

Capítulo IV Diseño del Sistema

4.1 Creación del sistema.

Para el desarrollo de nuestro sistema nos basamos en el desarrollo del proyecto anterior, el cual está diseñado sobre la tecnología JAVA, para el nuestro seguimos la misma metodología del sintetizador anterior para la creación del nuestro.

El proyecto que realizamos consta a comparación del anterior de una base de datos, una aplicación en Visual Basic, la cual nos sirve para la importación de los corpus de voz que vamos a utilizar.

4.1.1 Método Utilizado

El método Usado para la creación del sintetizador, está basado en la mejor selección de la unidad a concatenar, como lo realiza UNIT SELECTION, nuestro trabajo se basa en el mejoramiento de la calidad del sintetizador usando este método. Ya que este proceso estaba iniciado, nos dimos a la tarea de implementar una estructura de búsqueda, y que a su vez pudiera generar una lista de listas de los mejores candidatos a concatenar. Dentro de la estructura de búsqueda, se logra agilizar el acceso al corpus de voz por medio de la base de datos, usando las ventajas de la búsqueda por medio de la base de datos.

4.2 Instalación de Java 2.

El Java 2 SDK es un entorno para el desarrollo de las aplicaciones, tales como applets, servlets, frames, y demás componentes usando el lenguaje de programación JAVA . en la actualidad existen varias versiones del JSDK disponibles, pero para el desarrollo de nuestro sistema, utilizaremos la versión 1.4.2, debido a que es una versión estable, y reciente, la cual la podremos encontrar en <http://www.java.sun.com/j2se/1.4.2/download.html>

Para poder verificar cuales son las plataformas que soporta JAVA, es necesario consultar la pagina <http://www.java.sun.com/j2se/1.4.2/systemconfigurations.html>. Y una vez que se obtiene la información adecuada para la plataforma a utilizar, se deben verificar los requerimientos mínimos para poder realizar una adecuada instalación.

En nuestro caso se instalara en una PC, para lo cual tenemos que contar con los siguientes requerimientos: procesador Intel o 100% compatible, Pentium I a 166 MHz. ó superior, 32 Mb de memoria RAM ó superior y 70 Mb de espacio libre en disco [Sun Microsystems] [Adams, 2004].

Una vez que tenemos instalado el J2SDK es necesario indicar por medio de las variables de entorno, la ubicación de las librerías necesarias, para realizar lo anterior se crea una variable de entorno llamada CLASSPATH como se demuestra en la siguiente figura:

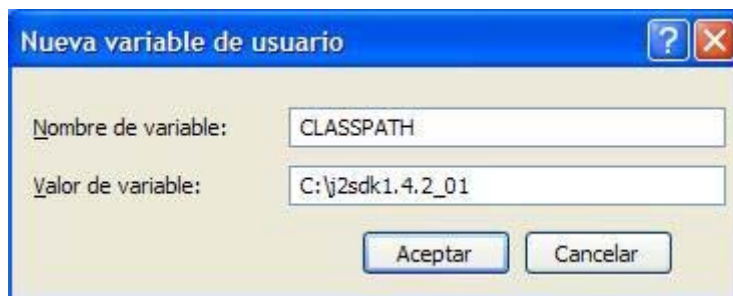


Fig. 4.1 Classpath para J2SDK

De la misma manera se le debe indicar al sistema operativo la ubicación de los archivos ejecutables de Java, como se muestra a continuación:

