Capítulo 3: IMPLEMENTACIÓN DEL TUTOR DE LECTO-ESCRITURA EN ESPAÑOL.

.

Este capitulo tratará el análisis y diseño de los módulos del sistema de Tutor de lectura, pero más específicamente lo módulos que se consideraron para el desarrollo del tutor de lecto-escritura del español. Cabe mencionar que hasta ahora no se cuenta con documentación de este sistema, por lo cual se realizó un proceso de ingeniería inversa para recuperar parte del diseño y especificaciones del sistema de tutores.

3.1 Análisis y Diseño del SW del CSLR

3.1.1 Casos de uso

A continuación se presentan los diagramas de casos de uso para las acciones del sistema que están relacionadas con los tutores y el tutor FourSquare.

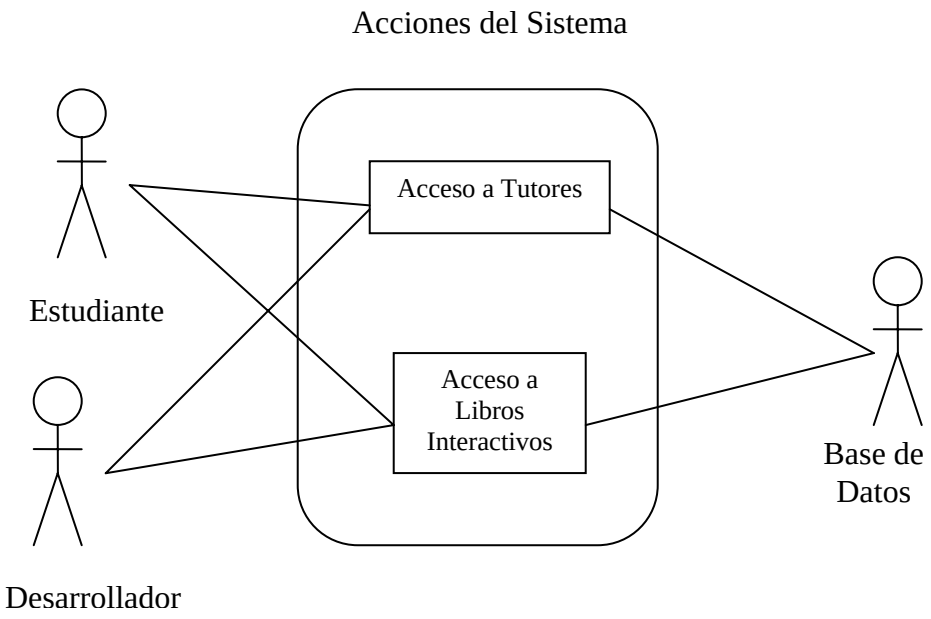

Figura 9. Caso de uso de los tutores del CSLR.

En el caso de uso de la Figura 9, se muestra que al acceder al sistema tanto el estudiante como el desarrollador pueden tener acceso a todos los tutores que estén disponibles en el sistema, incluyendo al tutor FourSquare y a los libros interactivos, estos a su vez se relacionan con la base de datos que es la contiene los datos de cada tutor o libro interactivo. En los diagramas que presentamos a continuación, se mostraran los casos de uso para el tutor de FourSquare.

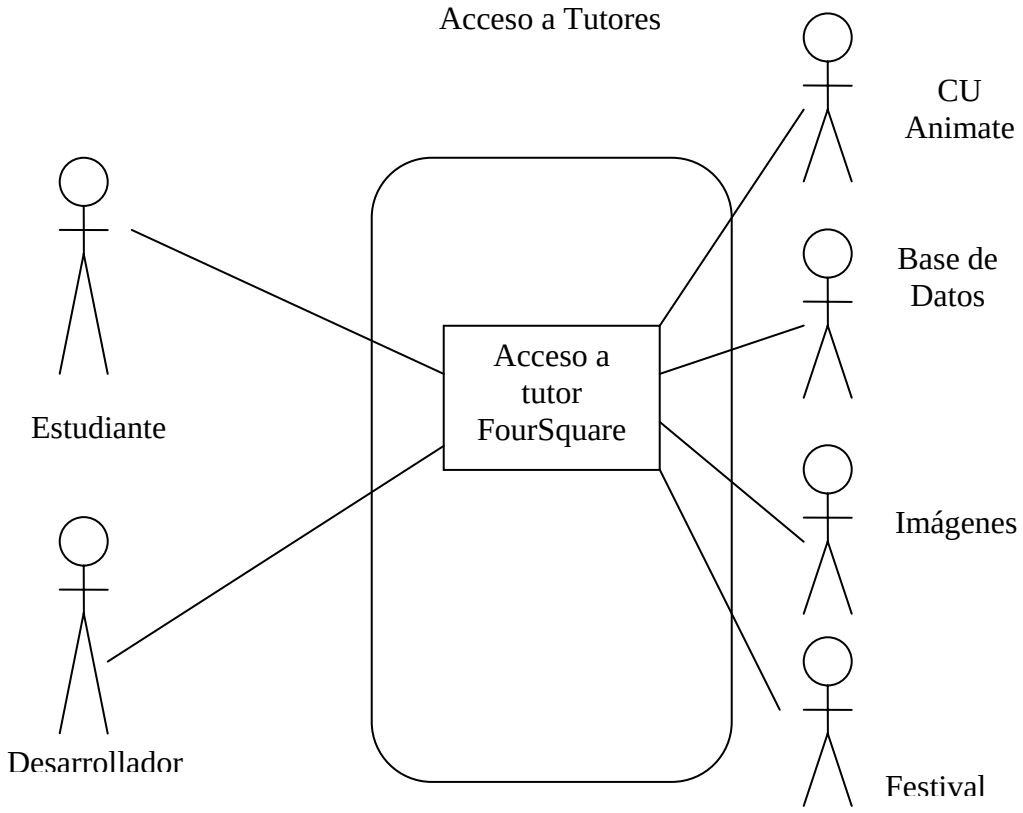

Figura 10. Casos de Uso para acceso a los tutores.

En el caso de uso de la figura 10, muestra que entre los tutores disponibles en el sistema está FourSquare, al cual puede tener acceso tanto el estudiante como el desarrollador. El tutor de FourSquare esta a su vez relacionado con sistemas como CUAnimate, que es el sistema encargado de la animación; con Festival que es el sistema

encargado de la producción de la voz; con la base de datos que contiene la información necesaria de los niveles que estudiará el alumno. También tiene acceso a un fólder de imágenes que son las que se muestran en la interfaz del tutor.

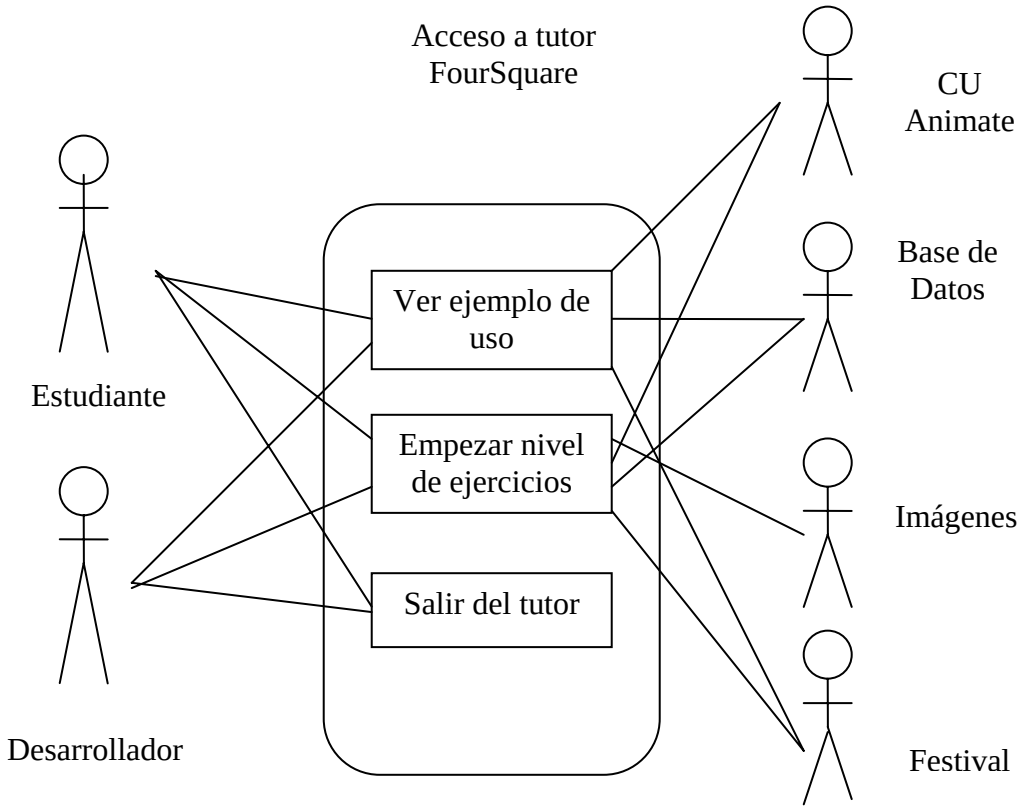

Figura 11. Caso de Uso para FourSquare

La figura once muestra el caso de uso para FourSquare. Al accesar a este tutor, existen tres opciones. La primera consiste en ver un ejemplo de uso, es decir, el tutor muestra un caso ejemplo para que el estudiante se oriente con respecto al manejo del tutor. La segunda opción consiste en comenzar con el primer nivel de ejercicios, para lo cual accesa a la base de datos y obtiene las palabras que corresponden a dicho nivel, también

accesa a CUAnimate y Festival. La tercera opción es salir del tutor, que como su nombre lo describe tiene la función de salir totalmente del tutor y regresar al ambiente del CSLR.

