).....	79
C.2. Implementación del Algoritmo de K-means.....	81
C.3 Aplicación del Algoritmo de Minería de Datos	82
Anexos D.....	85
D.1 Aplicación del Algoritmo de TTS en Python	85
Anexos E.....	90
Publicaciones	90
Glosario.....	91
Referencias Bibliográficas.....	95

Índice de Figuras

Figura 1. Diagrama Adaptación General del DEFE.....	9
Figura 2. Clasificación de los fonemas a través del Punto de Articulación	15
Figura 3. Clasificación de los fonemas a través del Modo de Articulación	15
Figura 4. Diagrama del sistema completo. Panel Izquierdo, Unidad de Análisis. Panel derecho Unidad de Despliegue Gráfico.....	22
Figura 5. Método usado para la validación del sistema de sincronización labial.....	23
Figura 6. Parámetros de Definición facial de la boca.....	26
Figura 7. Animación de voz basada en <i>vi-sílabas</i> en tiempo real.	28
Figura 8. Arquitectura General para la generación de Vi-sílabas.....	32
Figura 9. Modelo de Silabeo Automático para la generación de Vi-sílabas.	33
Figura 10. Modelo de Selección de Vi-sílabas.	36
Figura 11. Modelo de Sincronización Labial en tiempo real.	37
Figura 12. Asignación de los once grupos ó centroides a cada punto de articulación.	45
Figura 13. Primer Caso de Prueba: cada fonema de la palabra tiene un visema asociado. ..	46
Figura 14. Segundo Caso de Prueba: se anima el primer y ultimo fonema de una sílaba. ...	46
Figura 15. Tercer Caso Prueba. Se realiza la elección del fonema-punto de articulación que mejor represente a una sílaba.	47
Figura 16. Clasificación de las sílabas por los métodos, Bayes, knn, C45 y Tree.	48
Figura 17. Se presentan las sietes reglas de Presilabeo.	51
Figura 18. Se presentan las veintisietes reglas de Trascricpción de grafema a fonema.....	55
Figura 19. Se presentan las seis reglas que fueron añadidas en la conversión fonológica. ...	56
Figura 20. Se presentan las dieciochos reglas de Silabeo.....	65
Figura 21. Reglar1r2r3_ajustada realiza el proceso de silabeo del grafema <ch>.....	66
Figura 22. Se presentan las diecisiete reglas de Ajuste silábico.....	71
Figura 23. Carga del actor Ralph con todas las posibles combinaciones de visemas.	81
Figura 24. Base de datos de entrenamiento, usando el algoritmo de k-means para Python	82
Figura 25. Asignación de los once grupos ó centroides a cada punto de articulación.	82

Índice de Tablas

Tabla 1. Ejemplo de Aplicación de las reglas de Presilabeo	10
Tabla 2. Clasificación de los fonemas a-d correspondientes a los grafemas (a-d).....	10
Tabla 3. Clasificación de los fonemas e-w correspondientes a los grafemas (e-hu).....	11
Tabla 4. Ejemplo de aplicación de las reglas Fonológicas (FON1.REG)	11
Tabla 5. Clasificación de las reglas de silabeo (vocales y consonantes).....	13
Tabla 6. Ejemplo de aplicación de las reglas de silabeo (FON2.REG).....	13
Tabla 7. Ejemplos de aplicación de las reglas de ajuste silábico	14
Tabla 8. Clasificación de las vocales por su punto de articulación.	16
Tabla 9. Clasificación de los grafemas A-M y su relación con el punto de articulación.	17
Tabla 10. Clasificación de los grafemas N-V y su relación con el punto de articulación. ...	18
Tabla 11. Clasificación de los grafemas W-Z y su relación con el punto de articulación. ..	19
Tabla 12. Elección de los Identificadores correspondientes al punto de articulación.	19
Tabla 13. Aplicación de las reglas de punto de articulación a la palabra desyelo.....	20
Tabla 14. Tabla de clasificación de grupos de FAPs.....	24
Tabla 15. Visemas definidos por MPEG-4 y los fonemas generados por Festival.	25
Tabla 16. Algoritmo1: Seguimiento asistido de punto del contorno externo	27
Tabla 17. Tabla comparativa de los trabajos relacionados.....	31
Tabla 18. Análisis de frecuencia por Punto de Articulación para el grupo C1.	39
Tabla 19. Análisis de frecuencia por Punto de Articulación para el grupo C2.	39
Tabla 20. Análisis de frecuencia por Punto de Articulación para el grupo C3.	40
Tabla 21. Análisis de frecuencia por Punto de Articulación para el grupo C4.	40
Tabla 22. Análisis de frecuencia por Punto de Articulación para el grupo C5.	41
Tabla 23. Análisis de frecuencia por Punto de Articulación para el grupo C6.	41
Tabla 24. Análisis de frecuencia por Punto de Articulación para el grupo C7.	42
Tabla 25. Análisis de frecuencia por Punto de Articulación para el grupo C8.	42
Tabla 26. Análisis de frecuencia por Punto de Articulación para el grupo C9.	43
Tabla 27. Análisis de frecuencia por Punto de Articulación para el grupo C10.	44
Tabla 28. Análisis de frecuencia por Punto de Articulación para el grupo C11.	44
Tabla 29. Clasificación de los grafemas a fonemas y su punto de articulación.	78
Tabla 30. Clasificación de las sílabas por los métodos, Bayes, knn, C45 y Tree.....	84
Tabla 31. Clasificación del punto de articulación y su relación con el visema adecuado....	89

Capítulo 1

Introducción

Uno de los campos de investigación más activo en el modelado y animación de mundos virtuales, es la animación facial de modelos humanoides e imaginarios que habitan éstos mundos, los cuales pueden ser desarrollados en 2D y 3D.

Nuestro trabajo se interesa en aumentar la credibilidad de la sincronización labial en personajes virtuales, presentando una propuesta para la asignación de visemas a una sílaba (vi-sílabas) de frases escritas en el idioma español aplicando fundamentos lingüísticos.

1.1 Definición del problema

El problema de la sincronización labial, es ser demasiado literal en la animación del habla de un personaje, intentando animar cada una de las letras que componen las palabras, en vez de solamente acentuar los sonidos (fonemas y alófonos) principales que son necesarios para comunicar la “idea”. Los principiantes en la representación de la sincronización labial, crearán una replica exacta de la cara diciendo una ‘letra’ en forma aislada, como cuando se dice el sonido de una letra por si sola [Lango, 2006]. Esto no es igual que cuando esta letra, va acompañada de otra letra (consonante ó vocal). Por ejemplo, la “t” por si sola suena como “ti” y su forma de boca es diferente cuando ésta “t” va acompañada de una consonante como en “trompo”.

En este proyecto se busca automatizar la generación de la sincronización labial en personajes virtuales, partiendo de una descomposición en sílabas de la expresión oral de un conjunto de locuciones; con el objetivo de identificar los fonemas relevantes y sus visemas asociados.

1.2 Justificación

Con la creación de vi-sílabas, se permitirá elegir sólo los visemas más representativos de una sílaba para así reducir los movimientos labiales. Esta reducción permitirá crear una sincronización labial más creíble, dado que un visema es la reproducción visual de los fonemas, y que una sílaba puede estar compuesta de dos ó más fonemas visuales (visemas).

Con la implementación del proceso de Silabeo automático, se divide una palabra en sílabas; lo que permitirá que posteriormente cada una de éstas sílabas sean representadas por visemas, permitiendo así la sincronización visual de una sílaba, dicha agrupación la hemos denominado (vi-sílabas).

1.3 Objetivos

1.3.1 Objetivo General

El objetivo general de nuestra propuesta es la creación de vi-sílabas de frases escritas en el idioma español aplicando fundamentos lingüísticos para realizar el proceso de silabeo.

1.3.2 Objetivos Específicos

- Automatizar el proceso de transcripción Fonética y Fonológica de una frase aplicando fundamentos lingüísticos.
- Automatizar el proceso de generación de sílabas emitidas en el idioma español.
- Creación de un corpus de visemas y vi-sílabas.
- Sincronizar la expresión facial de un personaje virtual con la expresión oral resultado de un acto locutorio en el idioma español.

1.4 Alcances y Limitaciones

Este proyecto aplica el reconocimiento de una sílaba como unidad fonológica, a

partir de los procesos lingüísticos propuestos por Ríos Mestre [Ríos Mestre, 1999], y lo adapta para el español de México.

Este proyecto se encuentra limitado a sílabas que contenga como máximo seis fonemas y como mínimo sólo un fonema.

En este proyecto de tesis, no se toma en cuenta el modo de articulación, es decir, los procesos que se ponen en marcha para pronunciar un fonema basados en las cuerdas vocales, lengua, velo del paladar, paladar, alvéolos, dientes, labios y la cavidad nasal.

Cabe resaltar que en este proyecto solo se basó en el punto de articulación de los fonemas es decir, el lugar en la boca donde se pronuncia un determinado fonema (como son: los labios, la lengua, los dientes, y el paladar), ya que es la parte mas visible que se encuentra en la expresión labial, al momento de llevar a cabo la pronunciación de los fonemas. Por lo que en caso de existir variantes en la pronunciación de un mismo fonema (alófonos) debido a su posición en una sílaba, estos no son diferenciados.

Capítulo 2

Animación de vi-sílabas en personajes virtuales

La creación y animación de vi-sílabas en personajes virtuales implica el conocimiento de conceptos tales como sincronización labial, personajes virtuales, sílabas y visemas. De igual forma es importante conocer conceptos del área de lingüística ya que son aspectos fundamentales en la realización de ésta tesis.

2.1 Animación de personajes virtuales

El enfoque actual de la Inteligencia Artificial es el desarrollo de animaciones faciales de modelos humanoides e imaginarios en 3D. Con el objetivo de obtener más credibilidad en las simulaciones de personajes virtuales, se ha buscado establecer la forma correcta de los movimientos labiales a tal grado que se logre una mejor sincronización de las locuciones.

2.1.1 Personajes Virtuales Creíbles

Un personaje virtual es un actor animado representado por medio de una computadora que coexiste dentro de un universo virtual que podría interactuar con el usuario [Bailenson & Blascovich, 2004]. La principal finalidad del personaje virtual es crear una sensación de inmersión asociada a la experiencia emocional y social dentro del entorno virtual [Bailenson & Blascovich, 2004].

La animación es un proceso utilizado para dar la sensación de movimiento a imágenes o dibujos. En nuestro trabajo se busca mejorar la credibilidad de la sincronización labial, es decir, la animación de los labios de un personaje para que acompañen a la articulación de las frases emitidas para el idioma español.

2.1.2 Sincronización Labial

Se define sincronización labial como los movimientos emitidos por los labios cuando se elige el visema adecuado en relación con una voz humana artificial ó un archivo de audio [Abadía & Fernández, 2010].

Según Keith Lango [Lango, 2006], la tendencia para la generación de sincronización labial es la siguiente:

- Construir *visemas* (representaciones) para los “sonidos” (fonemas), por ejemplo la M, E, S, etc., algunos van más allá y hacen representaciones para “sonidos” como la H.
- Escuchar la pista de audio o la grabación de voz.
- Para cada sonido que escuchan, se da al “sonido” una representación de una imagen cercana o igual al 100%.
- Realizar una representación (render) de la animación de sincronización labial.

El problema de seguir esta tendencia, es ser demasiado literal en la animación del habla de un personaje, intentando animar cada una de las letras que componen las palabras, en vez de solamente acentuar los sonidos (fonemas y alófonos) principales que son necesarios para comunicar la “*idea*”. Es por ello que Keith Lango, propone dos principios básicos para solución de este problema [Lango, 2006].

Principio #1: las letras no son sonidos. Los sonidos no son letras, no hay letras en la animación de sincronización labial.

Generalmente los principiantes crearán una replica exacta de la cara diciendo una ‘letra’ en forma aislada, como cuando se dice el sonido de una letra por si sola. Esto no es igual que cuando va acompañada de otra letra (consonante ó vocal). Por ejemplo, la “t” por si sola suena como “ti” y su forma de boca es diferente cuando ésta “t” va acompañada de una consonante como en “trompo”.

Principio #2: Las formas de la boca para los sonidos deben ser animadas en contexto.

La forma de sonido que precede afecta a la forma de sonido actual. De igual modo, la siguiente forma de sonido es anticipada por la forma de sonido actual. Cuando un personaje dice una letra, por ejemplo la “U”, el personaje tiene la boca en forma de “U”, antes de que esta se escuche y cuando la “U” ya se escuchó, el personaje ya esta empezando a prepararse para la siguiente letra.

2.1.3 Visemas

Se conoce como *visemas* a los movimientos de labios en donde la boca asume una posición cuando se produce un determinado fonema [Cletus, 1968], por lo que un visema es la representación visual de un fonema.

El término visema fue introducido por Cletus en 1968 como una abreviación del término visual phoneme y denota grupos de consonantes que forman clases exclusivas mutuamente. En un estudio Cletus probó como los fonemas en la palabra inicial y final son percibidos visualmente [Engström, 2003].

2.2 Fundamentos de Lingüística

En éste trabajo se revisaron conceptos asociados a la Fonética y la Fonología, que son dos ramas de la Lingüística.

2.2.1 Fonología y Fonética

La fonología estudia los elementos fónicos, o unidades, de una lengua desde el punto de vista de su función [Obediente, 1998, & Alvar, 2000]. La fonética es la ciencia que estudia los sonidos del lenguaje en el plano del habla [Obediente, 1998].

La fonética es la encargada de estudiar los sonidos del lenguaje, es decir los alófonos y la fonología es la disciplina encargada de poder representar estos sonidos desde el punto de vista de la lengua, es decir los *fonemas*.

Cuando queremos trasladar los fonemas a la escritura, se utilizan representaciones gráficas a las que normalmente se llaman letras ó, grafemas [Obediente, 1998 & Alva, 2000].

2.2.2 Grafemas y Alfabeto fonético Internacional

Un *grafema* es la mínima unidad distintiva de un sistema de escritura, o sea, el mínimo elemento por el que se pueden distinguir por escrito dos palabras en una lengua. Se representa entre < > [Iribarren, 2005].

El Alfabeto Fonético Internacional es un sistema de transcripción fonética (notación fonética) creado por lingüistas. Su propósito es otorgar en forma regularizada, precisa y única la representación de los sonidos de cualquier lenguaje oral. Los símbolos del Alfabeto Fonético Internacional están divididos en tres categorías: letras (que indican sonidos “básicos”), diacríticos (que especifican esos sonidos) y suprasegmentales (que indican cualidades tales como velocidad, tono y acentuación). Estas categorías están divididas en secciones menores: las letras están divididas en vocales y consonantes, y los diacríticos y suprasegmentales están divididos según si indican articulación, fonación, tono, entonación o acentuación con relación al fonema dado [De la Mota Gorriz, 1995 & Ríos Mestre, 1999].

2.2.3 Fonemas, Alófonos y Archifonemas

Un *fonema* es la representación abstracta y generalizada de un sonido con capacidad para diferenciar significados (ver figura 1). Se escribe entre // [Gil Fernández, 2007].

Los fonemas de un idioma pueden ser clasificados en subgrupos basados en sus apariencias visuales. Estos subgrupos son llamados visemas y pueden ser considerados como equivalencia visual a los fonemas. Cómo se ha mencionado, los visemas pueden ser descritos como formas clave de la boca donde cada forma vocal corresponde a uno o más fonemas [Engström, 2003]. Podemos decir que fonema es una unidad fonológica diferenciadora, indivisible y abstracta.

Diferenciadora: porque cada fonema se delimita dentro del sistema por las cualidades que se distingue de los demás y además es portador de una intención significativa especial. Demostración: /p-e-s-o/ y /b-e-s-o/ son dos palabras que se distinguen

semánticamente debido a que /b/ se opone a /p/ por la sonoridad.

Indivisible: no se puede descomponer en unidades menores. Demostración la sílaba o el grupo fónico sí pueden fraccionarse. Un análisis pormenorizado del fonema revela que está compuesto por un haz de diversos elementos fónicos llamados rasgos distintivos cuya combinación forma el inventario de fonemas. El inventario de rasgos distintivos es asimismo limitado viene a constituir una especie de tercer articulación del lenguaje.

Abstracta: no son sonidos, sino modelos o tipos ideales de sonidos. La distinción entre sonido y fonema ha sido un gran hallazgo en los últimos tiempos.

Se define alófonos a cada una de las variantes que se dan en la pronunciación de un mismo fonema, según la posición de este en la palabra o sílaba, según el carácter de los fonemas vecinos y se representan entre corchetes ([]) [Obediente, 1998].

El archifonema es el conjunto de rasgos distintivos comunes a los dos fonemas de la oposición neutralizada. Para que se realice la neutralización de los fonemas /p, b, t, d, m, n k, g, r, r/ deben estar en posición implosiva. El archifonema se transcribe por medio de las letras mayúsculas y como se trata en el nivel fonológico debe ir entre barras. Los archifonemas se representan igual en ambos alfabetos: AFI y RFE y al sonido que se debe diferenciar al momento de escuchar la neutralización de dos fonemas [Mozas, 1996].

2.3 Procesos de Fonemización

Con la finalidad de asociar un visema a una sílaba (vi-sílaba) y realizar una adecuada sincronización labial utilizamos los procesos de Fonemización presentados en el documento de Ríos Mestre [Ríos Mestre, 1999] denominado “*La transcripción fonética automática del diccionario electrónico de formas simples flexivas del español: estudio fonológico en el léxico*”.

En éste documento, el autor propone la construcción de un Diccionario Electrónico Fonético del Español (DEFE) aplicando un conjunto de reglas (ver Figura 1). El DEFE esta constituido de varios módulos con el objetivo de simplificar frases y que éstas frases fueran verificadas en cada proceso, a continuación se presentan los cuatros módulos para el

proceso de Fonemización: 1) Reglas de transcripción grafema-fonema (FON1.REG), 2) Reglas de silabación (FON2.REG), 3) Reglas de ajuste silábico (FON3.REG), 4) Reglas de ajuste Fonémico (FON4.REG). El resultado de la aplicación de las reglas de cada módulo es la entrada para la aplicación de otro grupo, en una sucesión ordenada.

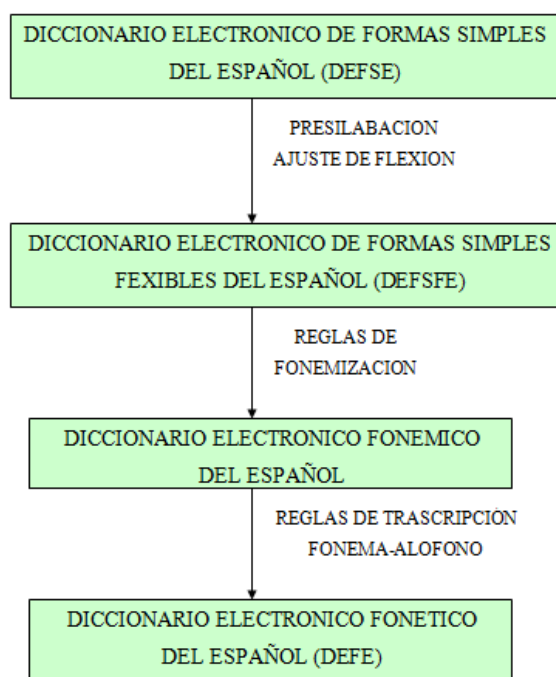


Figura 1. Diagrama Adaptación General del DEFE

Como primer paso se tendrá como entrada una palabra. Esta palabra pasará por cada uno de los procesos de fonemización en donde se insertarán marcas de división silábica en el módulo de Presilabeo. Posteriormente esta cadena presilabificada pasará al módulo de transcripción fonológica, en donde se tendrá los fonemas representativos. Se efectuarán algunos ajustes fonológicos a ciertas palabras que eran extranjerismos, con la finalidad de que solo fueran fonemas del español México.

Después se realizó un proceso de silabeo en donde la palabra estará dividida en sílabas. Y un ajuste silábico a aquellas palabras modificadas en los ajustes fonológicos. Por último la palabra silabificada se procesó en el módulo de transcripción por punto de articulación con la finalidad de procesar que órganos es el que participan al momento de pronunciar un determinado fonema.

A continuación se describen los procesos involucrados en los diferentes módulos de

reglas que conforman el proceso de fonemización.

2.3.1 Proceso de Presilabeo

Las reglas de Presilabeo actúan antes de que se apliquen el proceso de conversión de grafema a fonema. El proceso de Presilabeo consiste en la inserción de una marca de división silábica en aquellas palabras que son consideradas como excepciones de las reglas de silabeo. Por ejemplo, en la palabra *deshielo* la aplicación de la regla 2, agrega una marca de división silábica antes de la h (*ver Tabla 1*). Este proceso tiene la finalidad de que cuando se aplique las reglas fonológicas se haga una correcta transcripción del grafema a fonema.

Ejemplo	Entrada	Procedimiento	Salida	Regla 2
deshielo	CCV	_ds,, _ei,, sn,, h,, ,, ,, ,, ,, -0,=0.	C-CV	des-hielo

Tabla 1. Ejemplo de Aplicación de las reglas de Presilabeo

2.3.2 Proceso de conversión grafema a fonema

El idioma español tiene 29 letras o grafemas para 24 fonemas. (*Ver tabla 2 y 3*). Esto produce un desajuste que es el origen de las faltas de ortografía y que se debe a los cambios experimentados por el idioma a lo largo del tiempo.

Fonemas	Grafemas	Ejemplo
/a/	<a>	casa
/b/	 <v> <w>	bobina vivir wolframio
/ch/	<ch>	Cháchara
/d/	<d>	dado

Tabla 2. Clasificación de los fonemas a-d correspondientes a los grafemas (a-d).

/e/	<e>	leche
/f/	<f>	foco
/g/	<g> <gu>	Gárgola guerra
	<h>	Hombre humo
/i/	<i> <y>	Ilícito rey buey
/x/	<j> <g> <x>	Ejemplo general México
/k/	<c> <qu> <k>	Cacao queso kiosco
/l/	<l>	Libélula
/ll/	<ll>	Sabadell lluvia
/m/	<m>	mamífero
/n/	<n> <m>	invento campo
/ñ/	<ñ>	ñoñez
/o/	<o>	ornitorrinco
/p/	<p>	paralelepípedo
/r/	<r>	árbol aire
/rr/	<rr> <r>	arroba rosa
/s/	<s> <c> <z>	sisal cera azul
/t/	<t>	taco
/u/	<u> <ü>	tuerto cigüeña pingüino
/y/	<y> <ll>	Ayuno allí lluvia
/w/	<hu>	hueso huevo

Tabla 3. Clasificación de los fonemas e-w correspondientes a los grafemas (e-hu).