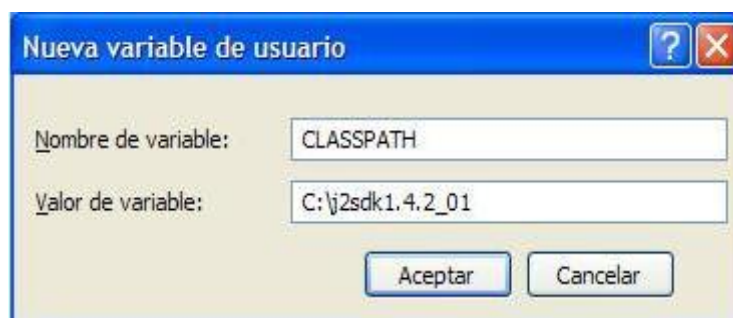
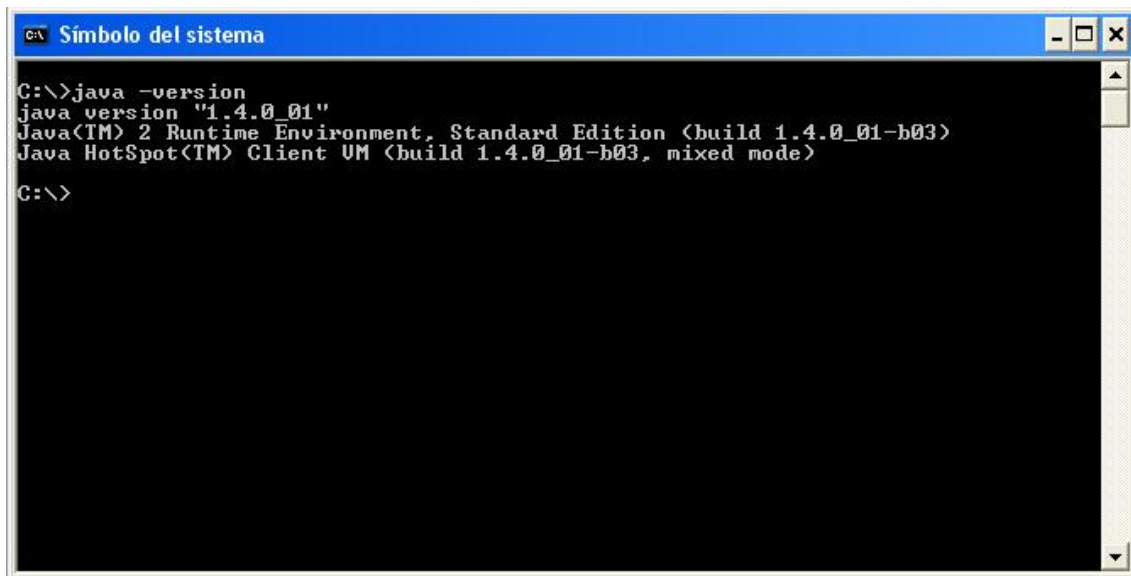


Fig. 4.2 Path para J2SDK

Para finalizar, se debe de reiniciar el equipo para que los nuevos cambios tengan efecto. Posteriormente, se abre una consola de MS-DOS y se ejecuta el comando *java -versión* para verificar que todo se haya instalado correctamente, si es así se debe desplegar una pantalla como el siguiente:



```
C:\>java -version
java version "1.4.0_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.0_01-b03)
Java HotSpot(TM) Client VM (build 1.4.0_01-b03, mixed mode)
C:\>
```

Fig. 4.3 Comprobación de instalación del J2SDK

Una vez instalado correctamente el compilador se procede a la instalación del mismo.

4.2.1 La Base de Datos

La base de datos creada, contiene toda la información necesaria sobre los corpus a usar, de manera que esto nos facilitará de manera considerable el trabajo. Para ello es necesario la instalación y configuración de MySQL.

4.2.2 Instalación de MySQL

MySQL es un manejador de bases de datos relacionales de código abierto, durante los últimos años ha tomado fuerte importancia en las aplicaciones web, debido a su rapidez de procesamiento, confiabilidad, flexibilidad y principalmente, a que el acceso a este software es gratuito. Este manejador de base de datos utiliza el lenguaje de consultas estructurado SQL para realizar las consultas a las bases de datos [Maslakowski, et al., 2001] [MySQL Site, 2004].