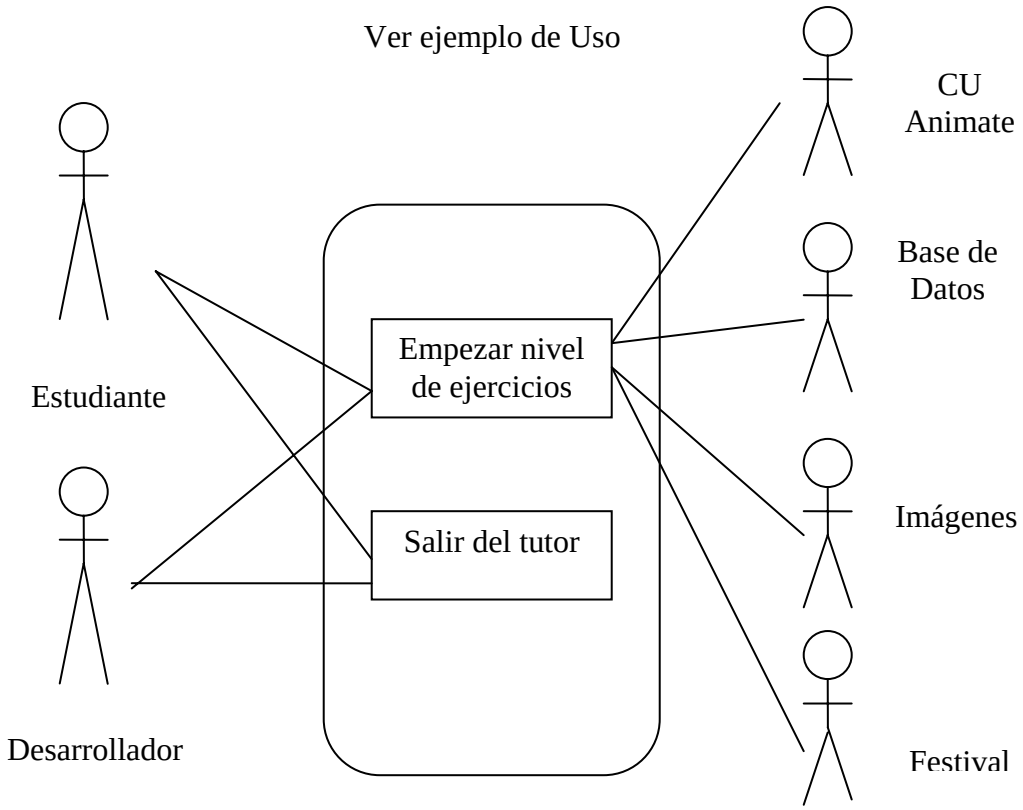

Figura 12. Caso de uso de la acción "Ver ejemplo de uso"

En el caso de uso de la figura doce, podemos ver las acciones que corresponden al caso de "ver ejemplo de uso". Esta opción, como se mencionó anteriormente, es la encargada de proporcionar al alumno un ejemplo del manejo del tutor, así el manejo del tutor para el estudiante será mucho más sencillo. Mientras muestra el ejemplo, el tutor habilita las opciones de "empezar nivel de ejercicios" y "salir de tutor". Al terminar la demostración, el tutor comienza de forma predeterminada con el nivel de ejercicios.

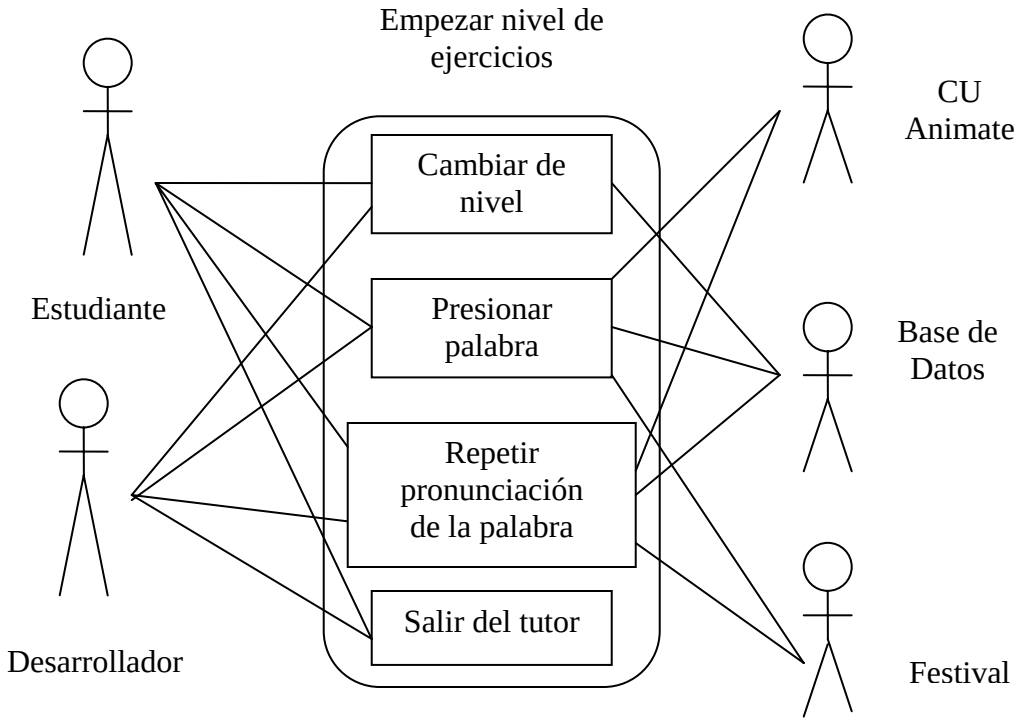

Figura 13. Caso de Uso de la acción "Empezar nivel de ejercicios".

En el caso de uso de la figura trece, se muestran las acciones que puede ejercer el usuario cuando comienza el nivel de ejercicios. Estas acciones pueden ser: cambiar de nivel, presionar una palabra, repetir la pronunciación de la palabra o salir del tutor. Al cambiar de nivel, el sistema accesa a la base de datos para obtener los datos del siguiente nivel de ejercicios. Al presionar una palabra de entre las cuatro opciones posible, el sistema accesa a la base de datos dado que ésta también contiene palabras que ya han sido previamente grabadas. Así si no existe la palabra grabada en la base de datos, Festival la construye. También se tiene acceso a CUAnimate ya que el agente debe de presentar los movimientos faciales correspondientes a la palabra que eligió el estudiante.

Al presionar la acción de “Repetir pronunciación de la palabra” el sistema accesa a la base de datos para buscar la grabación de la palabra que el tutor le está pidiendo al alumno que encuentre, si no la encuentra en la base de datos, la acción llama a Festival para que construya la palabra. Finalmente CUAnimate en base a los visemas, proporciona el movimiento de labios correspondiente a dicha palabra.

Finalmente la última acción a la que puede acceder el usuario es como lo mencionamos anteriormente, salir del tutor y regresar al ambiente del sistema de CSLR.