El proceso de conversión de grafema a fonema se realizó aplicando las reglas de transcripción fonológicas presentadas en el proceso de fonemización de Ríos-Mestre. Las reglas de transcripción fonológicas representan sólo los elementos fónicos (fonemas) dotados de función lingüística. Se representan entre barras oblicuas // [Cabiedes F. P. et, al., 2006]. Continuando con el ejemplo de la palabra *deshielo*, al aplicar la regla 5 se convertirá el grafema i al fonema y (ver Tabla 4).

Ejemplo	Entrada	Procedimiento	Salida	Regla 5
des-hielo	i	„_aáeéííoóuú-, h,, i,, aeou,, ,, ,, ,, y0.	y	des-yelo

Tabla 4. Ejemplo de aplicación de las reglas Fonológicas (FON1.REG)

2.3.3 Proceso de Silabeo

El Silabeo o Silabificación es la operación consistente en la división en sílabas pronunciando lentamente la separación de una a otra sílaba [Gil Fernández, 2007].

2.3.3.1 Sílaba

Se llama sílaba a cada una de las entidades fonéticas en las que se divide una palabra [Núñez, R., et al., 1999]. Existen dos tipos de sílabas, la sílaba abierta y la sílaba cerrada, la primera corresponde a la terminación de una vocal mientras que la segunda corresponde a la terminación de una consonante. La división silábica se realiza mediante guiones (-) y dependiendo del número de sílabas una palabra puede ser:

Monosílaba: Una sola sílaba; en español no se acentúa salvo que haya dos palabras iguales para diferenciarlas (tilde diacrítica). Ej: sol, sí, más, dos, sed.

Bisílaba: Dos sílabas. Ejemplos: calor, mano, árbol.

Trisílaba: Tres sílabas. Ejemplos: repetir, tímbrico, recoger.

Polisílaba: Más de tres sílabas. Ejemplos: azulado, diccionario, policlínica.

2.3.3.2 Reglas de Silabeo

Nuestro proceso de Silabeo esta basado en las reglas de silabación (silabeo) propuestas en el “*Diccionario Electrónico fonético Español*” (DEFE) [Ríos Mestre, 1999].

Las reglas de Silabeo se dividen en dos grupos, el de las vocales y las consonantes. En la Tabla 4 se presenta tres letras en mayúsculas como es la V que es la abreviatura de las vocales, la C que es la abreviatura de las consonantes y la letra R es el número de reglas que se esta aplicando.

Grupo de Reglas		Reglas
consonantes	VCV → V-CV	R1,R2 y R3
	VCCV → V-CCV	R4 y R5
	C s V → C - s V	R6 y R7
	CC → C-C	R8,R9,R10,R11y R12
vocales	V [-alta] - V [-alta]	R13
	V [+alta] - V[+alta]	R16,R17y R18
	V [+alta] - V [-alta]	R15
	V [-alta] - V [+alta]	R14

Tabla 5. Clasificación de las reglas de silabeo (vocales y consonantes)

Las *reglas de silabeo* para las *consonantes* marcan ataques silábicos; es decir que se encuentra en posición inicial de una sílaba. *El signo de silabación* se inserta ante una consonante situada en el foco:

X,, C (foco),, Y → X,, - C,, Y.

caSa → ca -Sa

Las reglas de silabeo para las vocales que siguen el siguiente esquema VV → V - V son consideradas diptongo en español, las reglas han de marcar los hiatos, es decir la separación de dos vocales juntas. Continuando con el ejemplo de la palabra *deshielo* al aplicar la regla 1 del proceso de silabeo (*ver tabla 6*) se insertará una marca de división silábica entre el fonema *e* y *l*.

Ejemplo	Entrada	Procedimiento	Salida	Regla 1
des-yelo	VCV	,, ,, aáeéííoóuú,, ptkbygfzxmñlLrRw ,, aeou,, ,, ,, ,, y0.	V-CV	des-ye-lo

Tabla 6. Ejemplo de aplicación de las reglas de silabeo (FON2.REG)

2.3.4 Proceso de Ajuste Silábico

Este proceso realiza la corrección de palabras que resultan erróneas en los extranjerismos, por ejemplo /án-g-s-trom/ → /ans-trom/. La finalidad es eliminar aquel fonema que no tiene una interrepresentación fonética en el idioma español, como se observa en el ejemplo el fonema que será eliminado es la /g/. Se presenta en la tabla (Ver tabla 7) con algunos ejemplos de palabras en donde se han aplicado los procesos de fonemización.

Ejemplo	Reglas de Presilabeo	Reglas de Transcripción fonológica	Reglas de Silabeo	Reglas de Ajuste Silábico	
				FON3.1	FON3.2
deshielo	des-hielo	des-yelo	des-ye-lo	des-ye-lo	des-ye-lo
piár	pi-ár	pi-ár	pi--ár	pi--ár	pi-ár
examen	examen	eksamen	ek-sa-men	ek-sa-men	ek-sa-men
ángstrom	ángstrom	ángstrom	án-g-s-trom	án--strom	án-strom
postdata	postdata	postdata	pos-t-da-ta	pos--da-ta	pos-da-ta

Tabla 7. Ejemplos de aplicación de las reglas de ajuste silábico

El ejemplo de la palabra “*deshielo*” no sufrió ningún cambio en este proceso de ajuste silábico por ser una palabra del idioma español. Pero en las palabras extranjeras como *angstrom*, *postdata*, se aplicaron los ajustes silábicos correspondientes.

2.3.5 Proceso de Punto de Articulación

A la hora de clasificar los fonemas, se utilizan dos criterios principales: el modo de articulación, que se refiere a los procesos que se ponen en marcha para pronunciar un fonema (las cuerdas vocales, lengua, velo del paladar, paladar, alvéolos, dientes, labios y la cavidad nasal) y el punto de articulación, es decir, el lugar en la boca donde se pronuncia un determinado fonema. [Poch-Olivé,1999].

A continuación se presentan las figuras 2 y 3 que corresponden a la clasificación de los fonemas con respecto al punto y modo de articulación.

RASGOS MOTIVADOS POR EL PUNTO DE ARTICULACIÓN		
Punto	Órganos que participan	Fonemas
Bilabial	Producidos por el contacto entre los labios	/p/, /b/, /m/
Labiodental	Se apoya el labio inferior sobre los dientes incisivos superiores	/f/
Interdental	La lengua se coloca entre los incisivos superiores e inferiores	/z/
Dental	La lengua choca contra los incisivos superiores	/t/, /d/
Alveolar	La lengua toca los alvéolos	/s/, /l/, /r/, /n/
Palatal	La lengua toca el paladar	/ch/, /ll/, /ñ/
Velar	La lengua toca el velo del paladar	/k/, /g/, /j/

Figura 2. Clasificación de los fonemas a través del Punto de Articulación







RASGOS MOTIVADOS POR EL MODO DE ARTICULACIÓN			
Rasgo	Órganos	Fonemas	Representación Rasgo de articulación
Oclusivo	El aire se retiene y se expulsa de golpe	/p/, /b/, /t/, /d/, /k/, /g/	
Fricativo	El aire sale lentamente a través de una pequeña abertura de la boca	/f/, /z/, /j/, /s/	
Africado	El aire se retiene y después se expulsa a través de una pequeña abertura	/ch/	
Lateral	El aire sale por los lados de la boca	/l/, /ll/	
Vibrante	La lengua produce una o más vibraciones	/r/, /rr/	
Nasal	El velo del paladar deja pasar el aire por la cavidad nasal	/m/, /n/, /ñ/	

Figura 3. Clasificación de los fonemas a través del Modo de Articulación

Cabe resaltar que en nuestro proyecto de Tesis solo se basará en el punto de articulación de los fonemas ya que es la parte más visible que se encuentra en la expresión labial, al momento de llevar a cabo la pronunciación de los fonemas.

En el *punto de articulación* participan los principales órganos como son: los labios, la lengua, los dientes, y el paladar, con la finalidad de llevar a cabo la pronunciación de las consonantes y vocales.

Existen 6 puntos de articulación para las consonantes y 5 puntos de articulación para las vocales se basan en los siguientes criterios.

- **Bilabial:** participación de los dos labios, el superior y el inferior.
- **Labiodental:** labio inferior con dientes superiores.
- **Dental:** la lengua toca la parte trasera de los dientes superiores.
- **Alveolar:** la lengua toca los alvéolos superiores.
- **Palatal:** la lengua toca el paladar.
- **Velar:** la lengua toca el velo.

El **punto de articulación para las vocales** se distinguen entre sí por dos factores (ver tablas 8,9,10 y 11) [Poch Olivé, 1999]:

La abertura o distancia que hay entre el paladar y la lengua. La localización o lugar de la boca en el que colocamos la lengua para pronunciarlos [Xandre Robotham, 2011].

Vocal	Abertura	Definición	Localización	Definición
a	Abierta	La lengua se sitúa en la parte baja de la boca [/a/]	Central	La lengua se sitúa en el centro de la boca. La única vocal es la /a/ .
e	Media	La lengua se eleva suavemente hacia el paladar [/e/, /o/]	Anterior	La lengua se acerca hacia los dientes. Son la /e/ y la /i/ .
i	Cerrada	La lengua toca el paladar [/i/, /u/]	Anterior	La lengua se acerca hacia los dientes. Son la /e/ y la /i/ .
o	Media	La lengua se eleva suavemente hacia el paladar [/e/, /o/]	Posterior	La lengua retrocede hacia el velo del paladar. Son la /o/ y la /u/ .
u	Cerrada	La lengua toca el paladar [/i/, /u/]	Posterior	La lengua retrocede hacia el velo del paladar. Son la /o/ y la /u/ .

Tabla 8. Clasificación de las vocales por su punto de articulación.

Grafema	Definición
A – a	Es la vocal que en el español representa el sonido de mayor perceptibilidad. Se pronuncia con los labios más abiertos que en las demás vocales.
B – b	Su articulación es bilabial sonora oclusiva cuando va en posición inicial absoluta o después de nasal (bien – ambos).
C – c	Con la vocales e – i se pronuncia como s. Con Las otras vocales y en cualquier otra posición se pronuncia con articulación velar, oclusiva y sorda (cama – coco – cuna – claro – clima – acto – efecto).
Ch – ch	La articulación es predorsal, prepalatal, africada sorda.
D - d	Su articulación es dental sonora y oclusiva en posición inicial absoluta o precedida de n o l (dado – andar – toldo). En los demás casos es, por lo general, fricativa (modo – piedra – adverbio).
E – e	Se pronuncia elevando un poco el predorso de la lengua hacia la parte anterior del paladar y estirando levemente los labios hacia los lados.
F – f	Se pronuncia con articulación labiodental fricativa sorda .
G – g	Seguida inmediatamente de e – i, su articulación es velar fricativa sorda, como la de la j (genio – gitano – colegio). En los otros casos tiene articulación velar sonora, oclusiva, en posición inicial absoluta o precedida de nasal (gata – gloria – angustia) y por lo general, fricativa, en las demás posiciones (paga – algo – dogma – ignorar). Cuando el sonido velar sonoro precede a una e – i, se interpone, en la escritura, una u, que no se pronuncia (guiso – pague). En los casos en que la u se pronuncia, debe llevar diéresis (cigüeña – pingüino).
H – h	No tiene sonido.
I – i	Se pronuncia elevando hacia la parte anterior del paladar el predorso de la lengua algo más que para articular la e. Se estiran los labios algo más hacia los lados.
J – j	Su articulación es velar sorda y fricativa. Se pronuncia con mayor o menor tensión lo que produce variedad en su pronunciación.
K – k	Tiene articulación velar oclusiva y sorda. Se emplea en voces de procedencia griega o extranjera (kilógramo – kimono).
L – l	Se pronuncia con articulación apicoalveolar, lateral, fricativa y sonora .
Li - ll	Su articulación tradicional es palatal lateral, fricativa y sonora , con contacto más o menos amplio y tenso de la lengua con el paladar.
M – m	Tiene articulación bilabial, nasal, oclusiva y sonora.

Tabla 9. Clasificación de los grafemas A-M y su relación con el punto de articulación.

N – n	Tiene articulación nasal, oclusiva y sonora . Su punto de articulación es alveolar cuando va en principio de palabra o entre vocales (nadie – cadena). Cuando va al final de sílaba seguida de consonante, toma por lo común el punto de articulación de la consonante siguiente. Así se hace bilabial (enmascar – envió); labiodental (enfermo – infiel); interdental (once – encima); dental (antes – donde); alveolar (enlodar); palatal (concha – conllevar) o velar (cinco – engaño).
Ñ – ñ	Tiene articulación dorso prepalatal nasal sonora (ñoño, ñandú)
O – o	Es la más sonora de las vocales después de la a. Se pronuncia emitiendo la voz con los labios un poco hacia fuera en forma redondeada y libre gran parte de la cavidad bucal por retroceso de la lengua, cuyo dorso se eleva hacia el velo del paladar.
P – p	Tiene articulación bilabial, oclusiva y sorda .
Q – q	Representa el mismo sonido de la c ante a – o – u o de la k ante cualquier vocal, por lo tanto es velar , oclusiva y sorda. Cuando va con la vocales e – i se interpone una u, que no suena (queque – quiso).
R – r	Tiene dos sonidos con valor fonológico diferencial: Simple, de una sola vibración apicoalveolar sonora (caro – para – pera). Múltiple , con dos o más vibraciones (carro – parra – perra). Para representar el sonido simple, se emplea una r. El múltiple , se representa con una r a principio de vocablo (rana – risa – ruca) y después de las consonantes l – n – s (alrededor – enredar – israelita); si va intervocálica se duplica la r (tierra – carroza – arruga).
S – s	Representa un fonema fricativo sordo. En América tiene pronunciación predorsal con salida del aire por los dientes (sapo – señal – soldados – pesito).
T – t	Su articulación es dental, oclusiva y sorda (tomate – pastel).
U – u	Es una de las vocales con sonido más débil. Se pronuncia emitiendo la voz con los labios más alargados y fruncidos que para pronunciar la o y con la lengua más retraída y más elevada en su dorso hacia el velo del palada. Es vocal muda en las combinaciones gue – gui; que – qui. Para que suene debe llevar diéresis (vergüenza – cigüeña).
V – v	Actualmente representa el mismo fonema que la b en todos los países de habla hispana. Por lo tanto, su articulación es bilabial , oclusiva y sonora en posición inicial o después de nasal (ventana – envió) y fricativa en los demás casos (arveja – ave).

Tabla 10. Clasificación de los grafemas N-V y su relación con el punto de articulación.

W – w	Es un préstamo lingüístico y se emplea en voces de procedencia extranjera. Es de origen anglosajón. En inglés su articulación es de u semiconsonante (sándwich) En alemán es fricativa, labiodental y sonora (Wagner)
X – x	Representa un sonido doble, compuesto de K o de g sonora (taxi – examen) y de s (axioma – xilófono). Es un sonido velar, fricativo, sordo como el de j, en palabras como México – Ximena - Oaxaca.
Y – y	Representa un sonido palatal, sonoro y generalmente fricativo, de articulación más o menos abierta o cerrada, según el caso. Precedida de nasal se hace africada (cónyuge). Al final de palabra se pronuncia como semivocal (soy – ley – buey). Representa la vocal i entre palabras (este y aquel; peras y manzanas).
Z – z	En Hispanoamérica se articula como una s. La Lengua toma una posición convexa, generalmente predorsal, con salida del aire dental o dentoalveolar.

Tabla 11. Clasificación de los grafemas W-Z y su relación con el punto de articulación.

Se colocó un identificador a cada fonema que corresponde a su punto de articulación. Dicho identificador se eligió en base al primer carácter inicial del nombre del punto de articulación, por ejemplo el fonema /t/ le corresponde el identificador D ya que el punto de articulación de ese fonema es Dental (*ver Tabla 12*).

Identificador	Fonema	Punto de Articulación	Numeración
'C'	/a/	Central	"001"
'B'	/p//b//m/	Bilabial	"002"
'P'	/ch/ /L/ /ñ/ /y/	Palatal	"003"
'D'	/d/ /t/	Dental	"004"
'I'	/e/	Anterior(Inicial)-Media	"005"
'L'	/f/	Labiodental	"006"
'V'	/g/ /k/ /x/ /w/	Velar	"007"
'N'	/i/	Anterior(Inicial)-Cerrada	"008"
'A'	/l/ /n/ /r/ /R/ /s/	Alveolar	"009"
'F'	/o/	Posterior(Final)-Media	"010"
'O'	/u/	Posterior(Final)-Cerrada	"011"

Tabla 12. Elección de los Identificadores correspondientes al punto de articulación.

Continuando con el ejemplo de la palabra “*deshielo*”, por consiguiente la codificación del punto de articulación de la palabra “*deshielo*” es “DIA-PIAF (ver tabla 13).

Reglas del Punto de Articulación			
Ejemplo	Fonema	Identificador	Clasificación
des-yelo	d	D	Dental
	e	I	AnteriorMedia
	s	A	Alveolar
	-	-	-
	y	P	Palatal
	e	I	AnteriorMedia
	l	A	Alveolar
	o	F	PosteriorMedia

Tabla 13. Aplicación de las reglas de punto de articulación a la palabra desyelo.

Capítulo 3

Estado del Arte de la Sincronización Labial

En esta sección se presentan seis trabajos relacionados al estado del Arte de la sincronización labial, estos trabajos relacionados se dividen en cuatro puntos principales: la sincronización labial, la selección de visemas, la creación de vi-sílabas y la comparación de los trabajos relacionados.

3.1 Trabajos relacionados con la Sincronización Labial

Cabiedes, Pelczer y Gamboa [Cabiedes F., et al., 2007] presenta en su trabajo “*Sincronización de Labios: Método Sin Visemas*”. Un método simplificado como solución para el problema de sincronización de labios con archivos audio. El algoritmo recibe como entrada el archivo de audio en formato estandarizado (WAV) y regresa un archivo de texto con valores que indican amplitudes promedios de la información del audio correspondiente a una fracción de segundo predefinido. Para lograr la sincronización del audio con movimientos de la boca, los valores determinados se correlacionan con archivos de animación. La solución adoptada permite la automatización, por una parte, del análisis de un archivo de sonido registrado por cualquier usuario y, por otra, la asociación del archivo audio con una secuencia de archivos de animación (ver *Figura 4*).

Aún cuando los resultados del presente trabajo han sido suficientes para la implementación de un sistema de Avatares parlantes en 2D, quedan por resolver tres problemas derivados del sistema de análisis de audio. El primero: cómo decidir cuando se pueden repetir los pseudovisemas y tiene que ver con el tipo de análisis, dado que el análisis actual no permite diferenciar si dos muestras de audio son idénticas o su diferencia es de calidad y no de cantidad de señal. El segundo problema esta relacionado con cuántos

pseudovisemas necesitamos para tener una variedad que asegure la sensación de naturalidad; y por último como se percibe si hay una falla, en el sentido de tener una boca cerrada en lugar de boca abierta.

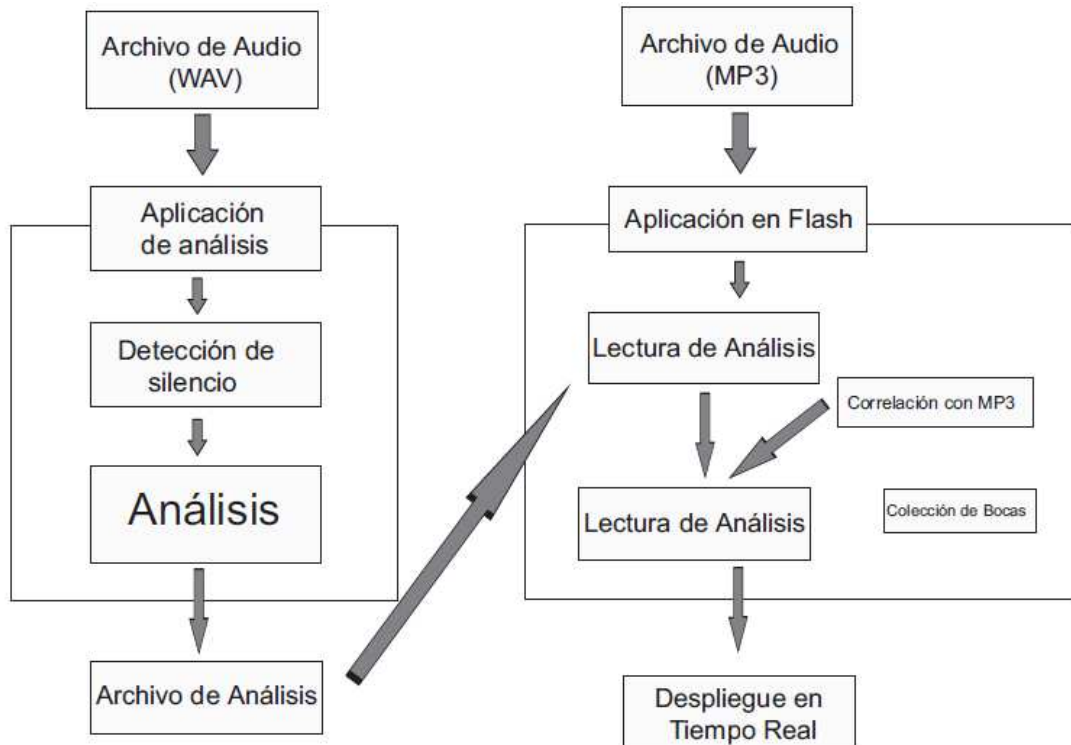


Figura 4. Diagrama del sistema completo. Panel Izquierdo, Unidad de Análisis. Panel derecho Unidad de Despliegue Gráfico.

Zoric´ y Pandzic´ [Zoric´ G. & Pandzic, I. S, 2006] presentan su trabajo “*Real-time language independent lip synchronization method using a genetic algorithm*” un sistema automático de sincronización de labios de avatares en 3D sintéticos basados sólo en una entrada de voz. El enfoque para el sistema de sincronización de labios se realiza mediante el análisis de señal de voz, la voz se clasifica en clases de visemas por medio de redes neuronales. Se utiliza un algoritmo genético para la obtención de un acercamiento óptimo de una topología de red neuronal.

El diseño automático de las redes neuronales con algoritmos genéticos ahorra mucho tiempo en el proceso de entrenamiento de datos (minería de datos). La voz natural siempre implica algunos gestos de la cara, una cara que sólo mueve sus labios, tiene un aspecto muy natural.