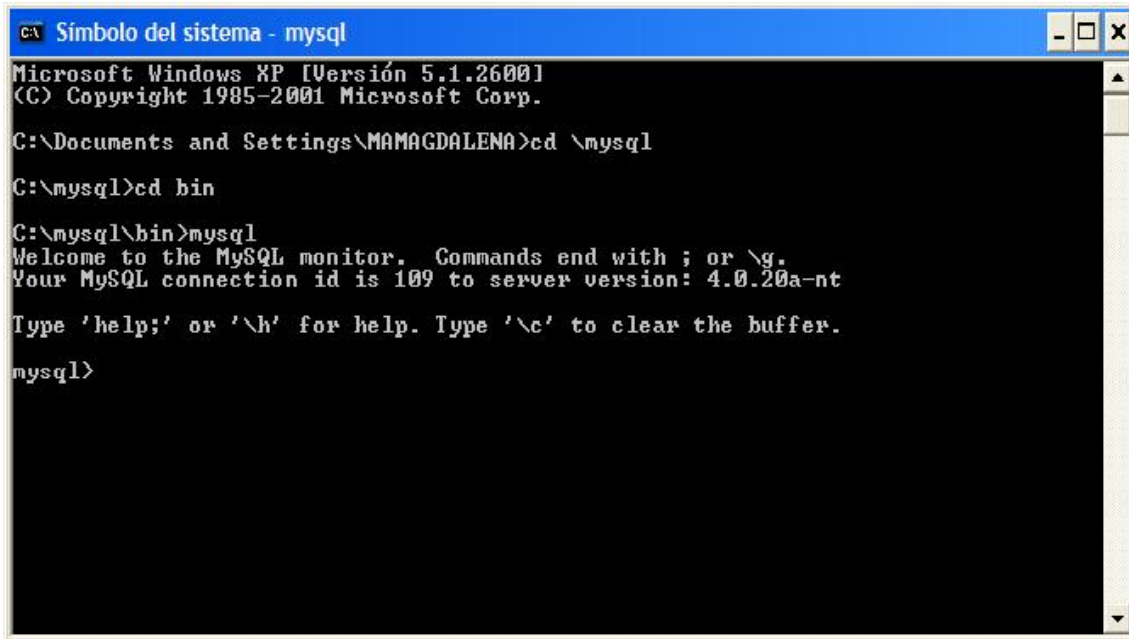
Ya que es de código abierto y gratuito, es decir, se puede obtener de la página de Internet oficial del sitio, existen versiones disponibles, la más reciente es la 5, pero para nuestro caso usaremos la versión más estable que es la 4.0.16. Para tener acceso a esta y a cualquier información sobre MySQL se obtiene ingresando a la página <http://www.mysql.com/download>.

MySQL contiene al igual que la mayoría de los sistemas de administración de bases de datos relacionales, una función que es como un servicio o daemon, que es un programa que se ejecuta continuamente en segundo plano, en el caso de MySQL se puede ver su funcionamiento o ejecución en Windows dentro del administrador de tareas de tareas, específicamente en la barra inferior derecha, como se muestra a continuación:



Fig. 4.4 Verificación de la instalación de MySQL

Con esto marcamos el punto final de la instalación de MySQL en la computadora, para identificar que la instalación realizada, estuvo correctamente, se debe de ejecutar una Terminal de MS-DOS y cambiarse al directorio bin de la instalación y ejecutar el script mysql. Si todo es correcto, deberá desplegarse algo como lo siguiente:



```
ca Símbolo del sistema - mysql
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\MAMAGDALENA>cd \mysql
C:\mysql>cd bin
C:\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 109 to server version: 4.0.20a-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Fig. 4.5 Verificación de ejecución de MySQL

Una vez que se haya verificado la instalación de MySQL, es necesario que se dar de alta una variable mas, esta variable es considerada de entorno, y nos sirve para darle a conocer al compilador de JAVA la ubicación del controlador que necesita el Java para poder comunicarse con MySQL, la variable de entorno se declara siguiendo las siguiente manera:

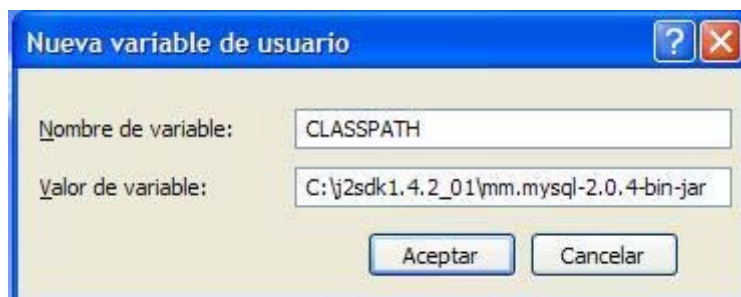


Fig. 4.6 Classpath para el controlador de MySQL

4.3 El Sistema

Para el mejoramiento del sistema actual, fue necesario en primera instancia, leer bien el código y entenderlo, saber que clases eran las más importantes y ver su desempeño dentro del sistema.

Una vez que se ha realizado esta acción, se limpio el código, y se empezó a rediseñar el sistema con las siguientes clases, las cuales algunos de sus métodos cambiaron, para el mejoramiento del mismo, las modificaciones de las clases ayudan a mejorar la calidad del sistema.