3.1.2 Diagramas de secuencia

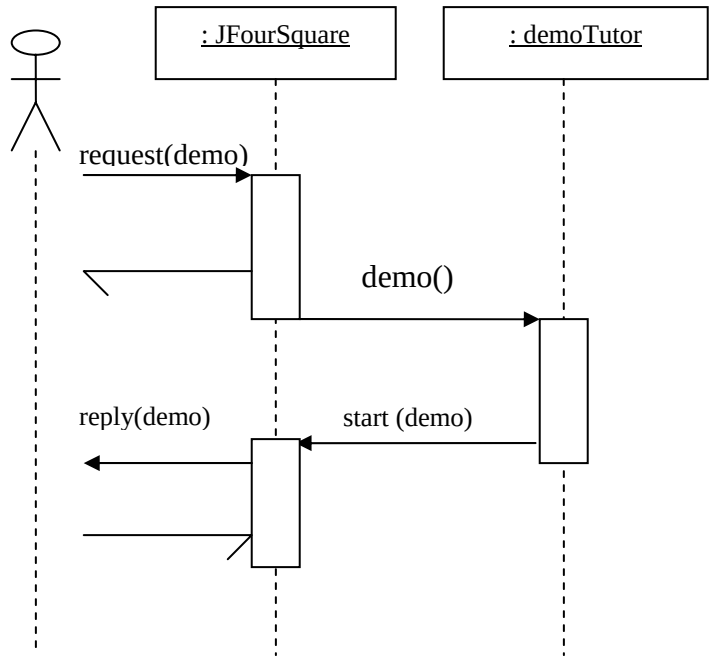

Figura 13.1 Diagrama de secuencia para solicitud de “Ejemplo de Uso”

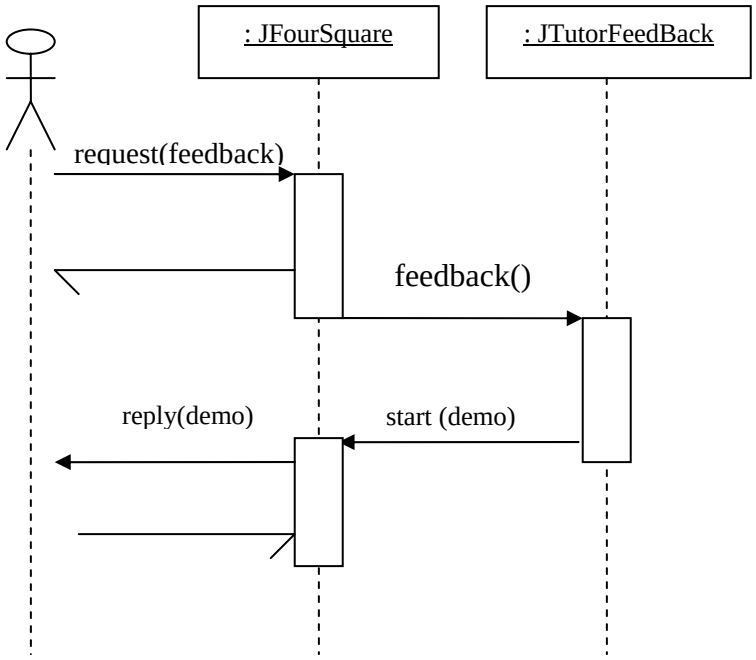

Figura 13.2 Diagrama de secuencia para solicitud de "Retroalimentación"

Los dos diagramas anteriores describen la secuencia de interacción entre los objetos JFourSquare y demoTutor, para el primer caso, en la figura 13.1 y JFourSquare y JTutorFeedBack para el segundo caso mostrado en la figura 13.2

El primer diagrama de secuencia muestra que el objeto JFourSquare recibe una petición de su entorno para mostrar el "Ejemplo de uso". Este objeto confirma que recibe la petición. La flecha con media cabeza indica que el emisor del mensaje no espera una contestación. Este objeto envía un mensaje a demoTutor para que despliegue el "Ejemplo de uso", entonces el tiempo de acción de JFourSquare termina; demoTutor toma el control, y comienza el "Ejemplo de uso", y regresa la demostración al usuario.

En el figura 13.2 se muestra un diagrama similar, solo que los objetos en este caso son JFourSquare y JTutorFeedBack, donde el primero le hace la petición al objeto JTutorFeedBack de crear una oración de retroalimentación, este objeto recibe la petición y crea dicha oración que posteriormente es enviada a JFourSquare quien lo despliega en la interfaz para que el usuario esté enterado de sus aciertos y errores.

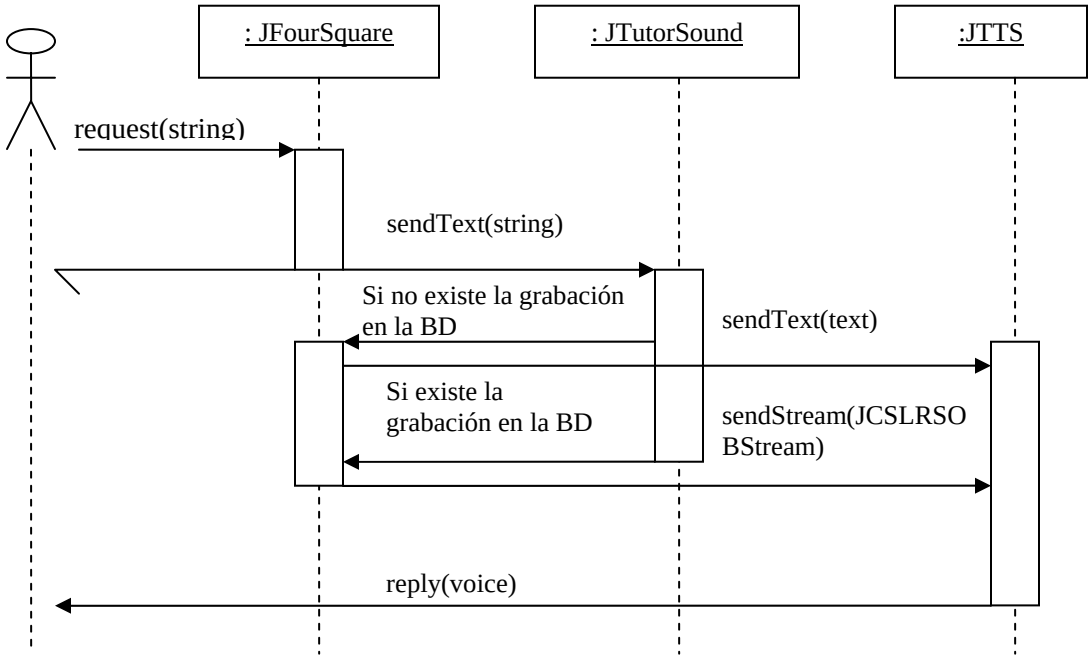

Figura 13.3 Diagrama de secuencia para solicitud del TTS

En este último diagrama se muestra la secuencia para la solicitud del sintetizador de voz. JFourSqure recibe la petición del usuario al presionar una palabra para que sea reproducida, este manda la cadena de la palabra que recibió a JTutorSound que desempeña la función de buscar la grabación de dicha palabra en la base de datos, si no existe JFourSquare manda dicha cadena al JTTS para sintetizar la palabra; si existe JFourSquare

manda la grabación que se obtuvo de la base de datos a JTT para que la reprodusca. JTT es en cual quiera de los dos casos se encarga de la reproducir el sonido de la palabra.

3.1.3 Diagramas de clase

En este punto se mostrarán los diagramas de clase correspondientes a JFourSquare, demoTutor, JTutorFeedBack, JTTS. En estos diagramas mostramos los métodos de estas clases a los que se les aplicó reingeniería para lograr el desarrollo del tutor en español.

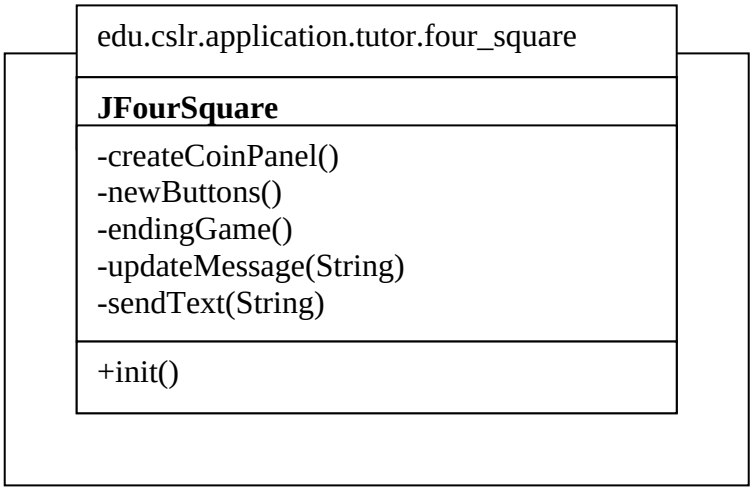

Figura 14. Diagrama de la clase JFourSquare

El diagrama de la figura catorce, muestra la clase JFourSquare que es la clase encargada de construir la interfaz del tutor y de instanciar los objetos que crean las funciones principales del tutor. El método createCoinPanel() es el encargado de construir el panel de la interfaz en el que se muestra el puntaje que obtiene el alumno, es decir, el

número de monedas de oro, plata y bronce que acumula el alumno, al responder cada uno de los ejercicios.

El método newButtons() es el encargado de crear y actualizar los cuatro botones, que corresponden a cada palabra de cada ejercicio, también es el encargado de mostrar la imagen que corresponde a la palabra que el alumno debe de encontrar entre las cuatro opciones posibles.