Las limitantes de este trabajo es que el sistema automático de la sincronización labial no tiene los aspectos fundamentales como son los gestos faciales y las emociones de un personaje animado y la siguiente limitante es que no aplicaron los procesos lingüísticos como es la fonología y la fonética en la realización de este proyecto.

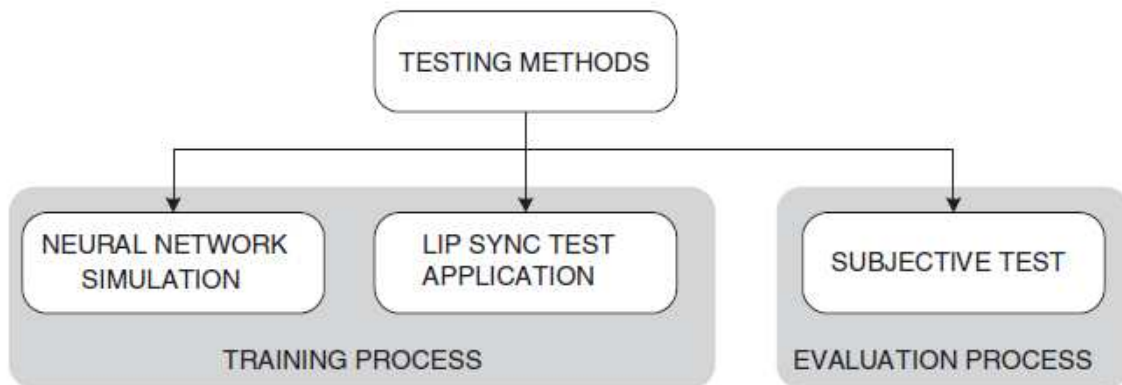


Figura 5. Método usado para la validación del sistema de sincronización labial.

3.2 Trabajos relacionados con la Selección de Visemas

Ferreira & De la Rosa en su trabajo “Una propuesta de integración de animación facial y voz sintética” [Ferreira J., et al., 2003] presentan el análisis, evaluación y estudio de factibilidad de la integración del estándar MPEG-4 con la utilización de herramientas de Text to Speech, que permite la traducción de texto escrito a voz sintética. El estándar MPEG-4 introduce el concepto de FDPs. Estos parámetros pueden ser usados ya sea para modificar la forma y apariencia de un modelo o para codificar la información necesaria en la transmisión de un modelo completo junto con los criterios que deben ser usados para animarlo.

Los FDPs son usados con el fin de cambiar la ubicación relativa de los puntos característicos definidos por el estándar MPEG-4. Los FDP se miden en unidades específicas, Face Animation Parameter Units (FAPU) (ISO/IEC, 1998). En la figura 6 se aprecian las medidas antropométricas normalizadas empleadas en el estándar, los cinco FAPU miden la distancia entre los ojos (ES0), el diámetro del iris (IRISD0), la separación entre los ojos y la nariz (ENS0), la separación entre la boca y la nariz (MNS0) y el ancho de

la boca (MW0). Los parámetros de animación facial, Facial Animation Parameters (FAP), son un conjunto de parámetros que permiten la animación de modelos de cara sintéticos.

El FAPs se basa en el estudio de un mínimo de acciones perceptibles (MPA) y están estrechamente relacionados con acciones musculares. Los 68 parámetros se clasifican en 10 grupos relacionados con las partes de la cara (*ver tabla 14*). FAPs representan un conjunto completo de acciones básicas de movimiento facial incluyendo la cabeza, la lengua, los ojos y el control de la boca. Permiten la representación de expresiones faciales naturales. Para cada FAP, la norma define el apropiado FAPU, el grupo de FAP, la dirección del movimiento positivo y si el movimiento de la FP es unidireccional, en los grupos 2 a 10 se consideran parámetros de bajo nivel.

Número	Grupo	Numero de FAPs
1	Viseme and Expressions	2
2	Jaw, chin, inner lowerlip, cornerlips, midlip	16
3	Eyeballs, pupils, eyelids	12
4	Eyebrow	8
5	Cheeks	4
6	Tongue	5
7	Head rotation	3
8	Outer-lip positions	10
9	Nose	4
10	Ears	4

Tabla 14. Tabla de clasificación de grupos de FAPs.

La traducción de fonemas a FAPs se basa en el siguiente método. Debido a que la animación tiene una velocidad predefinida dada en cuadros por segundo y la información de inicio de los fonemas y su duración está disponible, se realiza una traducción de los fonemas a parámetros FAP de alto nivel, codificando el visema correspondiente en el cuadro de la animación más cercano al tiempo especificado por la información de síntesis.

Debido a que esta aproximación introduce un pequeño desfase en la sincronización de la animación y la voz, se realiza un proceso de interpolación adicional entre dos cuadros que contienen visemas para generar la información intermedia en términos de FAPs de bajo nivel. La traducción de la información de los fonemas, obtenida del motor de traducción de Festival, a FAPs de alto nivel teniendo en cuenta los tiempos en los que los fonemas son reproducidos logra que las salidas de audio y animación gráfica se reproduzcan en sincronía

siempre y cuando los procesos encargados se lancen simultáneamente. A continuación se presenta una tabla con los atributos arrojados por el motor de traducción de Festival y los visemas correspondientes al estándar MPEG-4. (*ver tabla 15*).

VISEMAS MPEG-4		FONEMAS FESTIVAL ESPAÑOL	Ejemplo
0	Ninguno	#	
1	p,b,m	p,b,m	put, bed, mill
2	f,v	F	far, voice
3	T,D	Ninguno	think, that
4	t,d	t,d	Tip, doll
5	k,g	k,g	call, gas
6	tS, dZ, S	x, ch, ll	char, join, she
7	s,z	s, th (z)	Sir, zeal
8	n,l	n, ny, (ñ), l	Lot not
9	R	r, rr	Red
10	A	a, al	Car
11	E	e, el	bed
12	I	i, iO, il	Tip
13	Q	o, ol,	top
14	U	u, uO, ul	book

Tabla 15. Visemas definidos por MPEG-4 y los fonemas generados por Festival.

Este trabajo deja pendiente la generación de animaciones resolviendo la problemática de sincronización de labios con las voces.

Ceballos, Bernardo Gómez y Prieto, presentaron el trabajo “Seguimiento del contorno externo de la boca en imágenes de video” [Ceballos, A., et al., 2009]. El objetivo de esta investigación es proponer un algoritmo para el seguimiento del contorno externo de la boca; este trabajo está basado en la apariencia y en restricciones morfológicas definidas en el estándar MPEG-4, utilizando píxeles para el seguimiento del contorno externo de la boca en imágenes a color. Para la extracción de las características de la región de la boca existen dos enfoques clásicos. El primero estaba basado en la apariencia (bajo nivel) en el cual se realizan operaciones matemáticas manualmente, sin encontrar la forma exacta de la boca.

Con la finalidad de permitir la animación de rostros, en el estándar MPEG-4 se presentan dos conjuntos de parámetros que estandarizan los modelos del rostro respecto a

algunas medidas antropométricas y definen su deformación. Los parámetros de animación facial, Facial Animation Parameters (FAP), son un conjunto de parámetros que permiten la animación de modelos de cara sintéticos.

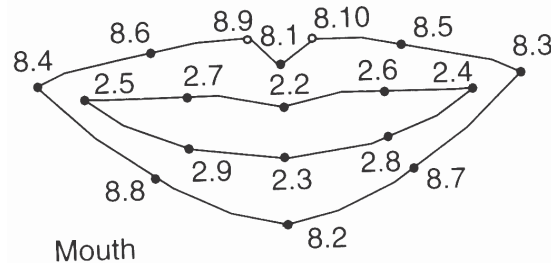


Figura 6. Parámetros de Definición facial de la boca.

En el reconocimiento del habla, generalmente se usan los grupos 2 y 8, que describen el movimiento del contorno interno y externo de la boca, respectivamente, mientras que para la síntesis visual del habla se usa el grupo 1, que define 14 visemas correspondientes del idioma inglés. (ver Tabla 15).

El algoritmo usa el grupo 8 que describe el contorno externo de los labios (ver Figura 6), pues algunos estudios psicológicos han sugerido que es el que más influencia tiene en la lectura de los labios. Se presenta a continuación el algoritmo seguimiento asistido de puntos del contorno externo de los labios (ver Tabla 16)

Algoritmo 1: Seguimiento asistido de puntos del contorno externo
Entradas: Video en forma de secuencia de imágenes $C_1, C_2, \dots, C_n \in C$
Salidas: La secuencia de puntos del contorno externo de la boca para todos los cuadros de video S_{n-10}^n
Los siguientes pasos se realizan para todos los cuadros
[Paso 1:] Localización de la región de interés
[Paso 2:] Ubicación de los 10 puntos del contorno externo de la boca para el primer cuadro de video. $p_1, p_2, p_{10} \in P$

<p>[Paso 3:] Para todos los puntos del contorno del cuadro anterior S_{n-1} hacer</p> <p>Calcular la similitud con los píxeles de la vecindad en el cuadro presente.</p> <p>Escoger el candidato a punto actual S_{n-i} como el píxel donde la similitud es mayor (más cercana a la unidad).</p> <p>Fin para</p> <p>Una vez obtenidos los 10 puntos candidatos, aplicar restricciones de forma</p>
<p>[Paso 4:] Cálculo de las características de la región de la boca</p>

Tabla 16. Algoritmo1: Seguimiento asistido de punto del contorno externo.

Su principal limitante es que no se ha logrado establecer una metodología adecuada para realizar seguimiento preciso de la forma de los labios en tiempo real.

En el 2007 Melenchón presentó su tesis doctoral denominada “Síntesis Audiovisual Realista Personalizable” [Melenchon, Maldonado, J., 2007]. Este trabajo se desarrolla a partir de una secuencia de video de entrenamiento con una persona pronunciando un conjunto equilibrado de visemas y mostrando diferentes expresiones y gestos. El conjunto de visemas se determina mediante el uso de algoritmos genéticos paralelos y el modelo facial se obtiene automáticamente siguiendo el algoritmo detallado que el mismo presento en el año 2005.

La propuesta de Melenchón utiliza un modelo basado en imágenes bidimensionales en las que se muestra un esquema de cómputo incremental. Cabe resaltar que la principal limitante de este trabajo es que la información de los alófonos y visemas solo fue realizada para el idioma español de España.

3.3 Trabajos relacionados con la creación de vi-sílabas

Kshirsagar [S. Kshirsagar, Nadia, 2003] presenta su trabajo “*Visyllable Based Speech Animation*”, el cual tiene por objetivo la implementación de *vi-sílabas* para la animación de voz. En este trabajo se divide principalmente en dos fases. En la primera fase

se definió un conjunto de unidades *vi-sílabas* (es decir, la representación de una sílaba como un visema) del idioma inglés de forma arbitraria, creando las posibles sílabas recorriendo el punto de división de la palabra (ver ejemplo x-ray). En la segunda fase se realizó una selección de sílabas que pudieran ser válidas, con la finalidad de que estas sílabas fueran verificadas para su aprobación sin la identificación de fundamentos lingüísticos como son la fonología y la fonética. Con base en estas reglas, Kshirsagar implementó un algoritmo de Silabificación que permite la segmentación de una cadena de fonemas en sílabas (ver *Figura 7*).

Kshirsagar presenta un ejemplo de cómo se realiza la división de las sílabas. Por ejemplo, la palabra "x-ray" se puede dividir en dos sílabas de las siguientes maneras:

- 1.e-ksre
- 2.ek-sre
- 3.eks-re
4. eksr-e

La principal limitante de este trabajo es que carece de fundamentos lingüísticos como son la fonología y la fonética en su proceso de silabificación, ya que este algoritmo no realiza de forma válida el proceso de división de sílabas (silabificación).

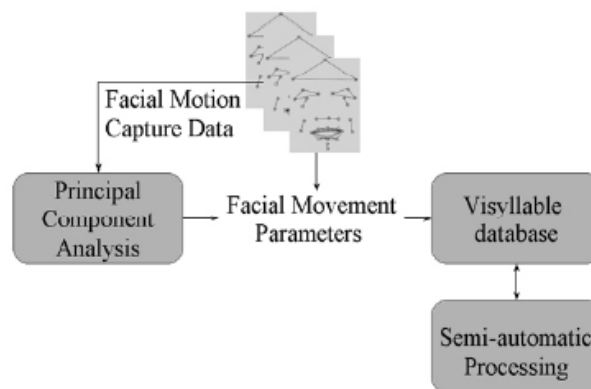


Figura 7. Animación de voz basada en *vi-sílabas* en tiempo real.

3.4 Comparación de Trabajos relacionados

En esta sección se presenta una tabla comparativa (ver tabla 17) que resume las características de los trabajos presentados en la *secciones 3.1, 3.2 y 3.3*, con la finalidad de comprender las principales limitantes y características de cada uno de los trabajos presentados en las diferentes secciones, que permitieron llevar a cabo la realización de este proyecto de Tesis. Los atributos de la tabla son: el tipo de modelo gráfico que se trabajó, las características del algoritmo, componentes involucrados en el proceso, si trabajan o consideran la prosodia, el tipo de información de entrada, el idioma con el que se trabaja y el formato de la salida gráfica que utiliza cada proyecto de investigación.

El trabajo que presenta Cabiedes, Pelczer y Gamboa [Cabiedes F., et al., 2007] tiene su principal limitante en que se desconoce cuando se deben repetir los pseudovisemas, cuántos se necesitan para tener una variedad que asegure la sensación de naturalidad; y como se percibe si hay una falla, en el sentido de tener una boca cerrada en lugar de boca abierta, también se intentó realizar la sincronización de labios entre el visema con la utilización de una herramienta de Text to Speech.

El segundo trabajo denominado “*Real-time language independent lip synchronization method using a genetic algorithm*” [Zoric´ G. & Pandzic, I. S., 2006] No presentan aplicación de la prosodia, reglas lingüísticas y se pretenden que el algoritmo tome los gestos faciales, expresiones de la cara y los movimientos de los labios de los avatares en 3D. Nuestro proceso de selección de sílabas aplica las reglas de silabeo para dividir una palabra en sílabas y se realizó la sincronización de una sílaba con un visema.

Los trabajos de [Ferreira, J. F., et al., 2003], [Ceballos, A., et al., 2009] y [Melenchon, Maldonado, J., 2007] utilizan MPEG-4 y FAP’s para la generación de animación en la sincronización de labios con las voces y se avocan a realizar seguimiento preciso de la forma de los labios en tiempo real. Nosotros, realizamos la sincronización de los visemas como texturas animadas en Panda3D, resultado de una sincronización de los fonemas generados por una herramienta TTS y proceso de silabeo y fonemización que identifica los fonemas relevantes.

Kshirsagar [Kshirsagar, Nadia, S., 2003], aborda el concepto de Vi-sílabas, sin embargo, su principal limitante es que carece de fundamentos lingüísticos como son la fonología y la fonética en su proceso de Silabificación. En esta tesis, se aborda los procesos de fonemización y silabeo basado en el trabajo de Ríos Mestre [Mestre, Ríos. A., 1999] para dividir una palabra en sílabas; y posteriormente representar esa sílaba en un visema.

Trabajos Relacionados	Modelo	Características	Proceso	Prosodia	Entrada	Idioma	Salida
[Cabiedes F., et al., 2007]	2D	Algoritmo para asociar movimientos de labios	Estándar MPEG-4	No se utiliza	Archivo de audio	Español (México)	Animación en flash
[Zoric´ G., e Pandzic´, I. S., 2006]	3D	Algoritmo Genético	Modelos Ocultos de Markov	No se utiliza	Archivo de audio	Inglés	Animación en MPEG-4
[Ferreira J. F & De la Rosa, 2003]	2D	Paramétrico	Voz sintéticas, Estándar MPEG-4	TTS, Festival	Texto Escrito	Español (Colombia)	Animación en MPEG-4
[Ceballos, A. et al., 2009]	2D	Paramétrico	Definición facial de Parámetros MPEG-4	No se utiliza	Formato de video	Español (Colombia)	Animación en MPEG-4
[Melenchon. Maldonado, J., 2007]	2D y 3D	Análisis de componentes Principales	Modelos Ocultos de Markov	Alofonos, Visemas	Formato de video	Español (España)	Animación en MPEG-4
[Kshirsagar, Nadia, S., 2003]	2D	Algoritmo de Silabificación	Definición facial de Parámetros, sílabas	A medias Fonemas, visemas	Cadena de fonemas	Inglés	FMP Estándar MPEG-4
[Radilla Ávila J., 2012]	3D	Aplicación de algoritmos K-Means y C4.5 (Librería Orange)	Implementación de algoritmo de lingüísticas	Ramas de la lingüísticas, sílabas, visema	Cadena de grafemas	Español (México)	Animación en Panda3D

Tabla 17. Tabla comparativa de los trabajos relacionados.

Capítulo 4

Arquitectura de Silabeo Automático para la generación de Vi-sílabas en español

En este capítulo se presenta una arquitectura para la generación de vi-sílabas basada en la automatización de silabeo de frases escritas en el idioma español aplicando fundamentos lingüísticos en el proceso de Fonemización.

4.1 Arquitectura General

Esta propuesta se compone de tres procesos principales: 1) generación de Silabeo Automático, 2) la Transcripción por Punto de articulación y 3) Selección de Vi-sílabas para así mostrar una correcta sincronización de los Visemas (ver Figura 8).

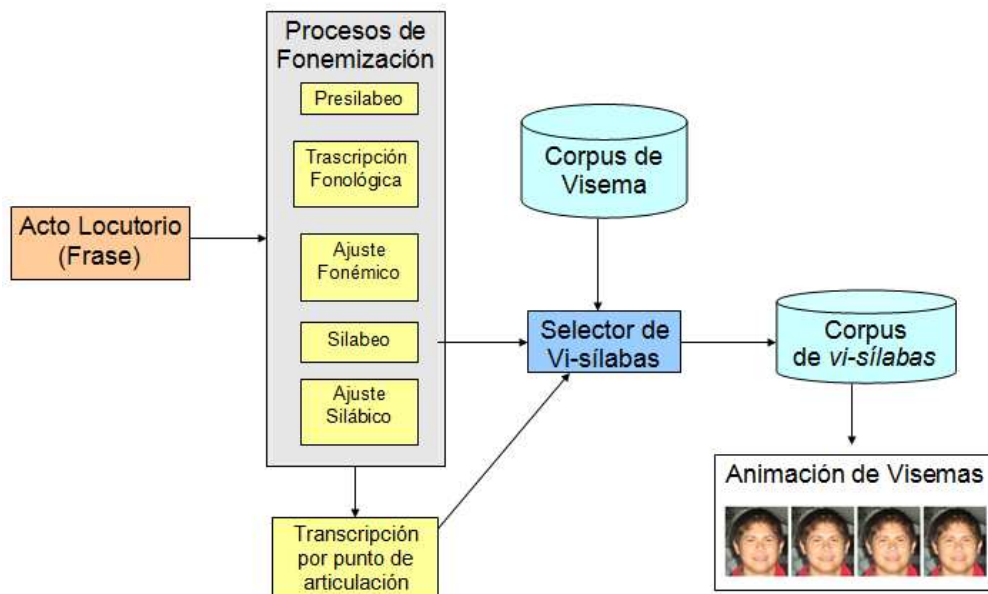


Figura 8. Arquitectura General para la generación de Vi-sílabas.

4.2 Modelo de Silabeo Automático

El modelo de silabeo automático es el proceso que permite dividir una palabra en sílabas, a partir de procesos de presilabeo, proceso de transcripción de grafema a fonemas y el proceso de ajuste silábico. Estos procesos están basados en la implementación de las reglas de las lingüísticas, como son la fonología y la fonética y se encuentran incluidos en el documento de Ríos Mestre [Mestre, Ríos, 1999].

La Figura 9 muestra el detalle del proceso de Silabeo Automático, en donde se presentan las reglas que representan los procesos de fonemización, las cuales se aplican a la entrada de una palabra/frase y cuya salida será la cadena ya silabificada.

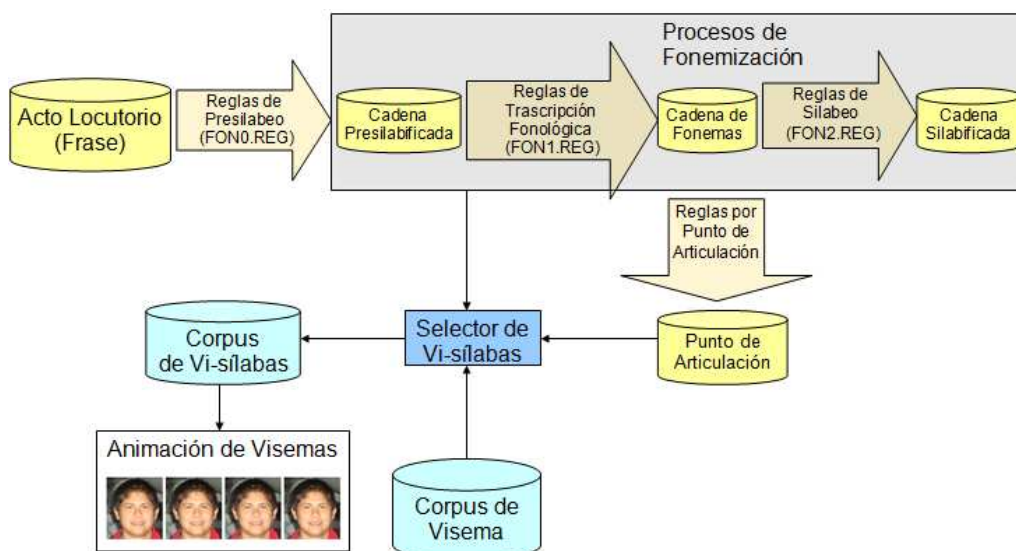


Figura 9. Modelo de Silabeo Automático para la generación de Vi-sílabas.

El proceso está compuesto de 3 tipos de reglas. Las reglas de Presilabeo FON0.REG tienen como objetivo colocar una marca de división silábica a la cadena de entrada de un acto locutorio ó palabra, esta cadena presilabificada servirá como entrada al proceso de transcripción fonológica.

La aplicación de los procesos de Fonemización comienzan con el conjunto de reglas llamado FON1.REG que realiza la transcripción fonológica, en donde convertirá el grafema (letras) a fonemas (*ver Figura 9*).