A continuación se describen las clases más importantes de nuestro sistema:



Fig. 4.7 Diagrama de la Clase Principal Unit Selection

La clase Unit Selection , es la clase principal de nuestro sistema, es la que ejecuta y realiza las peticiones del usuario, en esta clase se modificó en primera instancia el frame, que es la interfaz gráfica la cual se despliega al correr el programa, aquí el usuario ingresa la frase que desea sintetizar.



Fig. 4.8 Diagrama de la Clase CorpusDb

La clase CorpusDB es la que nos conectaron nuestra base de datos, la cual se explicará en el siguiente capítulo, aquí es donde se encuentran nuestros corpus de voz, que ya fue pregrabado. La clase tiene la función de acceder a dicha base de datos, ya sea para localizar las palabras, frases y fonemas, que sean necesarios para la síntesis de la frase proporcionada. Se encarga de hacer la búsqueda en la base de datos y generación de estructuras y clases necesarias para la aplicación.

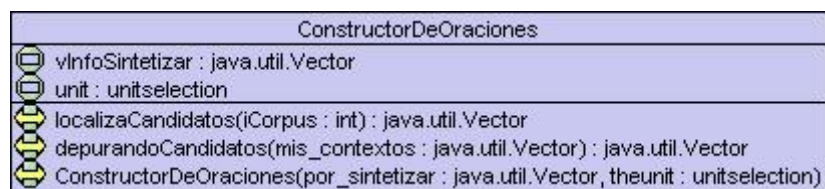


Fig. 4.9 Diagrama de la Clase Constructor de Oraciones

La clase Constructor de Oraciones, es aquella que proporciona la información adecuada para la generación de las oraciones a procesar, esta clase realiza la localización de los candidatos para ponerlos dentro de un vector, y a su vez este es depurado para la creación de las oraciones. Obtener los mejores candidatos para la síntesis.

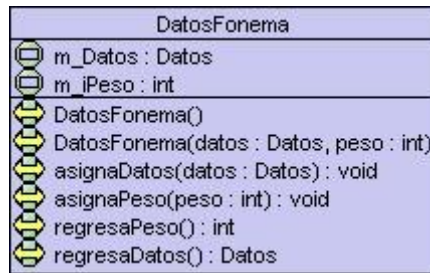


Fig. 4.10 Diagrama de la Clase Datos Fonema

Esta clase proporciona los datos de los fonemas existentes para ser agregados a la información a sintetizar. Cuando ya se tiene el texto proporcionado y se necesita de los fonemas para formar alguna palabra, esta clase nos proporciona dicha información de cada fonema.