El método endingGame() tiene la función de actualizar la información del usuario en la base de datos cuando se termina un nivel de ejercicios, es decir cuantos aciertos obtuvo en el nivel correspondiente. Además es el encargado de mandar al objeto JTutorFeedBack el número de aciertos correctos para que este construya la oración de retroalimentación, así como también llama al método updateMessage(String), el cual usa la oración construida en JTutorFeedBack para imprimir el letrero de retroalimentación en la interfaz de FourSquare.

Uno de los métodos importantes dentro de JFourSquare es sendText(String), este método recibe una cadena que contiene el texto que pronunciará el agente animado; toma el String que recibe y lo manda al método getJCSLRSOBStream de JTutorSound, para que éste ultimo realice una consulta a la base de datos. Si éste encuentra en la base de datos la grabación del sonido correspondiente a la palabra que recibió, regresa un dato tipo JCSLRSOBStream. JFourSquare.sentText(String) recibe este último dato y reproduce la grabación a través del método sendStream(JCSLRSOBStream) de la clase JTTS; si

JTutorSound no encuentra la grabación, regresa null y el mismo String que recibió anteriormente lo envía al método sentText(String) de la misma clase JTTS para que construya la palabra a través del TTS. El diagrama de la clase JTT se muestra en la siguiente figura.

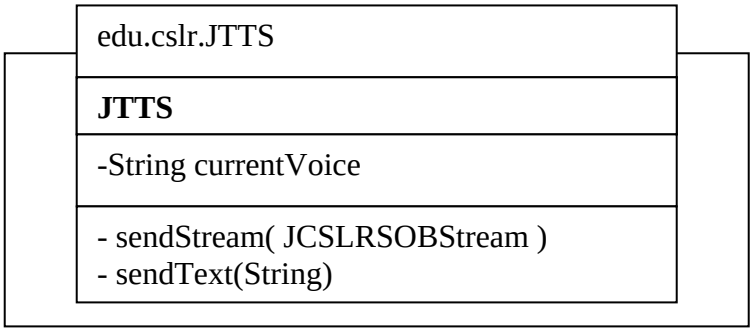

Figura 15. Diagrama de clase de JTTS

El diagrama de la figura 15 presenta los métodos que mencionamos anteriormente, sendStream(JCSLRSOBStream) y sendText(String), además de que muestra un dato privado tipo String llamado currentVoice. Este último es importante ya que a partir de éste se define el tipo de voces que el sintetizador de voz usará para construir las palabras, es decir, currentVoice puede ser igual a las cadenas “hvs”, “ttl”, “bcs”, entre otras. Estas últimas son los nombres que reciben las carpetas del sintetizador de voz, que guardan las voces en diferentes idiomas, por ejemplo “hvs” es el archivo del TTS para las voces en español femenino; “tll” es el archivo del TTS para las voces en inglés femenino y “bcs” es el archivo del TTS para las voces en alemán femenino. Es importante mencionar que para poder usar las voces del servidor Festival, en español femenino, en el tutor de español que se desarrolló, el String currentVoice de la clase JTTS se inicializó con la cadena “hvs” dentro de la misma clase de JTTS, debido a que es un dato privado y que no se puede modificar desde fuera de la clase a la que pertenece.

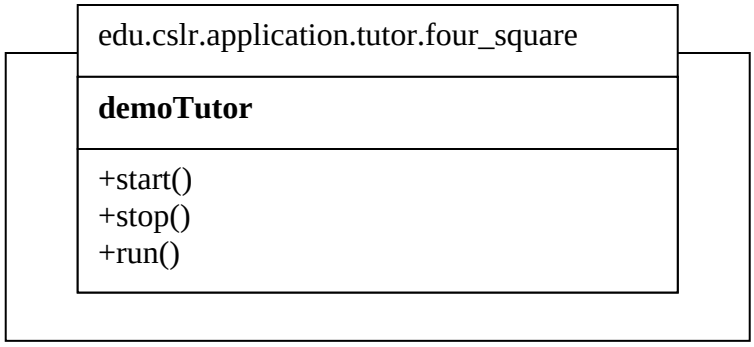

Figura 16. Diagrama de clase de demoTutor.

En la Figura 16, se muestra el diagrama de clase de demoTutor, esta clase es un thread encargado de controlar la demostración del ejemplo de uso, en su método run() se crea una lista de cuatro palabras ejemplo para mostrarlas en cada uno de los cuatro cuadros, y así el alumno pueda identificar cómo serán los ejercicios que se le presentarán. También es el encargado de mandarle la palabra ejemplo a Festival, para que este, la construya y la pronuncie en el las instrucciones del ejemplo.

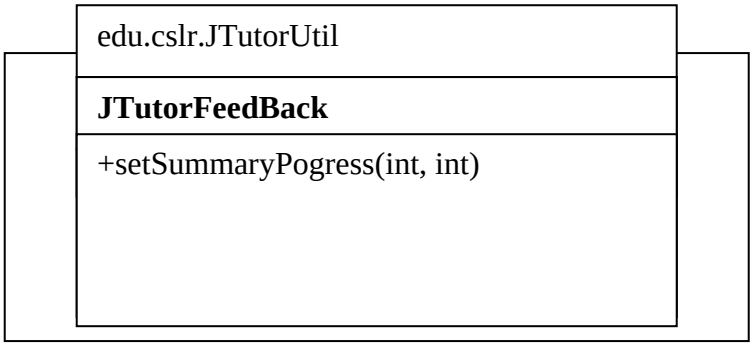

Figura 17. Diagrama de clase de JTutorFeedBack

En el diagrama de la figura 17, podemos ver la clase JTutorFeedBack y los métodos que usamos para implementar el tutor en español. Esta clase en su método setSummaryPogress() recibe dos enteros, son el numero de aciertos correctos que tuvo el alumno y el número de intentos en el que logró los aciertos.

3.1.4 Tablas de la Base de Datos del Sistema de Tutores

La base de datos del sistema de tutores contiene en total ciento un tablas, en las cuales se guarda información a cerca del usuario, del plan de estudios, de los grupos de estudiantes, de los libros interactivos, de las palabras y oraciones grabadas, de algunos fonemas en inglés, entre otros.

Como lo mencionamos anteriormente el tutor FourSquare accesa a la base de datos para obtener la información de cada uno de los ejercicios que se mostrarán en cada nivel. Esta información está guardada en la tabla llamada words_four_square, la cual se describe en la tabla diez.

words_four_square	
fq_id	integer
fq_word	character varying (30)
fq_foils	text
fq_word_phoneme	character varying(60)
fq_word_phoneme_cvt	character varying(20)

Tabla 7. Tabla de la base de datos words_four_square.

Esta tabla contiene cinco propiedades, la primera es fq_id que es un entero que describe el nivel al que pertenece cada ejercicio que se mostrará en la interfaz; fq_word es la palabra que el tutor solicita al alumno encontrar entre las cuatro opciones posibles; fq_foil es un conjunto de cuatro palabras, de entre las cuales son seleccionadas tres, y las cuales se mostrarán como posibles opciones de respuesta del ejercicio; fq_word_phoneme, contiene la descripción fonética de la palabra que solicita el tutor; fq_word_phoneme_cvt, contiene la descripción consonante-vocal de la palabra solicitada.

Un ejemplo del contenido de la tabla words_four_square se presenta en la siguiente tabla:

fq_id	fq_word	fq_foils	fq_word_phoneme	fq_word_phoneme_cvt
10	grouch	grinch grow roach couch	G R AW CH	ccvc
9	drum	dorm drim strum jump	D R AH M	ccvc
8	cow	couch coo cue com	K AW	cv

Tabla 8. Ejemplo de contenido de la tabla words_four_square

El tipo de dato character varying es un tipo de dato que permite definir una cadena de tamaño n, por ejemplo 60; pero que si la cadena almacenada en este tipo de dato no es lo

suficientemente grande para cubrir el tamaño definido, por ejemplo 40, cuando se realiza una consulta para obtener esta misma cadena, se recibirá como resultado una cadena de tamaño 40, sin el espacio extra de 20. El tipo de dato text, permite definir una cadena sin límite de tamaño, lo cual ayuda, ya que el tamaño de las cuatro palabras opcionales pude variar de longitud y puede ser muy grande.

3.1.5 SW y HW utilizado

El sistema de Tutor de Lectura del CSLR utiliza varios recursos en cuanto a software. A continuación se dará una breve explicación de cada uno de estos, para mayor información del por qué estos recursos y su instalación, se puede consultar el trabajo de investigación realizado por Baruch Roger López López, y que se encuentra descrito en su tesis "Implantación de Lecciones de Lecto_Escritura en Inglés para estudiantes de Primaria". [BARUCH, 03].