Cabe mencionar que en el proceso de transcripción fonológica se realizaron algunos ajustes fonológicos a ciertas palabras como por ejemplo: México, Texas, muchacho, zapato, ralph entre otras palabras, con el objetivo de realizar la adecuada transcripción fonológica a los fonemas del español México, por ejemplo los grafemas ch, w, x, z, ph; al realizar la transcripción fonológica no eran los fonemas correctos para el idioma español de México. Se añadieron más reglas para solucionar los problemas que anteriormente se mencionaron ya que este trabajo de tesis sólo nos basamos en las reglas propuestas por Ríos Mestre ya que estas reglas de las lingüísticas sólo fueron aplicadas para el idioma de español de España.

Las siguientes reglas del proceso de Fonemización son las reglas de silabeo FON2.REG, estas reglas lo que hacen es insertar una marca de división silábica a los fonemas ya procesados (ver sección 2.3.3). Se tendrá la palabra ya silabificada, que pasará a un proceso de ajuste silábico para determinar si la palabra es originaria de los fonemas del español México. Por ejemplo las palabras como angstrom, postdata se debe de omitir fonemas aplicando el ajuste silábico quedaría de la siguiente manera /án-g-s-trom/ → /ans-trom/, /pos-t-da-ta/ → /pos-da-ta/ (ver sección 2.3.4).

4.3 Transcripción por Punto de Articulación

Una vez obtenida la cadena silabificada y fonemizada, el siguiente paso es convertir los fonemas procesados al punto de articulación equivalente. Esto con la finalidad de saber a que punto de articulación corresponde cada elemento de la cadena silabificada y fonemizada. Los puntos de articulación son los que a continuación se mencionan: Central, Bilabial, Palatal, Dental, Anterior-Media, Labiodental, Velar, Anterior-Cerrada, Alveolar, Posterior-Media y Posterior-Cerrada. Cabe resaltar que el punto de articulación es el lugar en la boca donde se pronuncia un determinado fonema.

Por ejemplo en la palabra *deshielo* el punto de articulación será (DIA-PIAF) (ver sección 2.3.5, tabla 13).

4.4 Modelo de Selección de Vi-sílabas

La siguiente etapa de nuestro proyecto de Tesis, es el proceso de selección de vi-sílabas, es decir la asociación de visemas a una sílaba válida (vi-sílabas). Para realizarlo se obtiene el/los puntos de articulación relevante(s) de cada sílaba y se les asocian los visemas que los representan. Contando con la información de transformación a punto de articulación de una palabra, el siguiente paso es seleccionar las sílabas válidas a través del uso de librerías de Orange para Python y los visemas adecuados para que estos sean mostrados en la consola de juegos de Panda3D.

Se revisaron 3 casos posibles:

El primer caso de prueba tiene como finalidad asociar un visema a cada fonema que se encuentran en una sílaba, es decir animar cada fonema de la sílaba, por ejemplo si se desea animar la palabra “*casa*” serían cuatro visemas lo que se tendría que representar, esto fue una problemática que identificó Keith Lango [Lango, K., 2006], las letras no son sonidos. Los sonidos no son letras, no hay letras en la animación de sincronización labial.

El segundo caso de prueba implementado tiene por objetivo animar el fonema inicial y el fonema final de una sílaba, esto nos permitirá distinguir los dos fonemas visuales (visemas) más representativos que se encuentran constituidos en una sílaba.

En el tercer caso de prueba usa los resultados de un proceso de agrupamiento utilizando el algoritmo K-Means con la finalidad de obtener 11 grupos ó centroides que agruparían a una sílaba, cada grupo se considerará un punto de articulación. Se utilizó una base de datos de entrenamientos con ejemplos de palabras que son originarias del idioma español de México. Se realizaron varios análisis para identificar el fonema /punto de articulación que corresponde al grupo arrojado por el K-Means. Teniendo la información del grupo a que correspondía cada fonema, se utiliza el clasificador C4.5 para seleccionar los visemas que se seleccionarán para una nueva palabra o frase. Se presenta en la figura 10 el modelo de selección de vi-sílabas.

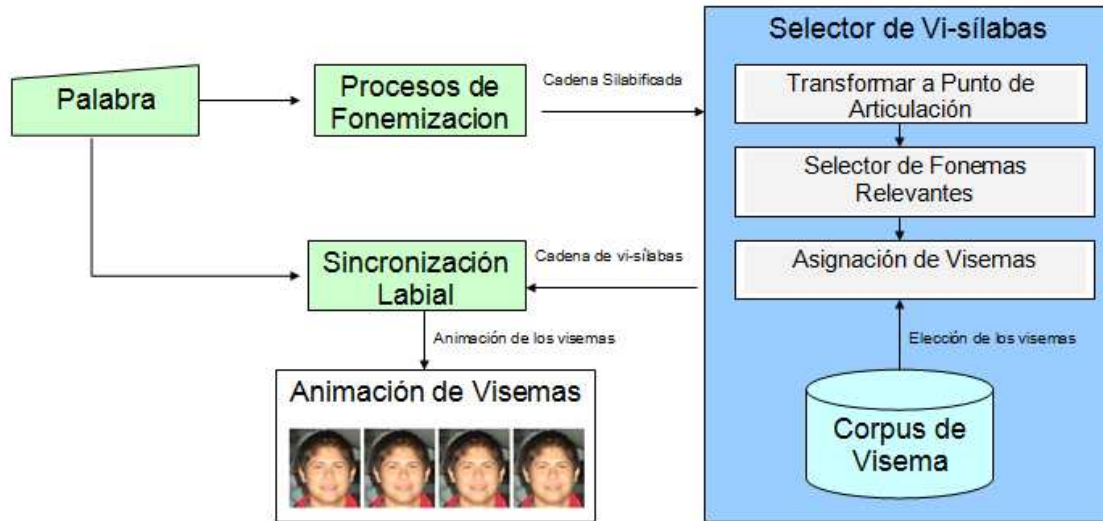


Figura 10. Modelo de Selección de Vi-sílabas.

4.5 Modelo de Sincronización Labial

La Figura 11 presenta el esquema del proceso de sincronización de los visemas seleccionados por el proceso de selección de vi-sílabas (ver sección 4.4) con la voz de la palabra/frase que se quiere comunicar. Cada palabra o frase que pasa por los procesos de fonemización (ver sección 4.2) también pasa por un proceso de Text-to-Speech (motor de sintetizador de voz) que la leerá y emitirá la voz y los fonemas asociados.

Se utilizó la librería para Python denominada PyTTS y la Voz de Juan del Sintetizador de Voz llamado Loquendo. La sincronización en tiempo real se realiza cada vez que la librería emite un fonema. Si el fonema que se está procesando se encuentra seleccionado por el selector de vi-sílabas, entonces el visema que representa a la vi-sílaba será mostrado, generándose la secuencia de animaciones.

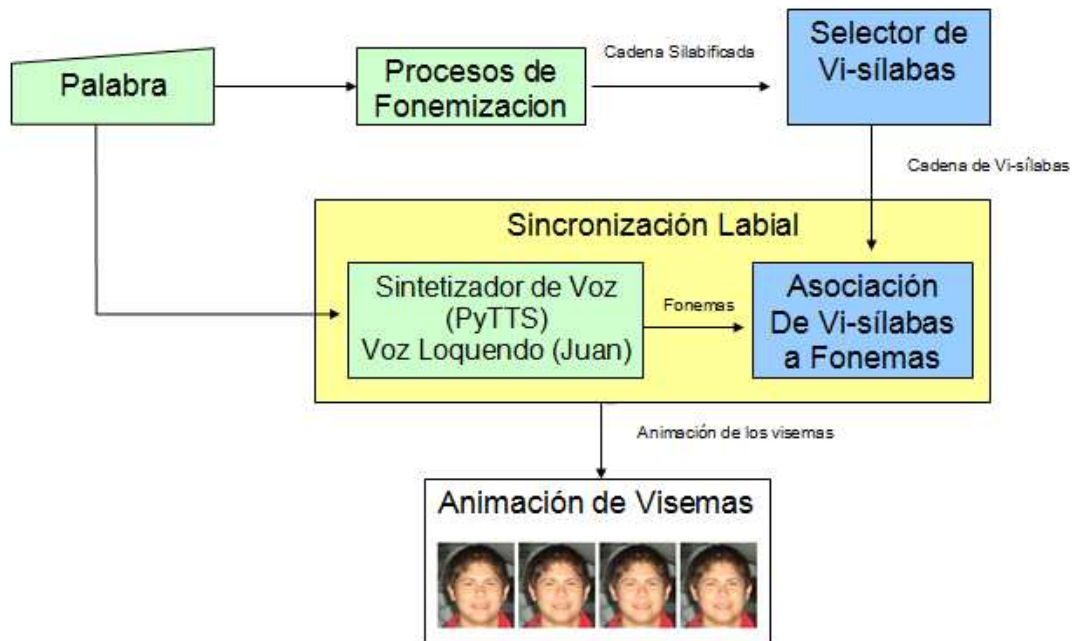


Figura 11. Modelo de Sincronización Labial en tiempo real.

Capítulo 5

Evaluación y Análisis de Resultados

En este capítulo se describen dos tipos de evaluaciones y análisis de resultados que se llevaron a cabo en el desarrollo de esta tesis. El primero es relativo a la asociación de un único visema a una sílaba, y el segundo a la percepción de naturalidad en las tres variantes de sincronización labial que se propusieron.

5.1 Asignación de un punto de Articulación a una sílaba

En esta sección se describe el análisis de frecuencia realizado a los resultados obtenidos de aplicar el método del aprendizaje no supervisado K-Means a una base de datos de sílabas para agruparlas en 11 clases que representen posibles puntos de articulación. Las sílabas de la base de datos de entrenamiento están compuestas de uno a seis fonemas (mismos que fueron relacionados al punto de articulación de cada fonema).

Según el análisis de frecuencia realizado, el grupo C1 (*ver Tabla 18*), representa en su mayoría al punto de articulación Posterior-Media, ya que éste punto de articulación obtuvo el mayor porcentaje de repetición en el grupo: 57.35 %.

GRUPO C1 ORANGE K-MEANS			
N° Elementos 1686			
Punto Articulación	Números	=	Porcentaje
CENTRAL	274	=	16.25%
BILABIAL	239	=	14.17%
PALATAL	38	=	2.25%
DENTAL	194	=	11.50%
ANTERIOR-MEDIA	191	=	11.32%
LABIODENTAL	28	=	1.66 %
VELAR	780	=	46.26%
ANTERIOR-CERRADA	237	=	14.05%
ALVEOLAR	762	=	45.19%
POSTERIOR-MEDIA	967	=	57.35%
POSTERIOR-CERRADA	109	=	6.46%

Tabla 18. Análisis de frecuencia por Punto de Articulación para el grupo C1.

Siguiendo el mismo análisis, al grupo C2 le corresponde el punto de articulación Anterior-Cerrada con el porcentaje de 69.36 %.

GRUPO C2 ORANGE K-MEANS			
N° Elementos 1110			
Punto Articulación	Números	=	Porcentaje
CENTRAL	111	=	10 %
BILABIAL	240	=	21.62%
PALATAL	6	=	0.54%
DENTAL	150	=	13.51%
ANTERIOR-MEDIA	19	=	1.71%
LABIODENTAL	27	=	2.43%
VELAR	15	=	1.35%
ANTERIOR-CERRADA	770	=	69.36%
ALVEOLAR	757	=	68.19%
POSTERIOR-MEDIA	104	=	9.36%
POSTERIOR-CERRADA	98	=	8.82%

Tabla 19. Análisis de frecuencia por Punto de Articulación para el grupo C2.

Según el análisis de las frecuencias de los ejemplos agrupados por el K-Means en el grupo C3, a éste le corresponde el punto de articulación Alveolar, ya que cuenta con el 99.59 %.

GRUPO C3 ORANGE K-MEANS			
N° Elementos 489			
Punto Articulación	Números	=	Porcentaje
CENTRAL	381	=	77.91%
BILABIAL	91	=	18.60 %
PALATAL	1	=	0.20%
DENTAL	102	=	20.85%
ANTERIOR-MEDIA	46	=	9.40%
LABIODENTAL	21	=	4.29%
VELAR	50	=	10.22%
ANTERIOR-CERRADA	388	=	79.34%
ALVEOLAR	487	=	99.59%
POSTERIOR-MEDIA	40	=	8.17%
POSTERIOR-CERRADA	73	=	14.92%

Tabla 20. Análisis de frecuencia por Punto de Articulación para el grupo C3.

Los resultados del análisis de frecuencia del grupo C4 muestran el porcentaje de 96.29 % asignado al punto de articulación Alveolar, y para el grupo C5 el punto de articulación es Anterior-Media (ver Tabla 21 y 22).

GRUPO C4 ORANGE K-MEANS			
N° Elementos 378			
Punto Articulación	Números	=	Porcentaje
CENTRAL	10	=	2.64 %
BILABIAL	87	=	23.01%
PALATAL	14	=	3.70%
DENTAL	72	=	19.04%
ANTERIOR-MEDIA	288	=	76.19%
LABIODENTAL	5	=	1.32%
VELAR	59	=	15.60%
ANTERIOR-CERRADA	16	=	4.23%
ALVEOLAR	364	=	96.29%
POSTERIOR-MEDIA	32	=	8.46%
POSTERIOR-CERRADA	45	=	11.90%

Tabla 21. Análisis de frecuencia por Punto de Articulación para el grupo C4.

GRUPO C5 ORANGE K-MEANS			
N° Elementos 419			
Punto Articulación	Números	=	Porcentaje
CENTRAL	0	=	0.0 %
BILABIAL	91	=	21.71%
PALATAL	43	=	10.26%
DENTAL	0	=	0.0%
ANTERIOR-MEDIA	408	=	97.37%
LABIODENTAL	1	=	0.23%
VELAR	0	=	0.0%
ANTERIOR-CERRADA	0	=	0.0%
ALVEOLAR	272	=	64.91%
POSTERIOR-MEDIA	0	=	0.0%
POSTERIOR-CERRADA	1	=	0.23%

Tabla 22. Análisis de frecuencia por Punto de Articulación para el grupo C5.

Siguiendo con los análisis de frecuencia, al grupo C6 le corresponde el punto de articulación Central con un porcentaje de 99.61%. El grupo C7 corresponde al punto de articulación Velar ya que cuenta con un porcentaje del 100%.

GRUPO C6 ORANGE K-MEANS			
N° Elementos 260			
Punto Articulación	Números	=	Porcentaje
CENTRAL	259	=	99.61%
BILABIAL	46	=	17.69%
PALATAL	13	=	5%
DENTAL	66	=	25.38%
ANTERIOR-MEDIA	0	=	0.0%
LABIODENTAL	7	=	2.69%
VELAR	48	=	18.46%
ANTERIOR-CERRADA	5	=	1.92%
ALVEOLAR	234	=	90.00%
POSTERIOR-MEDIA	0	=	0.0%
POSTERIOR-CERRADA	0	=	0.0%

Tabla 23. Análisis de frecuencia por Punto de Articulación para el grupo C6.

GRUPO C7 ORANGE K-MEANS			
N° Elementos 82			
Punto Articulación	Números	=	Porcentaje
CENTRAL	0	=	0.0%
BILABIAL	0	=	0.0%
PALATAL	0	=	0.0%
DENTAL	2	=	2.43%
ANTERIOR-MEDIA	82	=	100.00%
LABIODENTAL	0	=	0.0%
VELAR	82	=	100.00%
ANTERIOR-CERRADA	1	=	1.21%
ALVEOLAR	0	=	0.0%
POSTERIOR-MEDIA	0	=	0.0%
POSTERIOR-CERRADA	0	=	0.0%

Tabla 24. Análisis de frecuencia por Punto de Articulación para el grupo C7.

El punto de articulación más representativo para el grupo C8 es Dental con un 93.83 %. Prosiguiendo con los resultados, el grupo C9, representa en su mayoría al punto de articulación Alveolar con un porcentaje de 94.02 %. Cabe resaltar que los grupos C3 y C4, también representan en su mayoría a este punto de articulación.

GRUPO C8 ORANGE K-MEANS			
N° Elementos 389			
Punto Articulación	Números	=	Porcentaje
CENTRAL	113	=	29.04%
BILABIAL	6	=	1.54%
PALATAL	0	=	0.0%
DENTAL	365	=	93.83%
ANTERIOR-MEDIA	166	=	42.67%
LABIODENTAL	0	=	0.0%
VELAR	17	=	4.37%
ANTERIOR-CERRADA	25	=	6.42%
ALVEOLAR	65	=	16.70%
POSTERIOR-MEDIA	33	=	8.48%
POSTERIOR-CERRADA	62	=	15.93%

Tabla 25. Análisis de frecuencia por Punto de Articulación para el grupo C8.

Prosiguiendo con los resultados del análisis de frecuencia para el grupo C9 éste grupo representa en su mayoría el punto de articulación Alveolar con un porcentaje de

94.02 %. Cabe resaltar que los grupos C3 y C4, el punto de articulación es Alveolar.

Esto se debe a que los datos que fueron evaluados a partir de una base de datos de entrenamiento, no se encontraron con más ejemplos para determinar que los grupos C3 y C4 deberían de tomar otro punto de articulación que quedaron sin agrupación (*ver figura 12*) por mencionar los puntos de articulación pendiente como Bilabial, Palatal, Labiodental y Posterior-Cerrada. Es por ello que el punto de articulación Alveolar se asignó a tres grupos C3, C4 y C9.

GRUPO C9 ORANGE K-MEANS			
N° Elementos 201			
Punto Articulación	Números	=	Porcentaje
CENTRAL	39	=	19.40%
BILABIAL	130	=	64.67%
PALATAL	4	=	1.99%
DENTAL	18	=	8.95%
ANTERIOR-MEDIA	62	=	37.81%
LABIODENTAL	44	=	21.89%
VELAR	20	=	9.95%
ANTERIOR-CERRADA	52	=	25.87%
ALVEOLAR	189	=	94.02%
POSTERIOR-MEDIA	33	=	16.41%
POSTERIOR-CERRADA	66	=	32.83%

Tabla 26. Análisis de frecuencia por Punto de Articulación para el grupo C9.

La tabla 27 muestra los resultados del análisis de frecuencias del grupo C10 y corresponde al punto de articulación Central con un porcentaje del 60.0 %. Cabe resaltar que los grupos C6 y C11, el punto de articulación también es Central. Prosiguiendo, para el grupo C11 el punto de articulación es Central con un porcentaje del 60.0 %.

GRUPO C10 ORANGE K-MEANS			
N° Elementos 552			
Punto Articulación	Números	=	Porcentaje
CENTRAL	334	=	60%
BILABIAL	7	=	1%
PALATAL	1	=	0.1%
DENTAL	11	=	1.98%
ANTERIOR-MEDIA	49	=	8.86%
LABIODENTAL	1	=	0.1%
VELAR	10	=	1.80%
ANTERIOR-CERRADA	83	=	15%
ALVEOLAR	80	=	14.49%
POSTERIOR-MEDIA	65	=	15%
POSTERIOR-CERRADA	33	=	5.96%

Tabla 27. Análisis de frecuencia por Punto de Articulación para el grupo C10.

GRUPO C11 ORANGE K-MEANS			
N° Elementos 246			
Punto Articulación	Números	=	Porcentaje
CENTRAL	216	=	87.80%
BILABIAL	196	=	79.67%
PALATAL	39	=	15.85%
DENTAL	1	=	0.40%
ANTERIOR-MEDIA	0	=	0.0%
LABIODENTAL	12	=	4.87%
VELAR	7	=	1.62%
ANTERIOR-CERRADA	5	=	2.03%
ALVEOLAR	1	=	0.40%
POSTERIOR-MEDIA	0	=	0.0%
POSTERIOR-CERRADA	30	=	12.19%

Tabla 28. Análisis de frecuencia por Punto de Articulación para el grupo C11.

Como se ha observado en el análisis de frecuencia de los grupos que se asignaron a los once puntos de articulación, siete grupos fueron asignados a un punto de articulación sin ningún problema. La base de datos de entrenamiento se conformo de sílabas de palabras que fueron colocadas al azar siguiendo en algunos casos el orden alfabético de los fonemas que la conforman.

Se observó en el análisis que cuatro grupos los cuales son C4, C9, C10 y C11, no fueron asignados a los puntos de articulación ya asignados y ningún grupo representó a los puntos de articulación: Bilabial, Palatal, Labiodental y Posterior-Cerrada, esto se debió a que la base de datos de entrenamiento no se encontraron con palabras en donde el punto de articulación fueran los antes mencionados. A continuación se muestra en la Figura 12 como quedó la asignación de los grupos y los porcentajes que estos tuvieron.

Punto de Articulación										
Central	Bilabial	Palatal	Dental	Anterior-Media	Labiodental	Velar	Anterior-Cerrada	Alveolar	Posterior-Media	Posterior-Cerrada
C6- 99.61%			C8- 93.83%	C5- 97.37%		C7 100%	C2- 69.36%	C3 - 99.59%	C1- 57.35%	
C10 60%								C4- 96.29%		
C11 87.80%								C9- 93.53%		

Figura 12. Asignación de los once grupos ó centroides a cada punto de articulación.

5.2 Selección de Visemas que se asociarán a una sílaba

En este apartado se realizó el análisis de tres casos de prueba, para determinar que propuesta de asociación de la imagen de un punto de articulación (visema) a una sílaba es la que permite que la sincronización labial sea más natural y creíble. A continuación se describen cada uno de los casos y los resultados obtenidos.

El primer caso de prueba tiene como finalidad animar cada fonema que se encuentran en una sílaba, (*ver Figura 13*).

Palabra	Cadena Silabificada	Fonemas	Visema
escuela	es	e	
		s	
	kue	k	
		u	
		e	
	la	l	
		a	

Figura 13. Primer Caso de Prueba: cada fonema de la palabra tiene un visema asociado.

El segundo caso de prueba tiene por objetivo crear una animación considerando el fonema inicial y el fonema final de cada sílaba (ver Figura 14).

Palabra	Cadena Silabificada	Fonemas	Visema
escuela	es	e	
		s	
	kue	k	
		e	
	la	l	
		a	

Figura 14. Segundo Caso de Prueba: se anima el primer y último fonema de una sílaba.

En el tercer caso de prueba consiste en asignar un solo visema a una sílaba. El visema seleccionado es el resultado de aplicar el algoritmo de C4.5 al nuevo ejemplo que se quiere procesar. Este algoritmo tiene como base de entrenamiento los resultados del proceso de Agrupación (K-Means) descritos en la sección 5.1.