Fig. 4.11 Diagrama de la Clase Fonema Candidato.

Maneja la Estructura de los candidatos de los fonemas, es decir, cuando tenemos una información a sintetizar, se proporcionan los fonemas, esta clase se encarga de que este fonema sea el adecuado y se añade a una lista para ser concatenado.

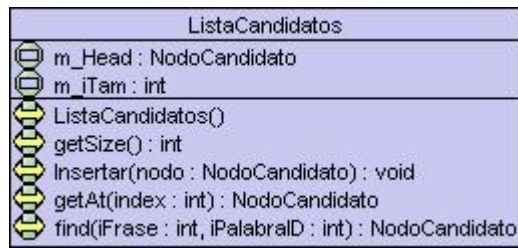


Fig. 4.12 Diagrama de la Clase ListaCandidatos

La clase Lista Candidatos es la que maneja los candidatos de los fonemas, una vez que se hayan sido seleccionados para su concatenación.

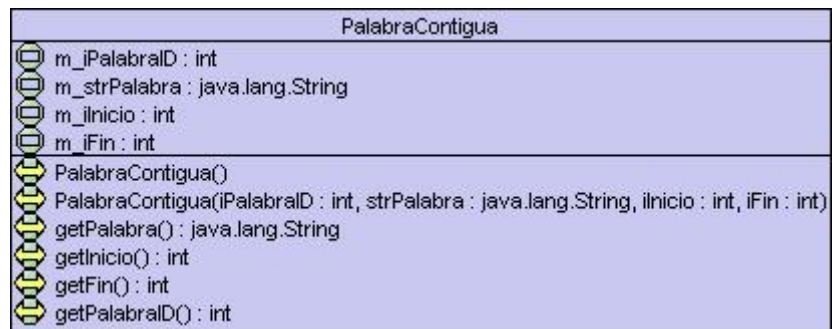


Fig. 4.13 Diagrama de la Clase Palabra Contigua

La clase Palabra Contigua maneja la información de las palabras, en cuanto a su identificación dentro del corpus, donde comienza, donde termina y toda esta información esta compuesta dentro de una cadena. El inicio y fin se refiere al milisegundo donde empieza y termina cada palabra.

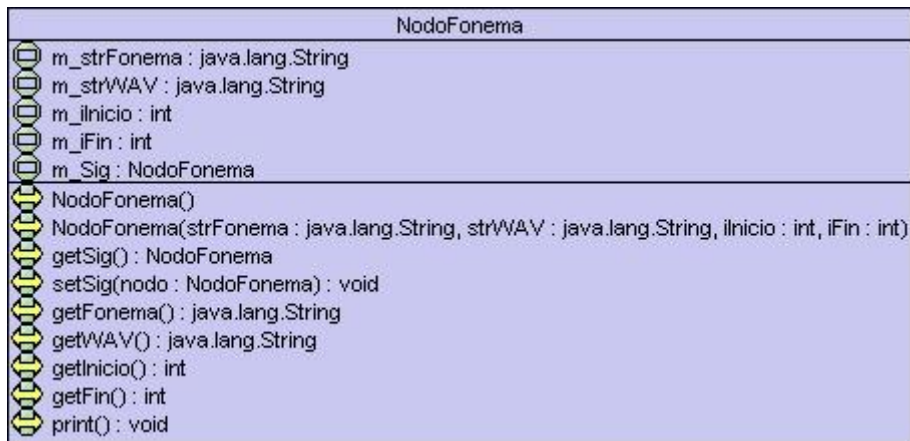


Fig. 4.14 Diagrama de la clase Nodo Fonema

La clase nodo fonema es la encargada de procesar la información de los archivos .wav de los candidatos a ser concatenados, contiene también un inicio y un fin, esto es para saber donde comienza y donde termina.



Fig. 4.15 Diagrama de la clase Nodo Candidato

La clase Nodo Candidato, contiene información de las palabras que son añadidas, al este, como lo es la anterior, siguiente, y la contiene a su vez información de los wav`s que contiene esta información.

4.4 Dependencia de Clases