3.1.5.1 SW utilizado

El sistema usa el servidor Apache, el cual permite guardar la información del sistema completo del CSLR, de tal forma que se puede proporcionar el servicio del tutor de lectura vía web. Los archivos del sistema de tutores se encuentran dentro del las carpetas htdocs/beginweb. Dentro de esta existe un fólder llamando scripts, que a su vez contiene más archivos y una carpeta llamada Tutors, es en esta última donde se guarda las carpetas

de cada tutor creado, entre ellos FourSquare, dichas carpetas contienen las clases, páginas php, archivos xml, e imágenes de cada tutor.

Una de las tecnologías que usan los tutores es PHP, este es un lenguaje de scripts que es utilizado en los tutores ya que ayuda a las páginas html a ser más dinámicas. Los archivo php que usan los tutores se encuentran en la carpeta htdocs/beginweb en el servidor de Apache. Algo importante a considerar es que este servidor debe de estar configurado para poder usar páginas php, dicha configuración se describe el la tesis anteriormente mencionada. [BARUCH, O3]

Para la base de datos el sistema de tutores utiliza Cywing que es un software que permite la emulación de Linux dentro del sistema operativo de Windows y es necesario ya que permite el manejo de PostgreSQL que es un sistema de administración de bases de datos de objetos relacionales, y con el cual se administra la información de los tutores. Los archivos que de contienen la información de la base de datos se encuentran dentro de la carpeta de Cywin/home/user_name. Esta carpeta contiene tres archivos importantes que son: system.sql que se encarga de crear las tablas e index de de la base de datos; otro archivo es view_word_level.sql que tiene la función de crear la vistas que se usarán en los tutores y las cuales, para el caso del tutor en español, se describirán más adelante; createuser.sql es otro archivo importante ya que contiene instrucciones para crear un usuario de la base de datos y le asigna un id y password.

Como lo mencionamos anteriormente en los diagramas de uso, el sistema usa el software llamado Festival, que es el encargado de la parte de síntesis de voz y que fue desarrollado por el “Centre for Speech Technology Research”. [CSTR, O4] El proceso de síntesis de voz se lleva acabo a través de la transformación de texto a una secuencia de sonidos, este proceso es mejor conocido como TTS (Text to speech). [LUNA, 2001]

El sistema de tutores utiliza el software de CUAnimate, que es un conjunto de herramientas para generar personajes animados en tercera dimensión. Actualmente existen ocho personajes completos, cada uno tienen un esqueleto totalmente articulado, y un conjunto de visemas que permiten la articulación de los sonidos en inglés. [CSLR,04]

Finalmente mencionaremos que se requiere tener instalado el componente de Run Time Environment de Java y configurar Windows para poder ejecutar los comandos de java desde el “command prompt". Esto es necesario ya que los códigos de los tutores fueron desarrollados usando el lenguaje de Java y para los cuales se requiere de versiones de JRE a partir de 1.4.2

Para configurar JRE en Windows se deben de crear dos variables de ambiente CLASSPATH y Path, en el menú de “System Properties” en la opción “Advanced”, y que deben contener los siguientes paths:

Variable name: CLASSPATH

Variable value: C:\WINDOWS\java\clases;

Variable name: Path

Variable value: C:\j2sdk1.4.2_04\jre\bin

3.1.5.2 HW utilizado

En base a las conclusiones de la tesis citada anteriormente, se puede decir que los requerimientos mínimos para el funcionamiento de los tutores son: procesadores Pentium III a 650Mhz de velocidad con 128 MB en memoria RAM. El principal inconveniente en utilizar requerimientos mínimos es que aunque los tutores pueden funcionar correctamente en casos donde su uso es moderado, habrá casos, principalmente cuando el usuario es el alumno, donde el sistema esté sometido a diversas peticiones, por lo cual la memoria y el procesador no serán suficientes, lo que puede ocasionar fallas en el funcionamiento. Por lo que se recomienda aumentar la memoria RAM a 512MB y usar procesadores Pentium IV. [BARUCH, O3].

3.2 Desarrollo de un tutor de lecto-escritura basado en FourSquare

Una vez descritos los casos de uso, la base de datos, los diagramas de clase, y el software y hardware utilizado por el sistema de tutores, podemos comenzar con la descripción de la implementación del tutor de lecto-escritura del español. A continuación se presentan cada uno de los pasos que fueron necesarios para crear dicho tutor.

3.2.1 Insertar material didáctico a la base de datos

Una vez recolectado el material didáctico que se presentaría en el tutor de FourSquare, fue necesario conocer el contenido de la base de datos, lo que implicaba aprender las instrucciones básicas de PostgreSQL, esto con el fin de poder manejar la base de datos, insertar datos, borrar datos, crear vistas, leer datos desde un archivo, entre otras funciones que presenta dicho administrador de bases de datos. Para este fin se consultó el libro en línea "Practical PostgreSQL" del autor Worsley, John. [WORSLEY, O4].

Para conocer sobre el acceso a la base de datos y la instalación del sistema de tutores se consultó al Ing. Eduardo Clemente Fragoso y al Ing. Mariana Valdivia Zamorano. De igual forma se consultó la tesis presentada por Roger Baruch López, ya que presentaba las instrucciones del manejo de Cygwin y de la inicialización de la base de datos.

Para accesar a la base de datos se debe de abrir una terminal del emulador de Linux, Cygwin. Una vez abierta, en la línea de comando se debe de teclear lo siguiente: **./db** y presionar la tecla de "enter". Esto hará que se inicie el servicio de la base de datos, una vez que se inicia este servicio, en la terminal se podrá leer la leyenda: **LOG: database system is ready.** Para poder acceder al administrador de la base de datos, después de leer la leyenda anterior, presionar "enter", para volver a la línea de comandos, y teclear la siguiente instrucción: **psql isystemDB test,** (donde test es el nombre del usuario de la base de datos) y presionar "enter", inmediatamente el sistema hará la petición del **password** del

usuario, para este caso el password es "**test**". A continuación se muestra en la figura 18, un snapshot del procedimiento anterior, para iniciar y tener acceso a la base de datos del sistema.

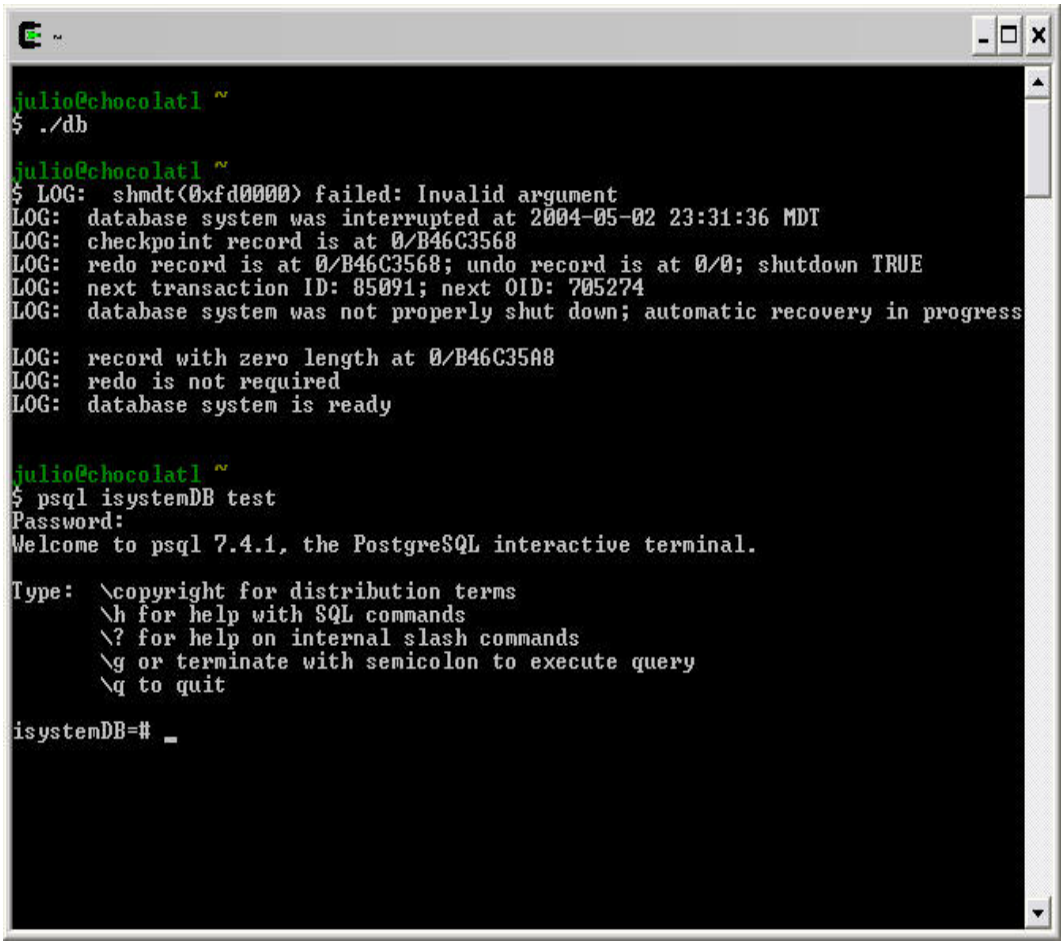

Figura 18. Procedimiento para iniciar y accesar a la BD del sistema de tutores.