Palabra	Cadena Silabificada	Fonemas (Según el clasificador c45)	Visema
escuela	es	s	
	kue	e	
	la	a	

Figura 15. Tercer Caso Prueba. Se realiza la elección del fonema-punto de articulación que mejor represente a una sílaba.

Con el propósito de elegir que clasificador era el adecuado para la clasificación de los datos, proporcionados por el proceso de Agrupación (K-Means). Se evaluaron 4 algoritmos de aprendizaje supervisado que se encuentran disponibles en Orange [Curk, T., et al., 2005] Bayes, Knn, C4.5 e ID3.

El algoritmo de aprendizaje Bayesiano se puede ver como el proceso de encontrar la hipótesis más probable, dado un conjunto de ejemplos de entrenamiento D y un conocimiento a priori sobre la probabilidad de cada hipótesis [Curk, T., et al., 2005].

El clasificador Knn es la regla de clasificación por vecindad más general es conocido como la clasificación de los K-Vecinos mas cercanos o simplemente K-NN. Se basa en la suposición de que los prototipos más cercanos tienen una probabilidad a posteriori similar [Curk, T., et al., 2005].

El algoritmo C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente. El árbol se construye mediante la estrategia de profundidad-primero (depth-first). El algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo [Curk, T., et al., 2005].

Otro tipo de clasificador de árbol es el ID3, están formados por dos tipos de nodos: los nodos de predicción y los divisivos. Los nodos de predicción están asociados a un número real mientras que los divisivos funcionan del mismo modo que un árbol de decisión normal, indicándonos el camino a seguir para cada instancia [Curk, T., et al., 2005].

Como ya se mencionó, se eligió el método de clasificación C4.5 (ver Anexo C, aplicación del algoritmo de minería de datos), ya que es el que tuvo el menor error, al evaluar las sílabas por cada uno de los clasificadores que anteriormente se mencionaron. A continuación se muestra en la figura 16 los ejemplos de prueba en el cual se determinó a utilizar el clasificador C4.5 (librerías de Orange para Python).

Palabra silabificada	Valor Esperado			BayesLearner			KNNLearner			C45Learner			TreeLearner		
	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo
Prems	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9
kri	Velar	Anterior-Cerrada	C7	Velar	Alveolar	C1	Velar	Anterior-Cerrada	C1	Velar	Anterior-Cerrada	C1	Velar	Anterior-Cerrada	C1
bir	Bilabial	Alveolar	C11	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2
ek	Anterior-Media	Velar	C7	Anterior-Media	Velar	C1	Anterior-media	Velar	C1	Anterior-media	Velar	C1	Anterior-media	Velar	C1
sa	Alveolar	Central	C6	Alveolar	Central	C2	alveolar	central	C2	alveolar	central	C2	alveolar	central	C2
men	Bilabial	alveolar	C4	Bilabial	alveolar	C4	bilabial	alveolar	C4	bilabial	alveolar	C4	bilabial	alveolar	C4
Te	Dental	Anterior-Media	C8	Dental	Anterior-Media	C8	dental	Anterior-media	C8	dental	Anterior-media	C8	dental	Posterior-media	C8
xas	Velar	Alveolar	C9	Velar	Alveolar	C6	velar	alveolar	C6	velar	alveolar	C6	velar	alveolar	C6
Com	Velar	Bilabial	C7	Velar	palatal	C1	velar	bilabial	C1	velar	bilabial	C10	velar	bilabial	C10
Pu	Bilabial	Posterior-Cerrada	C11	Bilabial	Central	C11	bilabial	central	C11	bilabial	central	C11	bilabial	central	C11
Ta	Dental	Central	C8	Dental	Central	C8	dental	central	C8	dental	central	C8	dental	Posterior-media	C8
Do	Dental	Posterior-Media	C8	Posterior-Cerrada	Central	C5	dental	Anterior-cerrada	C2	dental	central	C10	dental	Posterior-media	C10
ra	Alveolar	Central	C6	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2
es	Anterior-Media	Alveolar	C3	Anterior-Media	Anterior-Media	C1	Anterior-Media	Alveolar	C1	Anterior-Media	Alveolar	C1	Anterior-Media	Alveolar	C1
kue	Velar	Anterior-Media	C2	Velar	Alveolar	C1	Velar	Central	C1	Velar	Anterior-Media	C6	Velar	Anterior-Media	C6
la	Alveolar	Central	C6	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2
Porcentaje	100 %	100%	100%	87.5%	68.75%	31.25%	100%	87.5%	31.25%	100%	87.5%	31.25%	100%	81.25%	31.25%

Figura 16. Clasificación de las sílabas por los métodos, Bayes, knn, C45 y Tree.

Capítulo 6

Conclusiones y Trabajos Futuros

En este trabajo se presentó una propuesta para la creación de vi-sílabas de frases escritas en el idioma español con el fin de aumentar la credibilidad de la sincronización labial en personajes virtuales.

La arquitectura se basa en procesos lingüísticos para implementar las reglas de fonemización, basado en el trabajo de Ríos Mestre [Ríos, Mestre, 1999]. Con la finalidad de que una frase ó palabra se represente en forma de sílabas. Dichas sílabas están compuestas de fonemas; cada fonema esta representado por su punto de articulación, es decir el lugar en la boca donde se pronuncia el fonema.

Se utilizó las librerías de Panda3D para visualizar en tiempo real los visemas correspondientes a cada sílaba. Estos visemas fueron previamente procesados con la utilización de librerías Orange para Python en donde se utilizó el aprendizaje no supervisado K-Means con la finalidad de asignar los puntos de articulación que mejor representan a una sílaba.

Estos resultados se aplicaron en un clasificador C4.5 para determinar a que grupo pertenece una nueva sílaba que se desea procesar. La información obtenida ayuda a realizar la sincronización labial, animando el visema adecuado en Panda3D. Por último se utilizó la librería pyTTS para Python para sincronizar la voz con la animación de vi-sílabas.

Como trabajos futuros se puede observar la adaptación de los visemas de acuerdo al estado emocional de los personajes animados, esto permitirá reforzar los patrones faciales del personaje virtual, mostrando si se encuentra triste, enojado, normal ó feliz.

Este trabajo sienta las bases para crear herramientas ó software en animación de visemas, (movimientos labiales) con la finalidad de ayudar a personas con problemas de dislexias y también en la enseñanza de la pronunciación del idioma, para aquellas personas que el español no es su lengua nativa.

Las aportaciones que se realizaron en este trabajo de Tesis, son la

- 1) Automatización de los procesos de fonemización propuestos por Ríos-Mestre: Reglas de presilabeo, Reglas de transcripción fonológica, Reglas de Silabeo, Reglas de ajuste silábico que fueron realizados en el lenguaje de programación python.
- 2) Se añadieron seis reglas en los procesos de transcripción fonológica y una regla al proceso de silabeo para adaptarlo al idioma español de México.
- 3) Se creó un proceso para transcribir la cadena silabificada a su punto de articulación.
- 4) Se utilizó el algoritmo de K-Means para agrupar las sílabas, en once grupos que representen los once puntos de articulación.
- 5) Se realizó un análisis de frecuencia para identificar el punto de articulación que corresponde al grupo.
- 6) Durante el proceso de sincronización se aplicó el método de C4.5 para elegir el visema adecuado, el cual se mandará a visualizar cuando el motor de text to speech emita el fonema que mejor represente a la sílaba.
- 7) Se diseñaron las texturas (bocas) correspondientes a los puntos de articulación que se aplicaron al actor de Ralph que se anima con el motor de Panda3D.

Anexos A

Anexo A. Procesos de Fonemización

A.1 Implementación del Algoritmo de Presilabeo

Se presenta las reglas del presilabeo (*ver figura 17*) las cuales tiene por objetivo insertar una marca de división silábica, en los fonemas del español de México; se encuentran enumeradas de la regla 1 hasta la regla 7.

FON0.REG "Reglas de presilabación (FON0.REG)."				
Regla	Entrada	Procedimiento	Salida	Ejemplo
Regla 1 FON0	V O L	s,, aou,, b,, !lr,, , , , , , , , , -0,,=0.	V O -L	sub-lingual ab-rogativo ob-repticio
Regla 2 FON0	C C V	_ds,, _ei,, sn,, h,, , , , , , , , , -0,,=0.	C -C V	des-hielo des-hierbo des-hierro des-hueso des-hielo des-hielas
Regla 3 FON0	V C V	,, , , aeiod,, h,, , , , , , , , , -0,,=0.	V -C V	cuenta-hilos barbihecho cariharto
Regla 4 FON0	{t}C	,, _,, t,, !aáéííoóuúrl,, , , , , , , , , -0,,=0.	{t} -C	a-tlisqueño a-tlisquense ca-tleya cenzon-tle chahuis-tle
Regla 5 FON0	{t}C	,, , , anso,, t,, l,, , , , , , , , , -0,,=0.	{t} -C	tlachique tlaco tlacote tlacuache tlascalteca tlaxcalteca tlazol
Regla 6 FON0	V V	,, !qghc,, , , aeouáéóú,, , , aeiouáéíóú,, , , , , , , , , -0,,=0.	V -V	arre-ar lo-ar le-er
Regla 7 FON0	V V	Verbos en -iar con diptongo (no se aplica PRESILAB).	V -V	colegiar enlaciarse intercambiar providenciar

Figura 17. Se presentan las siete reglas de Presilabeo.

Estas siete reglas de Presilabeo presentadas en la figura 17; fueron implementadas en el lenguaje de programación Python. A continuación se presenta la codificación de cada una de estas reglas.

```

def
regla1fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_n
um_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=['s','aou','b']
    presi_posentrada=['','','','']

Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posent
rada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:
        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla1"
    return pre_silabeo,presi_regla

def
regla2fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_n
um_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=['_ds','_ei','_sn']
    presi_posentrada=['','','','']

Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posent
rada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:
        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla2"
    return pre_silabeo,presi_regla

def
regla3fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_n
um_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=['','','aeiou']
    presi_posentrada=['','','','']

Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posent
rada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:
        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla3"
    return pre_silabeo,presi_regla

def
regla4fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_n
um_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=['','_','t']
    presi_posentrada=['','','','']

Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posent
rada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:

```

```

        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla4"
    return pre_silabeo,presi_regla

def regla4ajustada(r4a_grafema_izq,r4a_grafema_der):
    Fonema=''
    num_regla=""
    r4a_pregrafema=['',' ','p']
    r4a_posgrafema=[' ',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r4a_pregrafema,r4a_posgrafema,r4a_grafema_izq,r4a_grafema_der)
    if Estatus_eval==True:
        Fonema='f'
        num_regla="regla4ajustada"
    return Fonema,num_regla

def
regla5fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_num_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=[' ',' ','anso']
    presi_posentrada=['l',' ',' ',' ']

    Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posentrada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:
        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla5"
    return pre_silabeo,presi_regla

def
regla6fon0(focus_presilabeo,presientrada_izq,presientrada_der,pre_num_regla):
    pre_silabeo=''
    presi_regla=""
    presi_preentrada=[' ','!qghc','aeouáéóú']
    presi_posentrada=['r',' ',' ',' ']

    Estatus_eval=procesa_presilabeoIzqDer(presi_preentrada,presi_posentrada,presientrada_izq,presientrada_der)
    if Estatus_eval==True:
        pre_silabeo=str("-")+focus_presilabeo
        presi_regla="regla6"
    return pre_silabeo,presi_regla

```

A.2. Implementación del Algoritmo de Transcripción Fonológica

En esta sección se presenta la implementación en el lenguaje de Programación Python, las reglas de Transcripción Fonológica que tiene como finalidad de convertir los grafemas a fonemas. A continuación se presenta en la figura 18 las veintisiete reglas que se

según Ríos-Mestre son las que permite la conversión de grafemas a fonemas.

FON1.REG "Reglas de transcripción grafema-fonema"					
Regla	Grafema	Procedimiento	Fonema	Definición	Ejemplo
Regla 1	<c>	... C .. leéih.. ... k0.	/k/	Representa al fonema velar oclusivo sordo /k/	casa, copa, cuadro, clase,
Regla 2	<c>	... C .. eéii.. ... z0.	/z/	Representa al fonema interdental fricativo sordo /θ/ cuando precede a los grafemas <e> <ɨ>	cena, cine
Regla 3	<g>	... g .. eéii.. ... x0.	/x/	Representa al fonema velar oclusivo sordo /x/, ante las vocales {e é i, í},	gente, girasol, general
Regla 4	<h>	... h	/Ø/	No representa ningún sonido	Hueso, hueco huevo
Regla 5	<ɨ>	.. _séeíioúú-. í .. seou.. ... y0.	/y/	Representa la vocal alta anterior /i/	Mohiento, romphielos
Regla 6	<ɨ>	.. _seio-.. í .. séóú.. ... y0.	/y/	Representa la vocal alta anterior /i/	Mohiento, romphielos
Regla 7	<ɨ>	.. _séeíioúú-. í .. seou.. ... y0.	/y/	Representa la vocal alta anterior /i/	
Regla 8	<ɨ>	.. lq.. í .. seou.. ... y0.	/y/	Representa la vocal alta anterior /i/	
Regla 9	<ɨ>	.. _seio-.. í .. séóú.. ... y0.	/y/	Representa la vocal alta anterior /i/	
Regla 10	<ɨ>	.. lq.. í .. séóú.. ... y0.	/y/	Representa la vocal alta anterior /i/	
Regla 11	<j>	... j x0.	/x/	Representa al fonema velar fricativo sordo /x/, en todos sus contextos de aparición.	Jamón, juntar, hijo, jabón
Regla 12	<l>	... l .. l.. ... L0.	/L/	Representa a la lateral alveolar sonora /l/	Laguna, leña, limón
Regla 13	<l>	... l	/Ø/	Representa a la lateral palatal sonora /l/	Caballo, lluvia, llama
Regla 14	<n>	... n .. vm.. ... m0.	/m/	Representa al alveolar nasal sonora /n/ los casos de /n/ + /bilabial/ siempre se representará como /m/.	enmendar, anverso, balonvolea balonmano
Regla 15	<q>	... q k0.	/k/	Esta letra forma un grupo grafemático complejo junto con u (este grafema, sin realización fonética)	Queso, química, quesón

FON1.REG "Reglas de transcripción grafema-fonema"					
Regla	Grafema	Procedimiento	Fonema	Definición	Ejemplo
Regla 1 r1r2_ajustada	<c>	„ C ch0.	/ch/	Se coloca a palabras en donde el grafema sea ch	muchacho, coche, chimenea
Regla 2_ajustada	<z>	„ Z z0.	/s/	Se coloca a palabras que el grafema sea Z	zapato, zima
Regla 4_ajustada	<ph>	„ P f0.	/f/	Se coloca a palabras en donde el grafema sea ph	Ralph, raphael
Regla 25_ajustada	<w>	„ W b0.	/b/	Se coloca a palabras en donde el grafema se w	wencelao
Regla 26_ajustada	<x>	„ m, eé, X , i, c, x0.	/x/	Se realizó estos ajustes a la palabra México y sus derivados	México, mexicano, mexicana
Regla26_ajustada_b	<x>	„ t, eé, X , a, x0.	/x/	Se realizó estos ajustes a la palabra Texas y sus derivados	Texas, Texano, texana

Figura 19. Se presentan las seis reglas que fueron añadidas en la conversión fonológica.

Se muestra a continuación la implementación de las veintisiete reglas de conversión de grafema a fonema; y las seis reglas añadidas para la conversión fonológica, mismas que fueron realizadas en el lenguaje de programación Python

```
def regl1(r1_grafema_izq,r1_grafema_der):
    Fonema=' '
    num_regla=" "
    r1_pregrafema=[' ',' ',' ']
    r1_posgrafema=['!eéiíh',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r1_pregrafema,r1_posgrafema,r1_g
rafema_izq,r1_grafema_der)
    if Estatus_eval==True:
        Fonema='k'
        num_regla="regl1"
    return Fonema,num_regla

def reglar1r2_ajustada(r1r2_grafema_izq,r1r2_grafema_der):
    Fonema=' '
    num_regla=" "
    r1r2_pregrafema=[' ',' ',' ']
    r1r2_posgrafema=['h',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r1r2_pregrafema,r1r2_posgrafema,
r1r2_grafema_izq,r1r2_grafema_der)
    if Estatus_eval==True:
        Fonema='ch'
        num_regla="reglar1r2ajustada"
    return Fonema,num_regla

def regla2(r2_grafema_izq,r2_grafema_der):
    Fonema=' '
    num_regla=" "
```

```

r2_pregrafema=[' ',' ',' ' ]
r2_posgrafema=['eéííy',' ',' ',' ' ]

Estatus_eval=procesa_grafemaIzqDer(r2_pregrafema,r2_posgrafema,r2_g
rafema_izq,r2_grafema_der)
    if Estatus_eval==True:
        Fonema='s'
        num_regla="regla2"
    return Fonema,num_regla

def regla2_ajustada(r2a_grafema_izq,r2a_grafema_der):
    Fonema=' '
    num_regla=""
    r2a_pregrafema=[' ',' ',' ' ]
    r2a_posgrafema=[' ',' ',' ',' ' ]

Estatus_eval=procesa_grafemaIzqDer(r2a_pregrafema,r2a_posgrafema,r2
a_grafema_izq,r2a_grafema_der)
    if Estatus_eval==True:
        Fonema='s'
        num_regla="regla2_ajustada"
    return Fonema,num_regla

def regla3(r3_grafema_izq,r3_grafema_der):
    Fonema=' '
    num_regla=""
    r3_pregrafema=[' ',' ',' ' ]
    r3_posgrafema=['eéíí',' ',' ',' ' ]

Estatus_eval=procesa_grafemaIzqDer(r3_pregrafema,r3_posgrafema,r3_g
rafema_izq,r3_grafema_der)
    if Estatus_eval==True:
        Fonema='x'
        num_regla="regla3"
    return Fonema,num_regla

def regla4(r4_grafema_izq,r4_grafema_der):
    Fonema=' '
    num_regla=""
    r4_pregrafema=[' ',' ',' ' ]
    r4_posgrafema=[' ',' ',' ',' ' ]

Estatus_eval=procesa_grafemaIzqDer(r4_pregrafema,r4_posgrafema,r4_g
rafema_izq,r4_grafema_der)
    if Estatus_eval==True:
        Fonema=' '
        num_regla="regla4"
    return Fonema,num_regla

def regla4ajustada(r4a_grafema_izq,r4a_grafema_der):
    Fonema=' '
    num_regla=""
    r4a_pregrafema=[' ',' ','p' ]
    r4a_posgrafema=[' ',' ',' ',' ' ]

Estatus_eval=procesa_grafemaIzqDer(r4a_pregrafema,r4a_posgrafema,r4
a_grafema_izq,r4a_grafema_der)

```

```

    if Estatus_eval==True:
        Fonema='f'
        num_regla="regla4ajustada"
    return Fonema,num_regla

def regla5(r5_grafema_izq,r5_grafema_der):
    Fonema=''
    num_regla=""
    r5_pregrafema=['','_aáeéííoóuú-', 'h']
    r5_posgrafema=['aeou', '', '', '']

Estatus_eval=procesa_grafemaIzqDer(r5_pregrafema,r5_posgrafema,r5_g
rafema_izq,r5_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla5"
    return Fonema,num_regla

def regla6(r6_grafema_izq,r6_grafema_der):
    Fonema=''
    num_regla=""
    r6_pregrafema=['','_aeiou-', 'h']
    r6_posgrafema=['áéóú', '', '', '']

Estatus_eval=procesa_grafemaIzqDer(r6_pregrafema,r6_posgrafema,r6_g
rafema_izq,r6_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla6"
    return Fonema,num_regla

def regla7(r7_grafema_izq,r7_grafema_der):
    Fonema=''
    num_regla=""
    r7_pregrafema=['','','_aáeéííoóú-']
    r7_posgrafema=['aeou', '', '', '']

Estatus_eval=procesa_grafemaIzqDer(r7_pregrafema,r7_posgrafema,r7_g
rafema_izq,r7_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla7"
    return Fonema,num_regla

def regla8(r8_grafema_izq,r8_grafema_der):
    Fonema=''
    num_regla=""
    r8_pregrafema=['','!gq', 'u']
    r8_posgrafema=['aeou', '', '', '']

Estatus_eval=procesa_grafemaIzqDer(r8_pregrafema,r8_posgrafema,r8_g
rafema_izq,r8_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla8"
    return Fonema,num_regla

```

```

def regla9(r9_grafema_izq,r9_grafema_der):
    Fonema=''
    num_regla=""
    r9_pregrafema=['','','_aeio-']
    r9_posgrafema=['áéóú','','','']

    Estatus_eval=procesa_grafemaIzqDer(r9_pregrafema,r9_posgrafema,r9_g
rafema_izq,r9_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla9"
    return Fonema,num_regla

def regla10(r10_grafema_izq,r10_grafema_der):
    Fonema=''
    num_regla=""
    r10_pregrafema=['','!gq','u']
    r10_posgrafema=['áéóú','','','']

    Estatus_eval=procesa_grafemaIzqDer(r10_pregrafema,r10_posgrafema,r1
0_grafema_izq,r10_grafema_der)
    if Estatus_eval==True:
        Fonema='y'
        num_regla="regla10"
    return Fonema,num_regla

def regla11(r11_grafema_izq,r11_grafema_der):
    Fonema=''
    num_regla=""
    r11_pregrafema=['','','']
    r11_posgrafema=['','','','']

    Estatus_eval=procesa_grafemaIzqDer(r11_pregrafema,r11_posgrafema,r1
1_grafema_izq,r11_grafema_der)
    if Estatus_eval==True:
        Fonema='x'
        num_regla="regla11"
    return Fonema,num_regla

def regla12(r12_grafema_izq,r12_grafema_der):
    Fonema=''
    num_regla=""
    r12_pregrafema=['','','!l']
    r12_posgrafema=['l','','','']

    Estatus_eval=procesa_grafemaIzqDer(r12_pregrafema,r12_posgrafema,r1
2_grafema_izq,r12_grafema_der)
    if Estatus_eval==True:
        Fonema='L'
        num_regla="regla12"
    return Fonema,num_regla

def regla13(r13_grafema_izq,r13_grafema_der):
    Fonema=''
    num_regla=""
    r13_pregrafema=['','','l']
    r13_posgrafema=['','','','']