La dependencia de clases nos muestra por medio de los diagramas comúnmente llamados de UML, la relación que mantienen las clases de nuestro sistema como se muestra en el apéndice B.

4.5 Diagramas de casos de Uso

Por medio de este diagrama se describen los procesos del sistema así como la interacción de los procesos de usuarios y de sistemas externos, como se muestra a continuación en el siguiente diagrama.

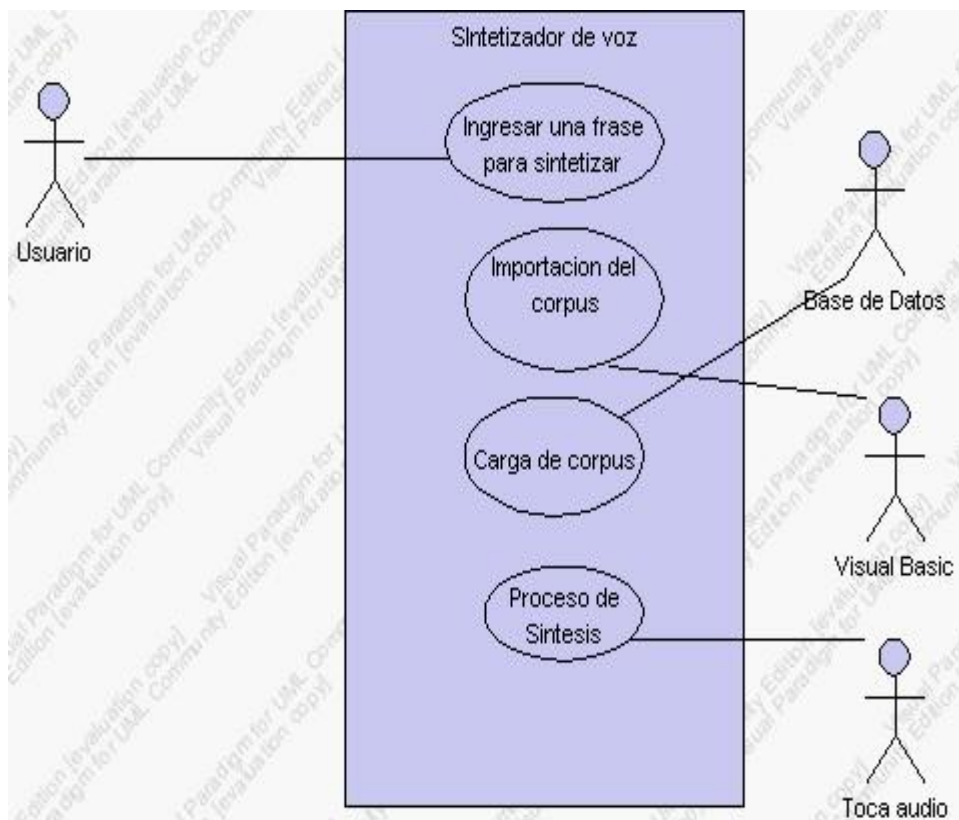


Fig. 4.16 Diagrama de Caso de uso

4.6 Diagramas de Secuencia

Los diagramas de secuencia, muestran la forma en como los usuarios se comunican con las clases del sistema, esto es, el seguimiento que hace el sistema para resolver las peticiones que el usuario ha determinado.

Los Diagramas de secuencia los podemos obtener de dos maneras distintas, por medio del diagrama de casos de uso, o por medio de los diagramas de clases, en este caso el diagrama de secuencia fue realizado por medio de los diagramas de clase como se muestra a continuación.

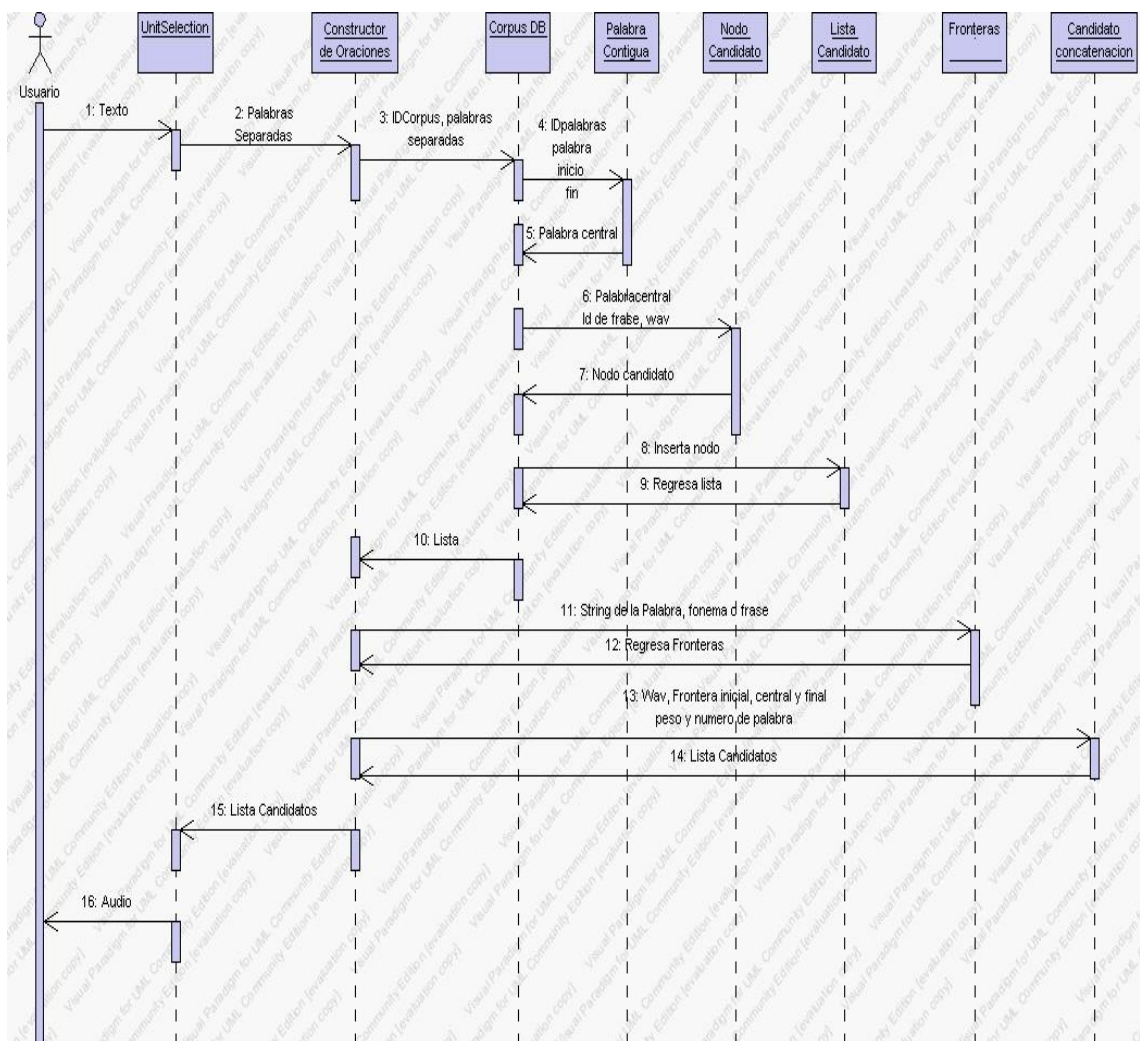


Fig. 4.17 Diagrama de Secuencia

4.7 Resumen

Dentro de este capítulo se vio como es que se relacionan las clases mas importantes de nuestro sistema, dando una breve explicación de su funcionamiento, así como el desarrollo del sistema y creación de nuestro sintetizador.

Por medio de los diagramas de UML podemos ver la forma en que están constituidas las clases, dados los diagramas de clases, con los diagramas de secuencia podemos ver la manera en la que interactúan o se relacionan las clases para atender las peticiones del usuario. Los casos de uso nos muestran las diferentes opciones que tiene el usuario para acceder al sistema.