Una vez dentro del ambiente de PostgreSQL se procedió a insertar los datos del material didáctico dentro de la tabla words_four_square. Como fue descrito anteriormente, esta tabla contiene los datos necesarios para cada nivel de ejercicios del tutor de FourSquare.

words_four_square	
fq_id	integer
fq_word	character varying (30)
fq_foils	text
fq_word_phoneme	character varying(60)
fq_word_phoneme_cvt	character varying(20)

Tabla 9. Tabla de la base de datos words_four_square.

Dentro de esta tabla existen datos relacionados con el tutor FourSquare en inglés, actualmente esta tabla cuenta con diez niveles de ejercicios para este idioma, a partir de este conocimiento y por acuerdo con del desarrollador del software de los tutores y libros interactivos, Nattawut Ngampatipatpong en la Universidad de Colorado, se crearon una lista de ejercicios para el español y a los cuales se les asignaron niveles a partir del 502. Cabe mencionar que no fue posible crear una tabla adicional para los datos del tutor FourSquare en español ya que el desarrollador no lo consideró conveniente.

Para insertar los datos de los ejercicios en español, se creo un archivo llamado DatosFourSquareEspanol.sql dentro de la carpeta **Cywin/home/user_name,** dicho archivo contiene líneas de instrucciones SQL y datos de los ejercicios, a continuación se presenta ejemplos del contenido del archivo sql:

INSERT INTO WORDS_FOUR_SQUARE VALUES ('502', 'sa', 'oso si so su', 's a', 'cv');
INSERT INTO WORDS_FOUR_SQUARE VALUES ('502', 'se', 'sea su si so', 's e', 'cv');
INSERT INTO WORDS_FOUR_SQUARE VALUES ('503', 'cono', 'cana cuna coco codo', 'kc k o n o', 'cvcv');
INSERT INTO WORDS_FOUR_SQUARE VALUES ('503', 'coco', 'cueva cono cuna codo', 'kc k o kc k o', 'cvcv');

Tabla 10. Ejemplo de contenido del archivo DatosFourSquareEspanol.sql

Debido a que el idioma español maneja caracteres especiales como lo son: diéresis (¨), eñe (ñ) y acentos (´), no fue posible insertar los datos leyendo el archivo sql creado, desde la línea de comando de PostgreSQL, por lo que fue necesario crear un programa en java llamado, CaracteresEspeciales.java para insertar los datos. A continuación se presenta en la figura diecinueve el diagrama de clase de CaracteresEspeciales.

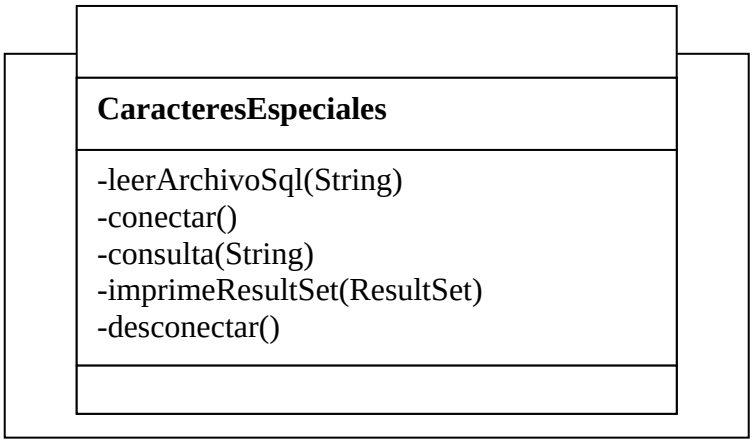

Figura 19. Diagrama de clase de CaracteresEspeciales.

El método leerArchivoSql(String) realiza la función de leer el archivo que contiene los datos de los ejercicios, mientras lee una por una las líneas del archivo, intercambia los caracteres: diéresis (¨), eñe (ñ) y acentos (´) por su equivalente en código ASCII para html, para que puedan ser desplegados correctamente desde un navegador.

Los métodos conectar() y desconectar(), como su nombre lo describe son los encargados de hacer la conexión a la base de datos y cerrarla. El método consulta(String) recibe un string, que es el query que se va a ejecutar en la base de datos. El método imprimeResultSet(ResultSet) recibe un conjunto de resultados del query que se ejecutó, y los imprime.

Así al ejecutar esta clase, se abre una conexión a la base de datos y se ejecutan las instrucciones que contiene el archivo DatosFourSquareEspanol.sql, de tal forma que los caracteres especiales son insertados en la base de datos sin problemas.

Una vez insertado el material en la tabla words_four_square, se construyeron un conjunto de vistas en la base de datos para facilitar el acceso, manejo y organización de los datos almacenados en dicha tabla.

Como se mencionó anteriormente existe en la carpeta de Cygwin/user/home un archivo llamado view_word_level.sql, que es el encargado de crear las vistas necesarias para dividir los datos de la tabla words_four_square (entre otras que contienen los datos del resto de los tutores) por niveles y simplificar las consultas a la base de datos.

Para el nuevo tutor se creo un archivo llamado vistas_datos_español.sql, que contiene la definición de las vistas que se usarán para dividir por niveles de dificultad, el contenido del material en español. En dicho archivo se crean 18 vistas una por cada nivel, a continuación se muestra un ejemplo de la primera vista.

```
=====================================================
-- Vista 502. Four Square : consonantes M,m, S,s, T,t, L,l, R,r, P,p
-- Sonidos cv - Nivel 502
=====================================================

CREATE VIEW WORDS_LEVEL_502 AS SELECT '[' || FQ_WORD || ' ' || FQ_FOILS ||']'
AS CMU_WORD, FQ_ID , FQ_WORD , FQ_FOILS , FQ_WORD_PHONEME , '0' AS
QUERY_TYPE FROM WORDS_FOUR_SQUARE
WHERE FQ_ID = '502' AND (FQ_WORD like 'm%' OR FQ_WORD like 's%' OR
FQ_WORD like 't%' OR FQ_WORD like 'l%' OR FQ_WORD like 'r%' OR FQ_WORD like
'p%' ) AND FQ_WORD_PHONEME_CVT = 'cv';

=====================================================
```

La vista anterior define el contenido del nivel 502, el cual muestra material sobre las consonantes m, s t, l, r y p. Se crea una vista llamada WORDS_LEVEL_502 que concatena en un arreglo llamado CMU_WORD el contenido de las columnas FQ_WORD y FQ_FOILS de la tabla WORDS_FOUR_SQUARE; selecciona los demás campos FQ_ID, FQ_WORD, FQ_FOILS_ FQ_WORD_PHONEME; asigna a un campo llamado QUERY_TYPE que es propio de la vista, el valor de 0 (cero); y seleccionará estos datos donde FQ_ID (el nivel) sea igual a 502 y FQ_WORD (la palabra que se escogerá en el ejercicio) empiece con n, s, t, l, r, p y además este construida por FQ_WORD_PHONEME_CVT igual a cv (consonante vocal).

La vista anterior regresará un conjunto de resultados de la siguiente forma:

cmu_word	fq_id	fq_word	fq_foils	fq_word_phoneme	query_type
[ma amo ama mu amo]	502	ma	mo ama mu amo	ma	0
[me mu mo mi ama]	502	me	mu mo mi ama	me	0

Tabla 11. Resultado de la vista WORDS_LEVEL_502

El resto de las vistas que se construyen en este archivo, siguen la misma estructura que la vista anterior, solo varían los niveles del material seleccionado y la forma de seleccionarlo. La siguiente vista define el material para el último nivel de ejercicios.

```
==========================================================
-- Vista 519. Four Square : Pl, Bl, Cl, Fl, Gl y  anteriores
==========================================================

CREATE VIEW WORDS_LEVEL_519 AS SELECT '[' || FQ_WORD || ' ' || FQ_FOILS ||']'
AS CMU_WORD , FQ_ID , FQ_WORD , FQ_FOILS , FQ_WORD_PHONEME , '0' AS
QUERY_TYPE FROM WORDS_FOUR_SQUARE WHERE FQ_ID = '519' AND
FQ_WORD_PHONEME_CVT not like 'ccv';
==========================================================
```

De la misma forma que la vista 502, la anterior define el contenido pero para el nivel 519, el cual muestra material sobre palabras que contienen la combinación de sonidos pl, bl, cl, fl, y gl. Así mismo, se crea una vista llamada WORDS_LEVEL_519 que

concatena en un arreglo llamado CMU_WORD el contenido de las columnas FQ_WORD y FQ_FOILS de la tabla WORDS_FOUR_SQUARE; selecciona los demás campos FQ_ID, FQ_WORD, FQ_FOILS_ FQ_WORD_PHONEME; asigna a un campo llamado QUERY_TYPE que es propio de la vista, el valor de 0 (cero); y seleccionará estos datos donde FQ_ID (el nivel) sea igual a 519 y además este construida por FQ_WORD_PHONEME_CVT igual a ccv (consonante consonate vocal).