```

```

Estatus_eval=procesa_grafemaIzqDer(r13_pregrafema,r13_posgrafema,r1
3_grafema_izq,r13_grafema_der)
    if Estatus_eval==True:
        Fonema=' '
        num_regla="regla13"
    return Fonema,num_regla

def regla14(r14_grafema_izq,r14_grafema_der):
    Fonema=' '
    num_regla=""
    r14_pregrafema=['',' ','']
    r14_posgrafema=['vm',' ',' ','']

Estatus_eval=procesa_grafemaIzqDer(r14_pregrafema,r14_posgrafema,r1
4_grafema_izq,r14_grafema_der)
    if Estatus_eval==True:
        Fonema='m'
        num_regla="regla14"
    return Fonema,num_regla

def regla15(r15_grafema_izq,r15_grafema_der):
    Fonema=' '
    num_regla=""
    r15_pregrafema=['',' ','']
    r15_posgrafema=['',' ',' ','']

Estatus_eval=procesa_grafemaIzqDer(r15_pregrafema,r15_posgrafema,r1
5_grafema_izq,r15_grafema_der)
    if Estatus_eval==True:
        Fonema='k'
        num_regla="regla15"
    return Fonema,num_regla

def regla16(r16_grafema_izq,r16_grafema_der):
    Fonema=' '
    num_regla=""
    r16_pregrafema=['',' ','_lnsz-mñ']
    r16_posgrafema=['!r',' ',' ','']

Estatus_eval=procesa_grafemaIzqDer(r16_pregrafema,r16_posgrafema,r1
6_grafema_izq,r16_grafema_der)
    if Estatus_eval==True:
        Fonema='R'
        num_regla="regla16"
    return Fonema,num_regla

def regla17(r17_grafema_izq,r17_grafema_der):
    Fonema=' '
    num_regla=""
    r17_pregrafema=['',' ','']
    r17_posgrafema=['r',' ',' ','']

Estatus_eval=procesa_grafemaIzqDer(r17_pregrafema,r17_posgrafema,r1
7_grafema_izq,r17_grafema_der)
    if Estatus_eval==True:
        Fonema='R'

```

```

        num_regla="regla17"
    return Fonema,num_regla

def regla18(r18_grafema_izq,r18_grafema_der):
    Fonema=''
    num_regla=""
    r18_pregrafema=['',' ','r']
    r18_posgrafema=['!r',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r18_pregrafema,r18_posgrafema,r18_grafema_izq,r18_grafema_der)
    if Estatus_eval==True:
        Fonema=' '
        num_regla="regla18"
    return Fonema,num_regla

def regla19(r19_grafema_izq,r19_grafema_der):
    Fonema=''
    num_regla=""
    r19_pregrafema=['',' ',' ']
    r19_posgrafema=['',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r19_pregrafema,r19_posgrafema,r19_grafema_izq,r19_grafema_der)
    if Estatus_eval==True:
        Fonema='u'
        num_regla="regla19"
    return Fonema,num_regla

def regla20(r20_grafema_izq,r20_grafema_der):
    Fonema=''
    num_regla=""
    r20_pregrafema=['',' ','qg']
    r20_posgrafema=['eéíí',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r20_pregrafema,r20_posgrafema,r20_grafema_izq,r20_grafema_der)
    if Estatus_eval==True:
        Fonema=' '
        num_regla="regla20"
    return Fonema,num_regla

def regla21(r21_grafema_izq,r21_grafema_der):
    Fonema=''
    num_regla=""
    r21_pregrafema=['','_aáeéííoóuú-','h']
    r21_posgrafema=['aeoi',' ',' ',' ']

    Estatus_eval=procesa_grafemaIzqDer(r21_pregrafema,r21_posgrafema,r21_grafema_izq,r21_grafema_der)
    if Estatus_eval==True:
        Fonema='w'
        num_regla="regla21"
    return Fonema,num_regla

def regla22(r22_grafema_izq,r22_grafema_der):
    Fonema=' '

```



```

num_regla=""
r22_pregrafema=['','_aeiou-','h']
r22_posgrafema=['áéóí','','','']

Estatus_eval=procesa_grafemaIzqDer(r22_pregrafema,r22_posgrafema,r2
2_grafema_izq,r22_grafema_der)
if Estatus_eval==True:
    Fonema='w'
    num_regla="regla22"
    return Fonema,num_regla

def regla23(r23_grafema_izq,r23_grafema_der):
    Fonema=''
    num_regla=""
    r23_pregrafema=['','','_aáeéííoóuú-']
    r23_posgrafema=['aeoi','','','']

Estatus_eval=procesa_grafemaIzqDer(r23_pregrafema,r23_posgrafema,r2
3_grafema_izq,r23_grafema_der)
if Estatus_eval==True:
    Fonema='w'
    num_regla="regla23"
    return Fonema,num_regla

def regla24(r24_grafema_izq,r24_grafema_der):
    Fonema=''
    num_regla=""
    r24_pregrafema=['','','_aeiou-']
    r24_posgrafema=['áéóí','','','']

Estatus_eval=procesa_grafemaIzqDer(r24_pregrafema,r24_posgrafema,r2
4_grafema_izq,r24_grafema_der)
if Estatus_eval==True:
    Fonema='w'
    num_regla="regla24"
    return Fonema,num_regla

def regla25(r25_grafema_izq,r25_grafema_der):
    Fonema=''
    num_regla=""
    r25_pregrafema=['','','']
    r25_posgrafema=['','','','']

Estatus_eval=procesa_grafemaIzqDer(r25_pregrafema,r25_posgrafema,r2
5_grafema_izq,r25_grafema_der)
if Estatus_eval==True:
    Fonema='b'
    num_regla="regla25"
    return Fonema,num_regla

```

```

def regla25_ajustada(r25a_grafema_izq,r25a_grafema_der):
    Fonema=''
    num_regla=""
    r25a_pregrafema=['',' ','']
    r25a_posgrafema=['eio',' ',' ','']

    Estatus_eval=procesa_grafemaIzqDer(r25a_pregrafema,r25a_posgrafema,
    r25a_grafema_izq,r25a_grafema_der)
    if Estatus_eval==True:
        Fonema='b'
        num_regla="regla25ajustada"
    return Fonema,num_regla

def regla26(r26_grafema_izq,r26_grafema_der):
    Fonema=''
    num_regla=""
    r26_pregrafema=['',' ','']
    r26_posgrafema=['',' ',' ','']

    Estatus_eval=procesa_grafemaIzqDer(r26_pregrafema,r26_posgrafema,r2
    6_grafema_izq,r26_grafema_der)
    if Estatus_eval==True:
        Fonema='ks'
        num_regla="regla26"
    return Fonema,num_regla

def regla26_ajustada(r26a_grafema_izq,r26a_grafema_der):
    Fonema=''
    num_regla=""
    r26a_pregrafema=[' ','m','ée']
    r26a_posgrafema=['i','c',' ','']

    Estatus_eval=procesa_grafemaIzqDer(r26a_pregrafema,r26a_posgrafema,
    r26a_grafema_izq,r26a_grafema_der)
    if Estatus_eval==True:
        Fonema='x'
        num_regla="regla26ajustada"
    return Fonema,num_regla

def regla26_ajustadab(r26b_grafema_izq,r26b_grafema_der):
    Fonema=''
    num_regla=""
    r26b_pregrafema=[' ','t','ée']
    r26b_posgrafema=['a',' ',' ','']

    Estatus_eval=procesa_grafemaIzqDer(r26b_pregrafema,r26b_posgrafema,
    r26b_grafema_izq,r26b_grafema_der)
    if Estatus_eval==True:
        Fonema='x'
        num_regla="regla26ajustadab"
    return Fonema,num_regla

def regla27(r27_grafema_izq,r27_grafema_der):
    Fonema=''
    num_regla=""
    r27_pregrafema=['',' ','']

```

```

r27_posgrafema=['!aáeéííoóú', ',', ' ', '']

Estatus_eval=procesa_grafemaIzqDer(r27_pregrafema,r27_posgrafema,r2
7_grafema_izq,r27_grafema_der)
if Estatus_eval==True:
    Fonema='i'
    num_regla="regla27"
    return Fonema,num_regla
    
```

A.3. Implementación del Algoritmo de Silabeo

Se presenta las dieciocho reglas de silabeo (ver figura 19) las cuales tiene por objetivo de insertar una marca de división silábica. Estas dieciochos reglas se encuentran en el documento de Ríos-Mestre [Mestre, Ríos., 1999].

FON2.REG "Reglas de silabación (FON2.REG)."					
Regla	Entrada	Procedimiento	Salida	Definición	Ejemplo
Regla 1	V C V	,, ,, ááééííoóú,, ptkbdygfz sxmnñlRrW,, aeiou,, , , , , -0,=0.	V - C V	Una consonante entre dos vocales siempre forma sílaba con la vocal siguiente".	cine, peine, cosas, amo
Regla 2	V C V	,, ,, aeiou,, ptkbdygfz sxmnñlRrW,, áéííoú,, , , , , -0,=0.	V - C V	Una consonante entre dos vocales siempre forma sílaba con la vocal siguiente".	casa, aze, piso, ocupar, sucio
Regla 3	{ y _ }	,, ,, ááééííoóú,, ptkbdygfz sxmnñlRrW , , y,, _ , , , , , -0,=0.	{ - y _ }	Las secuencias { C C y _ } (rugby) serán silabeadas por las reglas que separan los grupos de dos consonantes adyacentes.	Curry, ferry, rally, vichy
Regla 4	V O L	,, ,, ááééííoóú,, pkbgf,, r,, , , , , ,, -0,=0.	V - O L	En las reglas de silabación de las secuencias { V O L } = { V - O L }, el foco contendrá las consonantes oclusivas; el contexto posterior, las líquidas; y el contexto anterior, las vocales.	aprobar, afrenta, aplacar, reclamar
Regla 5	V O L	,, ,, ááééííoóú,, dt,, r,, , , , , -0,=0.	V - O L	A las líquidas sólo pueden seguir vocales, por los que estas reglas complementan la silabación de consonantes entre vocales:	Madre, atrio

Regla 6	C s V	,, , bdknlr,, s,, aáeéíioóú,, , , , , -0,=0.	C - s V	Pertenece a sílabas distintas cuando a /s/ le sigue una vocal: el límite silábico se inserta entre /b,d,k,r,l,n/ (que quedan en posición de coda) y /s/ (que queda como ataque).	Adsorción, persona, pulso, cansado
Regla 7	{s, z, x, c, m, ñ, l} C	,, , , szxcmñL,, !aáeéíioóú,, , , , , -0,=0.	{s, z, x, c, m, ñ, l} - C	La fricativa alveolar sorda estará en posición de coda, sea simple o compleja, siempre que preceda a una consonante.	Chispa, desde, mosca, asfalto, columna
Regla 8	{b, k} C	,, , , bk,, !aáeéíioóúúrls_ , , , , , -0,=0.	{b, k} - C	Las consonantes /b / y /k / serán coda silábica siempre que no precedan a una vocal o a las consonantes /r, l/, con la que formarían grupo de /obstruyente/ + /líquida/ en ataque.	Obtener, submarino, acción, acto, abdicar
Regla 9	{p, g, f} C	,, , , pgf,, !aáeéíioóúúrl_ , , , , , , -0,=0.	{p, g, f} - C	Las consonantes /p, g, f/ serán coda silábica siempre que no precedan a una vocal o a las consonantes /r, l/, con la que formarían grupo de /obstruyente/ + /líquida/ en ataque.	Adopción, cápsula, abrupto, dipneo, psiquiatra
Regla 10	{d} C	,, , , d,, !aáeéíioóúúrs_ , , , , , , -0,=0.	{d} - C	La consonante /d/ será coda silábica siempre que no preceda a una vocal o a la consonante /r/, con la que formarían grupo de /obstruyente/ + /líquida/ en ataque.	Advertir, adquirir, adjetivo, administrar
Regla 11	{t} C	,, , , t,, !aáeéíioóúúr_ , , , , , , , , -0,=0.	{t} - C	La consonante /t/ será coda silábica siempre que no preceda a una vocal o a la consonante /r/, con la que formarían grupo de /obstruyente/ + /líquida/ en ataque.	fútbol, algoritmo, atleta, etcétera
Regla 12	{m, n, l, r} C	,, , , nlr,, !aáeéíioóúús_ , , , , , , , , -0,=0.	{m, n, l, r} - C	Será coda silábica siempre que no preceda a una vocal o a la consonante /s/, seguida de consonante, con la que formaría un ataque complejo.	Arbol, arder, arte, ama, confesar, concha
Regla 13	V V	,, , , aáeéíioóú,, aeo,, , , , , , , , -0,=0.	V - V	Las vocales [-altas] adyacentes (acentuadas o no gráficamente) forman grupos heterosilábicos:	aéreo, faraónico, boa
Regla 14	V V	,, , , aeo,, áeíioóú,, , , , , , , , -0,=0.	V - V	La combinación de dos vocales adyacentes [-alta] no acentuada y [+alta] acentuada forma un hiato decreciente: / a , e , o / - / i , ú /	Baúl, rehusó, oleica Poético,
Regla 15	V V	,, , , i,, ii,, , , , , , , , -0,=0.	V V	La ortografía no permite distinguir cuando dos vocales [+altas] no iguales adyacentes forman diptongo o hiato. Habrá hiato cuando ambas sean iguales	chiita
Regla 16	V V	,, , , í,, í,, , , , , , , , -0,=0.	V - V	Habrá hiato cuando ambas sean iguales (independientemente de la posición de la tilde).	
Regla 17	V V	,, , , u,, úu,, , , , , , , , -0,=0.	V - V	Habrá hiato cuando ambas sean iguales (independientemente de la posición de la tilde).	Samuñi
Regla 18	V V	,, , , ú,, u,, , , , , , , , -0,=0.	V - V	Habrá hiato cuando ambas sean iguales (independientemente de la posición de la tilde).	

Figura 20. Se presentan las dieciocho reglas de Silabeo.

Se añadió una regla en el proceso de Silabeo esta regla la denominamos reglar1r2r3_ajustada, con la finalidad de que se realizó correctamente el silabeo al grafema <ch> en la aplicación de las reglas fonológicas a palabras como, muchacho, chimenea, chavo, etc.

FON2.REG "Reglas de Silabación"					
Regla	Entrada	Procedimiento	Salida	Definición	Ejemplo
Regla 1 r1r2r3_ajustada	VCCV	.., ááééííoóú., C., h., aeiou., .. -0=0.	V-CCV	Se coloca a palabras en donde el grafema sea ch.	muchacho, coche, chimenea

Figura 21. Reglar1r2r3_ajustada realiza el proceso de silabeo del grafema <ch>.

Se presenta la implementación de las dieciocho reglas de Silabeo y la regla r1r2r3_ajustada, que fueron realizadas en el lenguaje de programación en Python.

```
def regla1(r1_focus_silaba,r1_entrada_izq,r1_entrada_der):
    silabeo=''
    num2_regla=""
    r1_preentrada=['','','ááééííoóú']
    r1_posentrada=['aeiou','','','']

    Estatus_eval=procesa_silabeoIzqDer(r1_preentrada,r1_posentrada,r1_
    entrada_izq,
        r1_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r1_focus_silaba
        num2_regla="regla1"
    return silabeo,num2_regla

def
reglar1r2r3_ajustada(r1r2r3_focus_silaba,r1r2r3_entrada_izq,r1r2r3_
entrada_der):
    silabeo=''
    num2_regla=""
    r1r2r3_preentrada=['','','ááééííoóú']
    r1r2r3_posentrada=['h','aeiou','','']

    Estatus_eval=procesa_silabeoIzqDer(r1r2r3_preentrada,r1r2r3_posentr
    ada,r1r2r3_entrada_izq,r1r2r3_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r1r2r3_focus_silaba
        num2_regla="reglar1r2r3_ajustada"
    return silabeo,num2_regla

def regla2(r2_focus_silaba,r2_entrada_izq,r2_entrada_der):
    silabeo=''
    num2_regla=""
    r2_preentrada=['','','aeiou']
    r2_posentrada=['áéíóú','','','']

    Estatus_eval=procesa_silabeoIzqDer(r2_preentrada,r2_posentrada,r2_e
    ntrada_izq,r2_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r2_focus_silaba
        num2_regla="regla2"
    return silabeo,num2_regla
```

```

def regla3(r3_focus_silaba,r3_entrada_izq,r3_entrada_der):
    silabeo=''
    num2_regla=""
    r3_preentrada=['','','aáeéííoóuú']
    r3_posentrada=['y','_','','']

    Estatus_eval=procesa_silabeoIzqDer(r3_preentrada,r3_posentrada,r3_entr
ntrada_izq,r3_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r3_focus_silaba
        num2_regla="regla3"
    return silabeo,num2_regla

def regla4(r4_focus_silaba,r4_entrada_izq,r4_entrada_der):
    silabeo=''
    num2_regla=""
    r4_preentrada=['','','aáeéííoóuú']
    r4_posentrada=['r1','','','']

    Estatus_eval=procesa_silabeoIzqDer(r4_preentrada,r4_posentrada,r4_entr
ntrada_izq,r4_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r4_focus_silaba
        num2_regla="regla4"
    return silabeo,num2_regla

def regla5(r5_focus_silaba,r5_entrada_izq,r5_entrada_der):
    silabeo=''
    num2_regla=""
    r5_preentrada=['','','aáeéííoóuú']
    r5_posentrada=['r','','','']

    Estatus_eval=procesa_silabeoIzqDer(r5_preentrada,r5_posentrada,r5_entr
ntrada_izq,r5_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r5_focus_silaba
        num2_regla="regla5"
    return silabeo,num2_regla

def regla6(r6_focus_silaba,r6_entrada_izq,r6_entrada_der):
    silabeo=''
    num2_regla=""
    r6_preentrada=['','','bdknlr']
    r6_posentrada=['aáeéííoóuú','','','']

    Estatus_eval=procesa_silabeoIzqDer(r6_preentrada,r6_posentrada,r6_entr
ntrada_izq,r6_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r6_focus_silaba
        num2_regla="regla6"
    return silabeo,num2_regla

def regla7(r7_focus_silaba,r7_entrada_izq,r7_entrada_der):
    silabeo=''
    num2_regla=""
    r7_preentrada=['','','szxcmñL']

```

```

    r7_posentrada=[' ',' ',' ',' ']

Estatus_eval=procesa_silabeoIzqDer(r7_preentrada,r7_posentrada,r7_entr
ntrada_izq,r7_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r7_focus_silaba
        num2_regla="regla7"
    return silabeo,num2_regla

def regla8(r8_focus_silaba,r8_entrada_izq,r8_entrada_der):
    silabeo=''
    num2_regla=""
    r8_preentrada=[' ',' ','bk']
    r8_posentrada=[' ',' ',' ',' ']

Estatus_eval=procesa_silabeoIzqDer(r8_preentrada,r8_posentrada,r8_entr
ntrada_izq,r8_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r8_focus_silaba
        num2_regla="regla8"
    return silabeo,num2_regla

def regla9(r9_focus_silaba,r9_entrada_izq,r9_entrada_der):
    silabeo=''
    num2_regla=""
    r9_preentrada=[' ',' ','pgf']
    r9_posentrada=[' ',' ',' ',' ']

Estatus_eval=procesa_silabeoIzqDer(r9_preentrada,r9_posentrada,r9_entr
ntrada_izq,r9_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r9_focus_silaba
        num2_regla="regla9"
    return silabeo,num2_regla

def regla10(r10_focus_silaba,r10_entrada_izq,r10_entrada_der):
    silabeo=''
    num2_regla=""
    r10_preentrada=[' ',' ','d']
    r10_posentrada=[' ',' ',' ',' ']

Estatus_eval=procesa_silabeoIzqDer(r10_preentrada,r10_posentrada,r10_entr
ntrada_izq,r10_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r10_focus_silaba
        num2_regla="regla10"
    return silabeo,num2_regla

def regla11(r11_focus_silaba,r11_entrada_izq,r11_entrada_der):
    silabeo=''
    num2_regla=""
    r11_preentrada=[' ',' ','t']
    r11_posentrada=[' ',' ',' ',' ']

Estatus_eval=procesa_silabeoIzqDer(r11_preentrada,r11_posentrada,r11_entr
ntrada_izq,r11_entrada_der)
    if Estatus_eval==True:

```

```

        silabeo=str("-")+r11_focus_silaba
        num2_regla="regla11"
        return silabeo,num2_regla

def regla12(r12_focus_silaba,r12_entrada_izq,r12_entrada_der):
    silabeo=''
    num2_regla=""
    r12_preentrada=['',' ','nlr']
    r12_posentrada=['',' ',' ','']

    Estatus_eval=procesa_silabeoIzqDer(r12_preentrada,r12_posentrada,r1
2_entrada_izq,r12_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r12_focus_silaba
        num2_regla="regla12"
        return silabeo,num2_regla

def regla13(r13_focus_silaba,r13_entrada_izq,r13_entrada_der):
    silabeo=''
    num2_regla=""
    r13_preentrada=['',' ','aáéíóóú']
    r13_posentrada=['',' ',' ','']

    Estatus_eval=procesa_silabeoIzqDer(r13_preentrada,r13_posentrada,r1
3_entrada_izq,r13_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r13_focus_silaba
        num2_regla="regla13"
        return silabeo,num2_regla

def regla14(r14_focus_silaba,r14_entrada_izq,r14_entrada_der):
    silabeo=''
    num2_regla=""
    r14_preentrada=['',' ','aeo']
    r14_posentrada=['',' ',' ','']

    Estatus_eval=procesa_silabeoIzqDer(r14_preentrada,r14_posentrada,r1
4_entrada_izq,r14_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r14_focus_silaba
        num2_regla="regla14"
        return silabeo,num2_regla

def regla15(r15_focus_silaba,r15_entrada_izq,r15_entrada_der):
    silabeo=''
    num2_regla=""
    r15_preentrada=['',' ','i']
    r15_posentrada=['',' ',' ','']

    Estatus_eval=procesa_silabeoIzqDer(r15_preentrada,r15_posentrada,r1
5_entrada_izq,r15_entrada_der)
    if Estatus_eval==True:
        silabeo=str("-")+r15_focus_silaba
        num2_regla="regla15"
        return silabeo,num2_regla

def regla16(r16_focus_silaba,r16_entrada_izq,r16_entrada_der):

```



```

silabeo=''
num2_regla=""
r16_preentrada=['',' ','í']
r16_posentrada=['',' ',' ','']

Estatus_eval=procesa_silabeoIzqDer(r16_preentrada,r16_posentrada,r1
6_entrada_izq,r16_entrada_der)
if Estatus_eval==True:
    silabeo=str("-")+r16_focus_silaba
    num2_regla="regla16"
    return silabeo,num2_regla

def regla17(r17_focus_silaba,r17_entrada_izq,r17_entrada_der):
    silabeo=''
    num2_regla=""
    r17_preentrada=['',' ','u']
    r17_posentrada=['',' ',' ','']

Estatus_eval=procesa_silabeoIzqDer(r17_preentrada,r17_posentrada,r1
7_entrada_izq,r17_entrada_der)
if Estatus_eval==True:
    silabeo=str("-")+r17_focus_silaba
    num2_regla="regla17"
    return silabeo,num2_regla

def regla18(r18_focus_silaba,r18_entrada_izq,r18_entrada_der):
    silabeo=''
    num2_regla=""
    r18_preentrada=['',' ','ú']
    r18_posentrada=['',' ',' ','']

Estatus_eval=procesa_silabeoIzqDer(r18_preentrada,r18_posentrada,r1
8_entrada_izq,r18_entrada_der)
if Estatus_eval==True:
    silabeo=str("-")+r18_focus_silaba
    num2_regla="regla18"
    return silabeo,num2_regla

```

A.4. Implementación del Algoritmo de Ajuste Silábico

A continuación se presentan las diecisiete reglas de ajuste silábico (*ver figura 20*). La finalidad de este algoritmo es de corregir palabras que no son nativas en el idioma español de México, por ejemplo *ansgtron* → *ans-trom*.

FON3.REG "Reglas de ajuste silábico (FON3.REG)."				
Regla	Entrada	Procedimiento	Salida	Ejemplo
Regla 1	{ _ s - }	... _ , s , - , ... e0,=0.	{ _ es - }	sparring sprint stábat staccato
Regla 2	{ _ C - }	... _ , !aáéííóóúú , - , ...	{ _ Ø - }	gneis cneorácea gnéisico gnetáceo
Regla 3	{ - C - }	... , !aáéííóóúú f , - , ...	{ - Ø - }	metempicosis roentgen sprints
Regla 4	{ - C _ }	... , !aáéííóóúú f y , - , ...	{ - Ø _ }	dock trust roquefort portland lord
Regla 5	{ p - p }	... , aáéííóóúú , p , - , p , !- , ...	{ Ø - p }	hippy
Regla 6	{ t - t }	... , aáéííóóúú , t , - , t , !- , ...	{ Ø - t }	larghetto tutti frutti séter toilette
Regla 7	{ k - k }	... , aáéííóóúú , k , - , k , !- , ...	{ Ø - k }	staccato becqueriano jockey ticket
Regla 8	{ s - s }	... , s , - , s , ...	{ Ø - s }	Bessemer dossier grosso picassiano cross
Regla 9	{ f - f }	... , f , - , f , ...	{ Ø - f }	affaire offset bluff staff staffs
Regla 10	{ - C s }	... , bdknlr , s , ...	{ - Ø s }	
Regla 11	{ L }	... , aáéííóóúú , L , - , ... 10.	{ l }	detall nonparell hall roll bulldog
Regla 12	{ - f }	... , - , f , - , ...	{ f }	golf surf windsurf surf windsurf
Regla 13	{ - s }	... , - , s , - , ...	{ s }	biceps argots hámster módens
Regla 14	{ - y _ }	... , - , y , - , ...	{ y _ }	dandy hobby rugby whisky
Regla 15	{ _ - }	... , - , !- f sy , ...	{ _ Ø }	
Regla 16	{ - - }	... , - , - , ...	{ Ø - }	
Regla 17	{ - _ }	... , - , - , ...	{ Ø _ }	

Figura 22. Se presentan las diecisiete reglas de Ajuste silábico.

Se muestran la implementación de las diecisiete reglas de Ajuste Silábico en el lenguaje de Programación en Python.

```
def regla1(r1_focus_silabico,r1_entrada3_izq,r1_entrada3_der):
    silabico=''
    num3_regla=""
    r1_preentrada=[' ',' ','_']
    r1_posentrada=['-',' ',' ','_']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r1_preentrada,r1_posentra
    da,r1_entrada3_izq,r1_entrada3_der)
    if Estatus_eval==True:
        silabico=str("e")+r1_focus_silabico
        num3_regla="regla1"
    return silabico,num3_regla

def regla2(r2_focus_silabico,r2_entrada3_izq,r2_entrada3_der):
    silabico=''
```

```
num3_regla=""
r2_preentrada=[' ','_']
r2_posentrada=['-',' ','_']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r2_preentrada,r2_posentra
ada,r2_entrada3_izq,r2_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla2"
    return silabico,num3_regla
def regla3(r3_focus_silabico,r3_entrada3_izq,r3_entrada3_der):
    silabico=''
    num3_regla=""
    r3_preentrada=[' ','-']
    r3_posentrada=['-',' ','_']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r3_preentrada,r3_posentra
ada,r3_entrada3_izq,r3_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla3"
    return silabico,num3_regla
```

```

def regla4(r4_focus_silabico,r4_entrada3_izq,r4_entrada3_der):
    silabico=''
    num3_regla=""
    r4_preentrada=[' ',' ','-']
    r4_posentrada=['_',' ',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r4_preentrada,r4_posentra
    da,r4_entrada3_izq,r4_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla4"
    return silabico,num3_regla

def regla5(r5_focus_silabico,r5_entrada3_izq,r5_entrada3_der):
    silabico=''
    num3_regla=""
    r5_preentrada=[' ',' ','aáeéííoóuú']
    r5_posentrada=['-','p','!-_','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r5_preentrada,r5_posentra
    da,r5_entrada3_izq,r5_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla5"
    return silabico,num3_regla

def regla6(r6_focus_silabico,r6_entrada3_izq,r6_entrada3_der):
    silabico=''
    num3_regla=""
    r6_preentrada=[' ',' ','aáeéííoóuú']
    r6_posentrada=['-','t','!-_','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r6_preentrada,r6_posentra
    da,r6_entrada3_izq,r6_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla6"
    return silabico,num3_regla

def regla7(r7_focus_silabico,r7_entrada3_izq,r7_entrada3_der):
    silabico=''
    num3_regla=""
    r7_preentrada=[' ',' ','aáeéííoóuú']
    r7_posentrada=['-','k','!-_s','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r7_preentrada,r7_posentra
    da,r7_entrada3_izq,r7_entrada3_der)
    if Estatus_eval==True:
        silabico=r7_focus_silabico
        num3_regla="regla7"
    return silabico,num3_regla

```

```

def regla8(r8_focus_silabico,r8_entrada3_izq,r8_entrada3_der):
    silabico=''
    num3_regla=""
    r8_preentrada=['',' ','']
    r8_posentrada=['-','s',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r8_preentrada,r8_posentra
    da,r8_entrada3_izq,r8_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla8"
    return silabico,num3_regla

def regla9(r9_focus_silabico,r9_entrada3_izq,r9_entrada3_der):
    silabico=''
    num3_regla=""
    r9_preentrada=['',' ','']
    r9_posentrada=['-','f',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r9_preentrada,r9_posentra
    da,r9_entrada3_izq,r9_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla9"
    return silabico,num3_regla

def regla10(r10_focus_silabico,r10_entrada3_izq,r10_entrada3_der):
    silabico=''
    num3_regla=""
    r10_preentrada=['',' ','-']
    r10_posentrada=['s',' ',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r10_preentrada,r10_posen
    trada,r10_entrada3_izq,r10_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num3_regla="regla10"
    return silabico,num3_regla

def regla11(r11_focus_silabico,r11_entrada3_izq,r11_entrada3_der):
    silabico=''
    num3_regla=""
    r11_preentrada=['',' ','aáeéííoóuú']
    r11_posentrada=['-',' ',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r11_preentrada,r11_posen
    trada,r11_entrada3_izq,r11_entrada3_der)
    if Estatus_eval==True:
        silabico=r11_focus_silabico
        num3_regla="regla11"
    return silabico,num3_regla

```

```

def regla12(r12_focus_silabico,r12_entrada3_izq,r12_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r12_entrada3_der
    r12_preentrada=['',' ','']
    r12_posentrada=['f','-',' ','']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r12_preentrada,r12_posen
trada,r12_entrada3_izq,r12_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num32_regla="regla12"
    return silabico,num32_regla,entrada3_der

def regla13(r13_focus_silabico,r13_entrada3_izq,r13_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r13_entrada3_der
    r13_preentrada=['',' ','']
    r13_posentrada=['s','-',' ','']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r13_preentrada,r13_posen
trada,r13_entrada3_izq,r13_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num32_regla="regla13"
    return silabico,num32_regla,entrada3_der

def regla14(r14_focus_silabico,r14_entrada3_izq,r14_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r14_entrada3_der
    r14_preentrada=['',' ','']
    r14_posentrada=['y','-',' ','']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r14_preentrada,r14_posen
trada,r14_entrada3_izq,r14_entrada3_der)
    if Estatus_eval==True:
        silabico=r14_focus_silabico
        num32_regla="regla14"
    return silabico,num32_regla,entrada3_der

def regla15(r15_focus_silabico,r15_entrada3_izq,r15_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r15_entrada3_der
    r15_preentrada=['',' ','_']
    r15_posentrada=['!-_fsy',' ',' ','']

Estatus_eval=procesa_ajuste_silabicoIzqDer(r15_preentrada,r15_posen
trada,r15_entrada3_izq,r15_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num32_regla="regla15"
    return silabico,num32_regla,entrada3_der

```

```
def regla16(r16_focus_silabico,r16_entrada3_izq,r16_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r16_entrada3_der
    r16_preentrada=['',' ','']
    r16_posentrada=['-',' ',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r16_preentrada,r16_posen
trada,r16_entrada3_izq,r16_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num32_regla="regla16"
    return silabico,num32_regla,entrada3_der

def regla17(r17_focus_silabico,r17_entrada3_izq,r17_entrada3_der):
    silabico=''
    num32_regla=""
    entrada3_der=r17_entrada3_der
    r17_preentrada=['',' ','']
    r17_posentrada=['_',' ',' ','']

    Estatus_eval=procesa_ajuste_silabicoIzqDer(r17_preentrada,r17_posen
trada,r17_entrada3_izq,r17_entrada3_der)
    if Estatus_eval==True:
        silabico=str(' ')
        num32_regla="regla17"
    return silabico,num32_regla,entrada3_der
```

Anexos B

B.1 Implementación del Algoritmo por Punto de Articulación

Se muestra la implementación en el lenguaje de Programación Python las reglas del punto de articulación que tiene por objetivo convertir la cadena previamente procesada por los procesos de fonemización a su punto de articulación correspondiente, añadiendo un identificador inicial por ejemplo si el punto de articulación es Central, el identificador será C.

```
def punto_articulacion(Fonema,palabra):
    punto_artic=''
    cad_art=""
    lista_acento=["á","é","í","ó","ú","ñ"]
    for z in range(len(lista_acento)):
        a=unicode(lista_acento[z],"cp1252")
        cad_art =cad_art+a
    if Fonema=='p'or Fonema=='b'or Fonema=='m':
        punto_artic='B'
    elif Fonema=='d'or Fonema=='t':
        punto_artic='D'
    elif Fonema=='k'or Fonema=='g'or Fonema=='x' or Fonema=='w':
        punto_artic='V'
    elif Fonema=='f':
        punto_artic='L'
    elif Fonema=='s'or Fonema=='r'or Fonema=='R' or Fonema=='l' or
Fonema=='n'or Fonema=='ks':
        punto_artic='A'
    elif (Fonema=='c' and palabra[1]=='h')or Fonema=='y'or
Fonema=='L'or Fonema==cad_art[5]:
        punto_artic='P'
    elif Fonema=='a' or Fonema==cad_art[0]:
        punto_artic='C'
    elif Fonema=='e' or Fonema==cad_art[1]:
        punto_artic='I'
    elif Fonema=='i' or Fonema==cad_art[2]:
        punto_artic='N'
    elif Fonema=='o' or Fonema==cad_art[3]:
        punto_artic='F'
    elif Fonema=='u' or Fonema==cad_art[4]:
        punto_artic='O'
    else:
        punto_artic=sin_clasificar(Fonema)
    return punto_artic
```


Se presenta la tabla de grafemas correspondientes a cada uno de los fonemas del español de México, y la relación con el punto de articulación, en donde cada punto de articulación se ha representado con un identificador inicial (*ver tabla 29*).

Grafema	Fonema	Identificador	Punto de Articulación	Numeración
<a>	/a/	'C'	Central	001
 <v> <wo> <we>	/b/	'B'	Bilabial	002
<ch>	/ch/	'P'	Palatal	003
<d>	/d/	'D'	Dental	004
<e>	/e/	'I'	Anterior-Media	005
<f>	/f/	'L'	Labiodental	006
<g><gu>	/g/	'V'	Velar	007
<i>	/i/	'N'	Anterior-Cerrada	008
<ca> <co> <cu> <qu> <k>	/k/	'V'	Velar	007
<l>	/l/	'A'	Alveolar	009
<ll>	/L/	'P'	Palatal	003
<m>	/m/	'B'	Bilabial	002
<n>	/n/	'A'	Alveolar	009
<ñ>	/ñ/	'P'	Palatal	003
<o>	/o/	'F'	Posterior-Media	010
<p>	/p/	'B'	Bilabial	002
<r>	/r/	'A'	Alveolar	009
<rr>	/R/	'A'	Alveolar	009
<za><ze> <zi> <zo> <zu> <sa> <se> <si> <so> <su> <ce> <ci> <xo><xe>	/s/	'A'	Alveolar	009
<t>	/t/	'D'	Dental	004
<u>	/u/	'O'	Posterior-Cerrada	011
<x> <j> <g>	/x/	'V'	Velar	007
<hi> <y>	/y/	'P'	Palatal	003
<hu>	/w/	'V'	Velar	007

Tabla 29. Clasificación de los grafemas a fonemas y su punto de articulación.

Anexos C

C.1. Implementación del Algoritmo de Selección de Visemas (Panda3D)

Se realizó la implementación del algoritmo de selección de visemas con la finalidad de cargar los visemas (imágenes de labios) de cada uno de los puntos de articulación del español México. A continuación se muestra la codificación de la función cargar actor en donde se cargara al personaje Ralph, uno de los modelos de la consola de Panda3D.

```
def cargar_ralph(Actor, OnscreenText, DirectObject):
    actor_ralph = Actor.Actor("modelos_ralph/ralph",
                              {"walk": "modelos_ralph/ralph-walk",
                               "run": "modelos_ralph/ralph-run",
                               "jump": "modelos_ralph/ralph-jump"})

    #Carga al actor, (donde primero va el modelo y luego sus
    animaciones )
    # y su bucle de animacion

    #Se carga en memoria las texturas
    mitextura = loader.loadTexture ("modelos_ralph/ralph_callado.jpg")
    mitextura_A = loader.loadTexture("modelos_ralph/boca_Central.jpg")
    mitextura_B= loader.loadTexture("modelos_ralph/boca_Bilabial.jpg")
    mitextura_D= loader.loadTexture ("modelos_ralph/boca_Dental.jpg")
    mitextura_CH= loader.loadTexture("modelos_ralph/boca_Palatal.jpg")
    mitextura_E= loader.loadTexture
    ("modelos_ralph/boca_Anterior_Media.jpg")
    mitextura_F= loader.loadTexture
    ("modelos_ralph/boca_LabioDental.jpg")
    mitextura_G= loader.loadTexture ("modelos_ralph/boca_Velar.jpg")
    mitextura_I= loader.loadTexture
    ("modelos_ralph/boca_Anterior_Cerrada.jpg")
    mitextura_K= loader.loadTexture ("modelos_ralph/boca_Velar.jpg")
    mitextura_L= loader.loadTexture ("modelos_ralph/boca_Alveolar.jpg")
    mitextura_LL= loader.loadTexture ("modelos_ralph/boca_Palatal.jpg")
    mitextura_M= loader.loadTexture ("modelos_ralph/boca_Bilabial.jpg")
    mitextura_N= loader.loadTexture ("modelos_ralph/boca_Alveolar.jpg")
    mitextura_NN= loader.loadTexture ("modelos_ralph/boca_Palatal.jpg")
    mitextura_O= loader.loadTexture
    ("modelos_ralph/boca_Posterior_Media.jpg")
```

```

mitextura_P= loader.loadTexture ("modelos_ralph/boca_Bilabial.jpg")
mitextura_R= loader.loadTexture ("modelos_ralph/boca_Alveolar.jpg")
mitextura_RR= loader.loadTexture("modelos_ralph/boca_Alveolar.jpg")
mitextura_S= loader.loadTexture ("modelos_ralph/boca_Alveolar.jpg")
mitextura_T= loader.loadTexture ("modelos_ralph/boca_Dental.jpg")
mitextura_U= loader.loadTexture
("modelos_ralph/boca_Posterior_Cerrada.jpg")
mitextura_W= loader.loadTexture ("modelos_ralph/boca_Velar.jpg")
mitextura_X= loader.loadTexture ("modelos_ralph/boca_Velar.jpg")
mitextura_Y= loader.loadTexture ("modelos_ralph/boca_Palatal.jpg")

actor_ralph.setScale(0.5,0.5,0.5)
actor_ralph.setPos(-1.5,10,-1)
actor_ralph.setTexture (mitextura, 1)
actor_ralph.reparentTo(render)
#ralph.loop("walk")

#Cargar un tipo de fuente para el amnejo de tetxo
cmr12 = loader.loadFont('cmr12.egg')

textObject = OnscreenText(
text = ' Teclas: n,a,e,i,o,u,escape .',
bg=(0,0,0,1), frame=(1,1,1,0) ,mayChange= 1 , font =(cmr12)
,pos= (0,-0.97), fg=(1,1,1,1), scale = 0.07)

return
(actor_ralph,textObject,mitextura,mitextura_A,mitextura_B,mitextura_
CH,mitextura_D,mitextura_E,mitextura_F,mitextura_G,mitextura_I,

mitextura_K,mitextura_L,mitextura_LL,mitextura_M,mitextura_N,mitextu
ra_NN,mitextura_O,mitextura_P,mitextura_R,mitextura_RR,mitextura_S,

mitextura_T,mitextura_U,mitextura_W,mitextura_X,mitextura_Y)

```



Figura 23. Carga del actor Ralph con todas las posibles combinaciones de visemas.

C.2. Implementación del Algoritmo de K-means

Se utilizó el método de k-means que se encuentra incluido en las librerías de Orange en Python, con la finalidad de agrupar en once grupos los once elementos del punto de articulación a partir de una base de datos de entrenamiento. A continuación se muestra la codificación del algoritmo k-means

```
import orange
import orngClustering

data = orange.ExampleTable("id_punto_artic_medias")
km = orngClustering.KMeans(data, 11,
initialization=orngClustering.kmeans_init_diversity)
orngClustering.plot_silhouette(km, "kmeans-silhouette_artic.png")
print "valor de la iteracion",km.iteration
print "valor de la distance",km.distance
print "valor de k",km.k
```

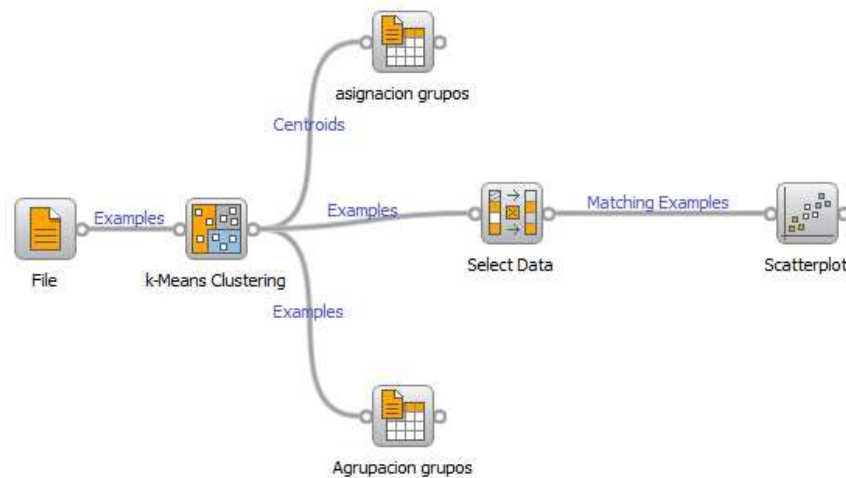


Figura 24. Base de datos de entrenamiento, usando el algoritmo de k-means para Python

Punto de Articulación										
Central	Bilabial	Palatal	Dental	Anterior-Media	Labiodental	Velar	Anterior-Cerrada	Alveolar	Posterior-Media	Posterior-Cerrada
C6- 99.61%			C8- 93.83%	C5- 97.37%		C7 100%	C2- 69.36%	C3 - 99.59%	C1- 57.35%	
C10 60%								C4- 96.29%		
C11 87.80%								C9- 93.53%		

Figura 25. Asignación de los once grupos ó centroides a cada punto de articulación.

C.3 Aplicación del Algoritmo de Minería de Datos

Se realizó la implementación de un algoritmo de minería de datos (clasificador de C4.5 que se encuentran incluido en las librerías de Orange con Python.) Para seleccionar que clase corresponde a un punto de articulación y así representar la animación de visemas. A continuación se presenta la codificación del método C4.5.

```
def clasificador(sil_articulada):
    res_kmeans=orange.ExampleTable("kmedias_orange")
    cad=""
    cad=""
    cad2=""
    cad3=""
    cad4=""
    cad5=""
    cad6=""
    print "La silaba articulada",sil_articulada
    cadena = sil_articulada
    cad= cadena[0:3]
```

```

cad2= cadena[3:6]
cad3= cadena[6:9]
cad4= cadena[9:12]
cad5= cadena[12:15]
cad6= cadena[15:18]
valor1=cad.lstrip('0')
valor2=cad2.lstrip('0')
valor3=cad3.lstrip('0')
valor4=cad4.lstrip('0')
valor5=cad5.lstrip('0')
valor6=cad6.lstrip('0')
valor7='?'
prueba_kmeans=orange.Example(res_kmeans.domain,[valor1,valor2,v
alor3,valor4,valor5,valor6,valor7])
c45=orange.C45Learner(res_kmeans) #creacion del clasificador
C45
kmedias_c45=c45(prueba_kmeans) #Evaluacion en C45
print "Resultado de Aplicar el clasificador de C45: "
print "Resultado BD Punto de articulacion segun kmedias ",
kmedias_c45
print

print kmedias_c45
return kmedias_c45

```

Se presenta la tabla con cada uno de los clasificadores, Bayes, Knn, C4.5 y Tree, métodos que se encuentran incluidos en las librerías de Orange en Python, que fueron utilizadas para hacer la validación de varios ejemplos de sílabas, en donde cada uno de los clasificadores mostraron diferentes resultados (*ver tabla 30*).