La vista WORDS_LEVEL_519 regresa un resultado de la siguiente forma:

cmu_word	fq_id	fq_word	fq_foils	fq_word_phoneme	query_type
[ma amo ama mu amo]	502	ma	mo ama mu amo	ma	0
[me mu mo mi ama]	502	me	mu mo mi ama	me	0

Tabla 12 . Resultado de la vista WORDS_LEVEL_519

3.2.2 Imágenes para palabras en español

Las imágenes son un punto importante en el desarrollo del tutor en español, ya que el estudiante puede relacionar el significado de la palabra, con la acción, objeto, persona, animal o cosa que se muestra en cada imagen.

Para el tutor de FourSquare se crearon y editaron en total 314 imágenes que fueron almacenadas en el fólder llamado tutor_imagenes_espanol. Algunas de las imágenes con las

que contaba el tutor de FourSquare en inglés fueron reutilizadas, otras fueron editadas de tal manera que representaran el significado de las palabras en español; otras imágenes fueron creadas, ya que no existían dentro de las imágenes del tutor en inglés.

Figura 19. Imagen creada que representa la palabra familia

Figura 20. Imagen reutilizada que representa la palabra “coco”

Figura 21. Imagen editada para representar la palabra examen

Figura 22. Imagen original del tutor FourSquare que representa la palabra “cheat” (hacer trampa)

3.2.3 Imágenes para interfaz en español

Como se puede notar en las figuras anteriores que describen la interfaz de FourSquare, las imágenes y etiquetas de los botones que se presentan están en inglés, ya que inicialmente los tutores fueron planeados para usuarios que hablan este idioma. Por lo que un punto importante a considerar en el desarrollo del tutor en español es proporcionar una interfaz entendible para el estudiante que habla español.

En base a las reglas de diseño de interfaces, más específicamente la regla cinco que habla sobre la claridad del lenguaje y la regla catorce que habla sobre las propiedades culturales [KIRSCHNING, O4], se realizaron cambios a las etiquetas e imágenes de la interfaz, con el objetivo de hacer que los mensajes visuales de la interfaz fueran claros y

entendibles para el nuevo grupo de usuarios cuya lengua es el español. Mas adelante en el se explicará qué parte del código de JFourSquare crea los botones de la interfaz y define el conjunto de imágenes que usa.

3.2.4 Análisis y modificaciones del código JFourSquare.java

Para llevar a cabo las modificaciones necesarias a la interfaz, se tuvieron que analizar y realizar cambios al código de JFourSquare, puesto que entre otras funciones, crea los componentes de la interfaz de dicho tutor. A continuación se describen dicho análisis y modificaciones:

a) Las siguientes líneas de código se encuentran en el método init() de la clase JFourSquare, en estas se muestra el nombre de las imágenes que se usan en la interfaz para los botones, lo que facilitó hacer una búsqueda de dichas imágenes, para posteriormente editarlas, de tal forma que las etiquetas se mostrarán en español. Este conjunto de imágenes editadas fue guardado en fólder llamado imágenes_español_interfaz.

```
/************************************************************************** */
AQUI PODEMOS SABER QUE IMAGENES USA LA INTERFAZ DE FOURSQUARE PARA LOS BOTONES.
**************************************************************************
167     try
168     {
169             exitIcon = new ImageIcon(new URL(imagePath + "exit_button.gif"));
170             demoIcon = new ImageIcon(new URL(imagePath + "demo_button.gif"));
171             startIcon = new ImageIcon(new URL(imagePath + "start_button.gif"));
172             nextIcon = new ImageIcon(new URL(imagePath + "next_button.gif"));
173             againIcon = new ImageIcon(new URL(imagePath + "listen_icon2.gif"));
174     }
175     catch (Exception e) {}
/* ************************************************************************** */
```

b) En el mismo método init() se encuentra la línea de código que crea el botón para la acción “Repetir pronunciación”. Este botón originalmente mostraba la etiqueta “Again” , por lo que se cambió por la etiqueta “Repetir pronunciación”.

```
/* ********************************************************************** */
EN ESTA LINEA DE CÓDIGO SE CREA EL BOTÓN PARA LA ACCIÓN “REPETIR PRONUCIACIÓN”
/* ********************************************************************** */

375     speakBut = new JButton("Repetir Pronunciación",againIcon);

377     speakBut.setFont(fontHelp);
378     speakBut.setBackground(Color.yellow);
379     speakBut.setActionCommand("SPEAK_AGAIN");
380     speakBut.addActionListener(this);

/********************************************************************** */
```

c) En el método createCoinPanel() se crea un panel para mostrar las monedas acumuladas por el alumno. Este panel presentaba las leyendas, “gold”, “silver” y “bronze”, y se realizaron los cambios necesarios para mostrarlas en español.

```
/* ********************************************************************** */
819     moneyScore[0] = new JCoinObj("oro",goldInt,imagePath);
820     moneyScore[1] = new JCoinObj("plata",silverInt,imagePath);
821     moneyScore[2] = new JCoinObj("bronce",bronzeInt,imagePath);

/* ********************************************************************** */
```

Es importante mencionar que el objeto JCoinObj() recibe el tipo de moneda , el número de monedas acumuladas por el alumno y el path dónde se localizan las imágenes de las monedas, ya que este objeto es el encargado de actualizar el puntaje del alumno, en el

panel de monedas. Es decir agrega al panel de monedas la imagen que describe el número de monedas de oro, plata y bronce acumuladas por el estudiante.

d) En la siguiente línea de código del método newButtons(), podemos observar que la imagen relacionada con la palabra, que se muestra por cada ejercicio, es construida de la siguiente manera.

```
/* ********************************************************************** */
1430  ImageIcon imgIcon = new ImageIcon(new URL(tutorImages + currentAnswer +
      ".jpg"));
1431  System.out.println(tutorImages+ currentAnswer + ".jpg");
/* ********************************************************************** */
```

Se crea un icono de imagen a partir del path del fólder que contiene las imágenes de cada palabra, la palabra que se está solicitando encontrar y la terminación “.jpg” que es el formato de la imagen.

e) Las siguientes líneas de código pertenecen al método sendText(String), que como se mencionó anteriormente se encarga de mandar el texto a la clase JTT, ya sea para que busque la grabación del texto en la base de datos o para que construya el sonido de la palabra.

```
/* ********************************************************************** */
/*ESTE ES EL METODO QUE LE MANDA EL TEXTO (PLABRAS Y FRASES ) AL        */
/* TTS                                                                    */
/* ********************************************************************** */
2250 private void sendText(String text)
2251{
2252   try
2253   {
2254            JCSLRSOBStream sob
2255            =JTutorSound.getJCSLRSOBStream(db,text,false,language);
2256
2257            if(sob != null)
2258            {
2259                     jtts.sendStream(sob);
2260            }
2261            else
2262            {
2263                     jtts.sendText(text);
2264            }
2265
/* ********************************************************************** */
```

f) En el método endingGame() se encuentran las siguientes líneas de código. La primera se encarga de mandar el número de aciertos y el número de intentos que hizo el alumno al contestar una pregunta, al método setSummaryProgressEspañol, que construye la oración de retroalimentación, al final de cada nivel de ejercicios. Como se puede notar este último método fue agregado al código JTutorFeedBack, ya que se necesitaba de un método que construyera únicamente las oraciones de retro alimentación en español.

En la segunda línea de código se manda la oración de retroalimentación construida al método updateMessage() que es el encargado de imprimir dicha oración en la interfaz, al terminar cada nivel de ejercicios.

```
/* ********************************************************************** */

        //metodo que crea la oración de retroalimentación en español
2594    feedback.setSummaryProgressEspanol(correctCount,maxTries);

        //metodo que se encarga de imprimir la oracion de retroalimentación en la interfaz
2595    updateMessage(feedback.getSummaryText());

/* ********************************************************************** */
```

El nuevo diagrama de clase de JTutorFeedBack se muestra en la siguiente figura 23 en el cual se puede notar, el método setSummaryProgressEspañol que se agregó a esta clase.