Palabra silabificada	Valor Esperado			Bayes			KNN			C4.5			ID3		
	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo	Inicial	Final	Gpo
Preins	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9	Bilabial	Alveolar	C9
kri	Velar	Anterior-Cerrada	C7	Velar	Alveolar	C1	Velar	Anterior-Cerrada	C1	Velar	Anterior-Cerrada	C1	Velar	Anterior-Cerrada	C1
bir	Bilabial	Alveolar	C11	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2	Bilabial	Alveolar	C2
ek	Anterior-Media	Velar	C7	Anterior-Media	Velar	C1	Anterior-media	Velar	C1	Anterior-media	Velar	C1	Anterior-media	Velar	C1
sa	Alveolar	Central	C6	Alveolar	Central	C2	alveolar	central	C2	alveolar	central	C2	alveolar	central	C2
men	Bilabial	alveolar	C4	Bilabial	alveolar	C4	bilabial	alveolar	C4	bilabial	alveolar	C4	bilabial	alveolar	C4
Te	Dental	Anterior-Media	C8	Dental	Anterior-Media	C8	dental	Anterior-media	C8	dental	Anterior-media	C8	dental	Posterior-media	C8
xas	Velar	Alveolar	C9	Velar	Alveolar	C6	velar	alveolar	C6	velar	alveolar	C6	velar	alveolar	C6
Com	Velar	Bilabial	C7	Velar	palatal	C1	velar	bilabial	C1	velar	bilabial	C10	velar	bilabial	C10
Pu	Bilabial	Posterior-Cerrada	C11	Bilabial	Central	C11	bilabial	central	C11	bilabial	central	C11	bilabial	central	C11
Ta	Dental	Central	C8	Dental	Central	C8	dental	central	C8	dental	central	C8	dental	Posterior-media	C8
Do	Dental	Posterior-Media	C8	Posterior-Cerrada	Central	C5	dental	Anterior-cerrada	C2	dental	central	C10	dental	Posterior-media	C10
ra	Alveolar	Central	C6	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2
es	Anterior-Media	Alveolar	C3	Anterior-Media	Anterior-Media	C1	Anterior-Media	Alveolar	C1	Anterior-Media	Alveolar	C1	Anterior-Media	Alveolar	C1
kue	Velar	Anterior-Media	C2	Velar	Alveolar	C1	Velar	Central	C1	Velar	Anterior-Media	C6	Velar	Anterior-Media	C6
la	Alveolar	Central	C6	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2	Alveolar	Central	C2
Porcentaje	100 %	100%	100%	87.5%	68.75%	31.25%	100%	87.5%	31.25%	100%	87.5%	31.25%	100%	81.25%	31.25%

Tabla 30. Clasificación de las sílabas por los métodos, Bayes, knn, C45 y Tree.

Anexos D

D.1 Aplicación del Algoritmo de TTS en Python

Se utilizó las librerías de PyTTS con Python, con el objetivo de reproducir la voz, en sintonía con la cadena de sílabas. Se presenta la codificación del método pytts, la voz que se utilizó fue la de Juan de la página de Loquendo.

```
# se hace uso de la librería Pytts
import pyTTS
#Se crea el objeto para hacer uso de la voz
tts = pyTTS.Create()
tts.Rate = 1
#print "Speech rate =", tts.Rate
# set the speech volume percentage (0-100%)
tts.Volume = 90
#print "Speech volume =", tts.Volume
print "List of voices =", tts.GetVoiceNames() #Se manda a imprimir las
listas de voces disponibles
# explicitly set a voice
tts.SetVoiceByName('LQJuan') #Se elige que tipo de voz se utilizara
```

Se presenta la codificación de la clase VisemeEvents, dicha clase ejecutara los eventos del método tts.speak, esta clase tendrá la función OnPhoneme y OnEndStream. Esta función recibe como parámetros el numero de fonema, posición, duración, el siguiente fonema y el identificador de fonema constituidos en una misma sílaba.

La segunda función OnEndStream lo que hará es finalizar la longitud de una cadena de fonemas. Cada vez que se active un fonema con el método tts.speak se mandará a llamar a la función visualizar visema y esta cargará la imagen a animar; todo esto se realizará en fracción de segundos. A continuación se presenta la codificación en el lenguaje de programación Python la clase VisemeEvents.


```

finished = False
class VisemeEvents:
    def OnPhoneme(self, StreamNumber, StreamPosition, Duration,
        NextPhoneId, Feature, CurrentPhoneId):
        global procesada
        global cargar_actor, mostrar_texto, textura_normal, textura_a,
            textura_b, textura_ch, textura_d, textura_e, textura_f
        global textura_g, textura_i, textura_k, textura_l, textura_ll,
            textura_m, textura_n, textura_nn, textura_o, textura_p
        global textura_r, textura_rr, textura_s, textura_t, textura_u,
            textura_w, textura_x, textura_y
        global procesada_pos
        print "***ON PHONEME***: StreamNumber: " + str(StreamNumber)
+      ", StreamPosition: " + str(StreamPosition) + ", Duration: "
+      str(Duration) + ", NextPhoneId: " + str(NextPhoneId) + ",
        Feature: " + str(Feature) + ", CurrentPhoneId: " +
        str(CurrentPhoneId)
        if procesada[procesada_pos] == 10: #Fonema A
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_a, "Animar boca A") #Animar boca A

        elif procesada[procesada_pos] == 18: #Fonema b
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_b, "Animar boca B") #Animar boca B

        elif procesada[procesada_pos] == 21: #Fonema ch
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_ch, "Animar boca C") #Animar boca CH

        elif procesada[procesada_pos] == 7: #Fonema d
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_d, "Animar boca D") #Animar boca D

        elif procesada[procesada_pos] == 11: #Fonema E
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_e, "Animar boca E") #Animar boca E

        elif procesada[procesada_pos] == 23: #Fonema f
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_f, "Animar boca F") #Animar boca F

        elif procesada[procesada_pos] == 20: #Fonema g
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_g, "Animar boca G") #Animar boca G

        elif procesada[procesada_pos] == 12: #Fonema i
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_i, "Animar boca I") #Animar boca I

        elif procesada[procesada_pos] == 19: #Fonema k
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_k, "Animar boca K") #Animar boca K

        elif procesada[procesada_pos] == 29: #Fonema l
            visualizar_visema(cargar_actor, mostrar_texto,
                textura_l, "Animar boca L") #Animar boca L

```

```

elif procesada[procesada_pos] == 29: #Fonema ll
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_ll,"Animar boca LL") #Animar boca LL

elif procesada[procesada_pos] == 26: #Fonema m
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_m,"Animar boca M") #Animar boca M

elif procesada[procesada_pos] == 27: #Fonema n
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_n,"Animar boca N") #Animar boca N

elif procesada[procesada_pos] == 28: #Fonema ñ
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_nn,"Animar boca Ñ") #Animar boca ñ

elif procesada[procesada_pos] == 13: #Fonema o
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_o,"Animar boca O") #Animar boca O

elif procesada[procesada_pos] == 17: #Fonema p
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_p,"Animar boca P") #Animar boca P

elif procesada[procesada_pos] == 32: #Fonema r
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_r,"Animar boca R") #Animar boca r

elif procesada[procesada_pos] == 31: #Fonema rr
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_rr,"Animar boca RR") #Animar boca rr

elif procesada[procesada_pos] == 24: #Fonema s
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_s,"Animar boca S") #Animar boca s

elif procesada[procesada_pos] == 15: #Fonema t
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_t,"Animar boca T") #Animar boca t

elif procesada[procesada_pos] == 14: #Fonema u
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_u,"Animar boca U") #Animar boca u

elif procesada[procesada_pos] == 25: #Fonema x
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_x,"Animar boca X") #Animar boca x

elif procesada[procesada_pos] == 33: #Fonema y
    visualizar_visema(cargar_actor,mostrar_texto,
        textura_y,"Animar boca Y") #Animar boca y

```

```

        elif procesada[procesada_pos] == 34: #Fonema w
            visualizar_visema(cargar_actor,mostrar_texto,
                textura_w,"Animar boca W") #Animar boca w
    procesada_pos+=1
    return

def OnEndStream(StreamNumber, StreamPosition, StreamReleased):
    global finished
    global fin_palabra
    global procesada
    global procesada_pos
    procesada_pos = 0
    print fin_palabra
    global cargar_actor,mostrar_texto,textura_normal,textura_a,
        textura_b,textura_ch,textura_d,textura_e,textura_f
    global textura_g,textura_i,textura_k,textura_l,textura_ll,
        textura_m,textura_n,textura_nn,textura_o,textura_p
    global textura_r,textura_rr,textura_s,textura_t,textura_u,
        textura_w,textura_x,textura_y
    if fin_palabra:
        visualizar_visema(cargar_actor,mostrar_texto,textura_normal,
            "boca normal")
    finished = True
    return

```

Se muestra en la tabla 31, los identificadores de los fonemas, correspondientes al punto de articulación que se ha de mostrar, y los visemas mas representativos de cada punto de articulación.












Identificador	Punto de Articulación	Fonemas	Id_TTS CurrentPhoneId	Visemas
'C'	Central	/a/	/10/	
'B'	Bilabial	/b/	/18/	
		/m/	/26/	
		/p/	/17/	
'P'	Palatal	/ch/	/21/	
		/L/	/7/	
		/ñ/	/28/	
		/y/	/33/	
'D'	Dental	/d/	/16/	
		/t/	/15/	
'I'	Anterior-Media	/e/	/11/	
'L'	Labiodental	/f/	/23/	
'V'	Velar	/g/	/20/	
		/k/	/19/	
		/x/	/25/	
		/w/	/34/	
'N'	Anterior-Cerrada	/i/	/12/	
'A'	Alveolar	/l/	/29/	
		/n/	/27/	
		/r/	/31/	
		/R/	/32/	
		/s/	/24/	
'F'	Posterior-Media	/o/	/13/	
'O'	Posterior-Cerrada	/u/	/14/	

Tabla 31. Clasificación del punto de articulación y su relación con el visema adecuado.

Anexos E

Publicaciones

J. Radilla Avila, M.L. Morales Rodríguez, A. Hernández Ramírez, J. Ramirez Saldivar, J. Gonzalez Barbosa, H. Fraire-Huacuja “Silabeo Automático para la Generación de Vi-Sílabas en Español” presentado en el 17th International Congress on Computer Science Research, en Morelia, Michoacán, México, 2011; una versión extendida de éste artículo fue redactado en el idioma inglés ya que fue preseleccionado para su posible publicación en una edición especial de la revista *IJCOPI*.

J. Radilla Avila, M.L. Morales Rodríguez, A. Hernández Ramírez, J. Ramirez Saldivar, J. Gonzalez Barbosa “Generación Automaticas de Vi-Sílabas en el idioma Español” presentado en el V encuentro de Investigadores desarrollado por el Instituto Tecnológico de cd. Madero; en Cd. Madero, Tamaulipas, México, 2011.

Glosario

Acento:	Es la mayor intensidad con la que se pronuncia una sílaba dentro de una palabra.
Animación:	Es un proceso utilizado para dar la sensación de movimiento a imágenes o dibujos.
Alfabeto:	Es el conjunto ordenado de sus letras; la agrupación, con un orden determinado, de las grafías utilizadas para representar el lenguaje que sirve de sistema de comunicación.
Alófono:	En fonética, se llama alófono a cada uno de los fonos o sonidos que en un idioma dado se reconoce como un determinado fonema, sin que las variaciones entre ellos tengan valor diferenciativo.
Articulación:	Se refiere a la forma en que se produce la transición de un sonido a otro, o sobre la misma nota.
Atónas:	Se denominan átonas a aquellas sílabas sobre las que no recae el acento prosódico de la palabra, es decir, a aquellas distintas a la tónica.
Consonantes:	Una consonante es el sonido de la lengua oral que se produce por el cierre o estrechamiento del tracto vocal al acercar o poner en contacto los órganos ligados con la articulación provocando una turbulencia audible.
Corpus:	Conjunto lo más extenso y ordenado posible de datos que pueden servir de base a una investigación.
Corpus lingüístico:	Es un conjunto, normalmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (típicamente), o muestras orales (normalmente transcritas).
Diptongo:	Es la unión de dos vocales una fuerte y una débil ó dos débiles en la misma sílaba que se pronuncian en un solo golpe de voz.
Elisión:	Supresión de la vocal final de una palabra cuando la siguiente comienza por vocal: en la contracción "al", por "a el", ha habido elisión de la vocal del determinante.
Fonema:	Son unidades naturales que dotan de gran flexibilidad a los sistemas de voz y que resultan económicas desde el punto de vista del número

de unidades (en el Español existen 23 fonemas); sin embargo, constituyen una unidad abstracta que está sometida a muchas variaciones contextuales, que originan una baja calidad en la voz sintetizada.

- Fonemización:** La operación de asociar un sonido con su fonema distintivo.
- Fonética:** Conjunto y estudio de los sonidos de un idioma.
- Fonología:** Subcampo de la lingüística. Mientras que la fonética estudia la naturaleza acústica y fisiológica de los sonidos o alófonos, la fonología describe el modo en que los sonidos funcionan (en una lengua en particular o en las lenguas en general) en un nivel abstracto o mental.
- Frase:** Cadena de palabras conectadas sintácticamente y gramaticalmente.
- Grafema:** En Lingüística, mínima unidad significativa en el plano de la lengua escrita.
- Gramática:** Es el estudio de las reglas y principios que regulan el uso de las lenguas y la organización de las palabras dentro de una oración. También se denomina así al conjunto de reglas y principios que gobiernan el uso de un lenguaje muy determinado; así, cada lenguaje tiene su propia gramática.
- Hiato:** Es la separación de dos vocales que no forman diptongo y que, por tanto, pertenecen a dos sílabas distintas.
- Idioma:** Sistema de signos orales y escritos que utiliza una comunidad de hablantes para comunicarse.
- Lengua:** Es un sistema de signos orales y escritos que son utilizados por los seres humanos para comunicarse entre sí en determinada región o país. La lengua como código de comunicación es un instrumento común que el hablante debe utilizar correctamente para facilitar su comprensión.
- Lenguaje:** Conjunto de sonidos y palabras con que se expresa el pensamiento. Toda forma de comunicar el pensamiento.
- Léxico:** Conjunto de palabras y expresiones propias de una región, de una actividad, de un periodo determinado, o de una persona.
- Lingüística:** Es una disciplina cuyo objeto de estudio es el lenguaje humano. Se trata de una ciencia teórica dado que formula explicaciones diseñadas para justificar los fenómenos del lenguaje, esto es, el diseño de teorías sobre algunos aspectos del lenguaje y una teoría general del mismo.
- Modelo Visual:** Contenedor de la información visual esencial que se extrae a partir del corpus audiovisual.
- Oración:** Unidad gramatical que es sintácticamente independiente y tiene un sujeto expreso o tácito y un predicado que contiene al menos un verbo conjugado.

Palabras:	Cada uno de los segmentos limitados por pausas o espacios en la cadena hablada o escrita, que puede aparecer en otras posiciones, y que está dotado de una función.
Panda3D:	Es un motor 3D enfocado a la creación de videojuegos en PC. También es considerada como una biblioteca de funciones escritas en C++ que sirven para simplificar enormemente la creación de videojuegos 3D.
Personaje Virtual:	Son actores animados representados por medio de una computadora que coexisten dentro de un universo virtual que podrían interactuar con el usuario.
Python:	Es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas Web.
Pytts:	El módulo pyTTS es la librería para manejar los servicios Text-to-Speech de Microsoft Speech API (SAPI) para su uso en Python. Se basa en la biblioteca win32com para la obtención y la comunicación con las interfaces COM SAPI.
Prosodia:	Es una rama de la lingüística que analiza y representa formalmente aquellos elementos de la expresión oral, tales como el acento, los tonos y la entonación.
Pronunciación:	Es la manera en que una palabra o idioma es hablada; ó el modo en que alguien pronuncia una palabra.
Sincronización:	Se refiere a que dos o más elementos, eventos u operaciones sean programadas para que ocurran en un momento predefinido de tiempo o lugar.
Sílabas:	Cada una de las divisiones fonológicas en las que se divide una palabra.
Signo:	En Lingüística, cualquier unidad lingüística que posee significación, inclusive los radicales o los afijos.
Silabificación:	Pronunciación lenta y pausada de las palabras, de forma que se distingan sus sílabas.
Sonidos:	Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico, como el aire.
Tónicas:	Es aquella en donde se carga la voz. Es decir, la sílaba de una palabra que es pronunciada con mayor intensidad que las demás.

Transcripción fonética:	También llamada notación fonética es un sistema de símbolos gráficos para representar los sonidos del habla de una persona. Típicamente se usa como convención para superar las peculiaridades alfabéticas usadas en cada lengua escrita y también para representar lenguas sin tradición escrita.
Trivisemas:	Visemas correspondientes a una serie de tres alófonos consecutivos.
Triptongo:	El conjunto de tres vocales que se pronuncian en una misma sílaba. Están formados por una vocal abierta (a, e, o) que ocupa la posición intermedia entre dos vocales cerradas (i, u), ninguna de las cuales puede ser tónica".
Visema:	Representación visual de un fonema, visualmente distinguible de otras.
Visemas clave:	Visemas asociados a conjuntos de alófonos de apariencia visual similar a partir de los cuales se generan transiciones entre ellos. Son las apariencias clave de las regiones orofaciales del modelo visual.
Vi-sílabas:	El término vi-sílaba fue llamado así por Kshirsagar en su trabajo "Visyllable Based Speech Animation" y se puede definir como la asociación de un visema a una sílaba.
Vocales:	Es un sonido de una lengua natural hablada que se pronuncia con el tracto vocal abierto, no habiendo un aumento de la presión del aire en ningún punto más arriba de la glotis.

Referencias Bibliográficas

Abadía, P. M., Fernández, F. (2010), “Manual del productor audiovisual” Ed., UOC, 2010, pp. 437.

Alvar, M., 2000, “Introducción a la lingüística española Ariel lingüística” Ed., Ariel, pp. 636.

Bailenson, J. N. y Blascovich, J., 2004, “Avatars,” in *Encyclopedia of Human-Computer Interaction*: G. Barrington, MA: Berkshire Publishing Group, pp. 64–68.

Cabiedes F. P. et. al., 2006, Bilbao 20-23 de junio, 2006, Laboratorio de Interacción Humano Máquina y Multimedia, Centro de Ciencias Aplicadas y Desarrollo Tecnológico, UNAM.

Cabiedes, F., et al., 2007, “Sincronización de labios: Métodos sin visemas” *Universidad Nacional Autónoma de México, y Dirección de Investigación e Innovación-Enciclomedia, ILCE(México), RIED. Revista Iberoamericana de Educación a Distancia, volumen 10, n° 1, pp. 37-50. México D.F, 2007.*

Ceballos, A. et al., 2009, “Seguimiento del contorno externo de la boca en imágenes de video” *Revista de ingenierías Universidad de Medellín, Colombia, Vol. 8, Núm. 14, pp. 129-144, enero-junio.*

Cletus. F., 1968, “Confusions among visually perceived consonants”. *Journal of Speech and Hearing Research*, vol. 11, pp. 796–804, Diciembre, 1968.

Curk, T., et al., 2005. *Microarray data mining with visual programming*. *Bioinformatics*. 2005 Feb 1;21(3):396-8. online- <http://orange.biolab.si/>

De la Mota Gorriz, C., y Ríos Mestre, A. 1995. Problemas en torno a la transcripción fonética del español: los alfabetos fonéticos propuestos por IPA y RFE y su aplicación a un sistema automático. *Acta Universitatis Wratislaviensis* N° 1660. Estudios Hispánicos IV, Universidad de Wroclaw, pp.97-109.

Engström, C., 2003, *Articulatory Analysis of Swedish Visemes*. Centrum för talteknologi. Institutionen för tal, musik och hörsel. Stockholm

Ferreira, J. F. y Fernando de la Rosa, 2003., “Una propuesta de integración de animación facial y voz sintética” *Universidad de Los Andes, Departamento de Sistemas y Computación, Bogotá D.C., Colombia*.

Gil Fernández, J., 2007 “Fonética para profesores de español: de la teoría a la práctica” Ed., Arco/Libros, 2007, pp. 614.

Iribarren, I. C., 2005 “Ortografía española: bases históricas, lingüísticas y cognitivas” Ed., El Nacional, pp.106.

Kshirsagar, Nadia, S., 2003, “Visyllable Based Speech Animation”, *University of Geneva, Switzerland*, Volume 22, Number 3.

Lango, K. 2006, Center for Advanced Technologies de la Universidad de Nueva York y profesor en la Universidad ORT del Uruguay. “Principios para la animación”, Septiembre 2006. [Online]. Disponible <http://www.escardo.com/charanim/?p=33>.

Melenchon. Maldonado, J., 2007 “Síntesis Audiovisual Realista Personalizable”, Tesis Doctoral, Dpto. de Comunicación y teoría del Señal, Centro de Ingeniería y Arquitectura de la Salle, Barcelona, España.

Mestre, Ríos. A. (1999) “Estudios de Lingüística del Español, La transcripción Fonética Automática del Diccionario Electrónico de Formas Simples Flexivas del Español: Estudio Fonológico en el Léxico”, Dpto. de Filología Española.

Mozas, A. B., 1996, Gramática práctica Editor EDAF, ISBN8476405960, 9788476405963 N. ° de páginas 330 páginas.

Núñez, R. et al., 1999 “Fonología generativa contemporánea de la lengua española Edición ilustrada Editor Georgetown University Press, 1999 ISBN0878406948, 9780878406944 N. ° de páginas 294 páginas.

Obediente, E., 1998, “*Fonética y fonología Colección Ciencias sociales: Serie Lingüística*” Ed., Universidad Los Andes, pp. 432.

Poch-Olivé, D., 1999, Fonética para aprender español: pronunciación Colección E. Serie Estudios Serie estudios Edición ilustrada Editor Editorial Edinumen, N. ° de páginas 110 páginas.

Robotham, X., 2011 A. M., Apuntes: Articulación de las letras del abecedario Español, Universidad Metropolitana de Ciencias de la Educación. On-line <http://www.umce.cl/facultades/filosofia/basica/docentes.html#xandre> (fecha de consulta 30 de Septiembre del 2011).

Zoric´ G., e Pandzic´, I. S., 2006 “*Real-time language independent lip synchronization method using a genetic algorithm*” *Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000 Zagreb, Signal Processing 86, pp. 3644-3656, Croatia.*