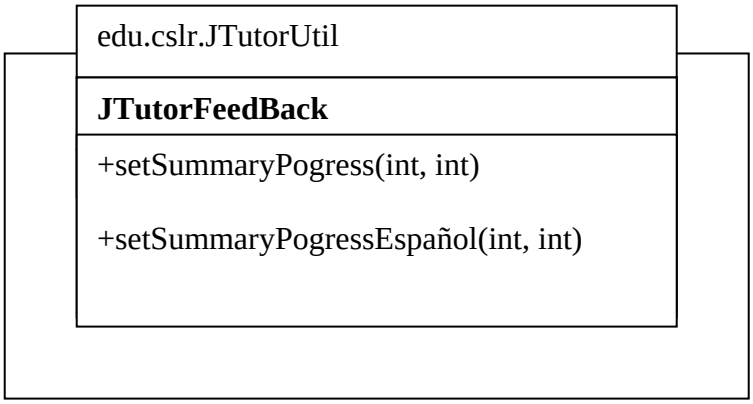

Figura 23. Diagrama de clase de JTutorFeedBack con nuevo método para retroalimentación en español.

g) Finalmente la clase demoTutor dentro de JFourSquare tiene la función de construir el ejemplo de manejo del tutor de FourSquare. Este ejemplo de uso, originalmente mostraba un conjunto de cuatro palabras en inglés, las cuales fueron modificadas para mostrar un ejemplo donde se usan palabras en español.

```
/* ********************************************************************** */
1068   if(isRun)
1069   {
1070            wordList[0].setText("papá");
1071            wordList[1].setText("pi");
1072            wordList[2].setText("pe");
1073            wordList[3].setText("pipa");
1074   }
/* ********************************************************************** */
```

3.2.5 Implementación del código en Apache

Como lo mencionamos anteriormente el servidor de Apache, contiene los archivos del sistema de tutores dentro de las carpetas htdocs\beginweb, cada tutor se encuentra en una carpeta almacenada en htdocs\beginweb\scripts\Tutors. Estas carpetas contienen los siguientes archivos:

a) Un archivo xml que tiene el mismo nombre de la carpeta que lo contiene, por ejemplo, para el tutor JFourSquare que está dentro de la carpeta JFourSquare, el archivo xml llevará el nombre JFourSquare.xml, o si la carpeta fuera JSightWords, el nombre del archivo sería JSightWords.xml. Este archivo contiene etiquetas xml, que describen frases grabadas en inglés y español, que se encuentran almacenadas en la base de datos.

Originalmente el sistema fue desarrollado pensando en usuarios cuya lengua fuera el inglés, por lo que el sistema utilizaba frases como: Good Checking!, Good Work!, Nice!, Oh! Yes!, Very Nice!, entre otras. Al comenzar a implementar los tutores en otros lenguajes, se recurrió a la siguiente solución: si el sistema se está usando en el idioma español y requiere utilizar una cierta frase; el sistema accesa al archivo xml, encuentra la

frase en inglés y busca su equivalente en español, una vez que tiene su equivalente en español, accesa a la base de datos y obtiene la frase grabada en español.

Por ejemplo, si el sistema usa la frase Nice job!, este accesa al archivo xml, y encuentra su equivalente en español:

```
<command>
        <target>        Nice job                </target>
        <spanish_mx>        Buen trabajo    </spanish_mx>
</command>
```

El equivalente en este caso sería la frase “Buen trabajo”, con este dato el sistema accesa a la base de datos y busca esta frase en la tabla tutor_sound_mx que contiene las frases grabadas en español.

b) Una imagen llamada index.gif. Esta imagen es representativa de cada tutor, y es la que se muestra en el menú inicial del sistema.

c) Un archivo php que tiene el mismo nombre de la carpeta que lo contiene, por ejemplo si se encuentra en la carpeta JFourSquare, el archivo se llamará JFourSquare.php. Este archivo tiene la función de crear la página que desplegará el applet del tutor y usa las funciones de los archivos .inc, localizados en htdocs\beginweb\lib\inc , para especificar el servidor de la base de datos, el path de las imágenes que se usarán en la interfaz, el servidor de TTS, el id del usuario, entre otras funciones.

d) Un jar en el que se encuentran almacenadas las clases del tutor y que lleva al igual que los casos anteriores, el mismo nombre de la carpeta que lo contiene, por ejemplo: JFourSquare.jar

3.2.5.1 Proceso de compilación de JFourSquare.java

Hasta ahora se ha descrito el contenido de las carpetas de cada tutor; los cambios que se realizaron al código de JFourSquare y los cambios a las imágenes que usa la interfaz de JFourSquare. Los siguientes pasos consisten en compilar el código, almacenar las clases en un jar llamado JFourSquareEspanol.jar y firmarlo, para posteriormente, poder crear una carpeta del tutor en español dentro de Apache.

1) Para poder compilar el código de JFourSquare.java se deben de tener los siguientes archivos y carpetas instalados en el directorio de JRE de java:

C:\j2sdk1.4.2_04\jre\lib\ext\ postgres.jar

C:\j2sdk1.4.2_04\jre\lib\ext\ CSLR.jar

C:\j2sdk1.4.2_04\jre\lib\ext\ JCUAnimate_1.2

C:\j2sdk1.4.2_04\jre\lib\ext\ Sonic

C:\j2sdk1.4.2_04\jre\bin\JCUAnimate_1.2.dll

C:\j2sdk1.4.2_04\jre\bin\ JScreenSetting.dll

C:\j2sdk1.4.2_04\jre\bin\ SonicAPI.dll

C:\j2sdk1.4.2_04\jre\bin\ JFestival.dll

C:\j2sdk1.4.2_04\jre\bin\ JSpeex.dll

C:\j2sdk1.4.2_04\jre\bin\ SonicApp.dll

Estos archivos son necesarios, ya que proveen de las herramientas necesarias para poder compilar los códigos del CSLR, como los son: Festival, JCUAnimate, Sonic y Postgres.

2) Una vez compilado el código de JFourSquare.java fue necesario crear un archivo JFourSquareEspanol.jar y firmarlo. El proceso de firmar un jar se lleva a cabo con el objetivo de garantizar la seguridad de los archivos almacenados en este. Es decir, si se envían electrónicamente archivos como un jar, un applet o una aplicación; la persona que reciba estos archivos necesita estar segura que la información que recibe es la original y que no fue modificada en el proceso de envío. Las firmas digitales, los certificados y los archivos de almacenamiento de llaves, ayudan a garantizar la seguridad de los archivos enviados. [TUTORIALJAVA, O4]

El proceso de firmar un jar es el siguiente:

- Desde el Command Prompt, agregar los .class a un jar con la siguiente instrucción:

```
jar cvf JFourSquareEspanol.jar edu\cslr\application\tutor\four_square\*.class
```

- Generar un par de llaves necesarias para firmar el jar, con la siguiente instrucción:

keytool -genkey -alias foursquareFiles -keystore fsquarestore -keypass aenc136 -dname "cn=UDLA" -storepass savv48

Donde keytool –genkey es la instrucción de java para generar las llaves electrónicas del archivo, a dichas llaves se les asigna un alias; en este caso por ejemplo es foursquareFiles; -keystore fsquarestore es el archivo donde se van a guardar las llaves pública y privada del jar; el password para accesar a la llave privada es por ejemplo: aenc136 y el password para accesar a al archivo fsquarestore es por ejemplo: savv48; por último –dname es la instrucción para asignar el nombre de la empresa o dueño de los documentos que se envían.

- El siguiente paso es firma el jar con la siguiente instrucción:

jarsigner -verbose -keystore fsquarestore JFourSquareEspanol.jar foursquareFiles

Donde jarsigner –verbose es la instrucción para firma el jar llamando JFourSquareEspanol.jar; -keystore indica que las llaves se encuentran dentro del archivo de almacenamiento fsquarestore

Al ejecutar esta instrucción, se hará la petición tanto del password para accesar a la llave privada, como del password para accesar a archivo de almacenamiento fsquarestore, los cuales son para este ejemplo: aenc136 y savv48 respectivamente.

- El siguiente paso consiste en exportar un certificado de seguridad, ejecutando la siguiente instrucción:

  keytool -export -keystore fsquarestore -storepass savv48 -alias foursquareFiles -file FourSquareExport

- El último paso consiste en ejecutar la siguiente instrucción, la cual permite importar el certificado de seguridad, para que se pueda ejecutar el código que se encuentra dentro del jar que se firmó anteriormente.

  keytool -import -alias foursquareFiles -file FourSquareExport.cer

  Al ejecutar esta instrucción nuevamente será solicitado el password del archivo de almacenamiento y el sistema preguntará si la información contenida en el archivo de almacenamiento es segura.

3) Una vez firmado el jar, en la carpeta Tutors dentro de beginweb/scripts en el directorio de Apache, se crea la carpeta especial para los archivos del tutor en español. En este caso la carpeta que fue creada para guardar los archivos relacionados al tutor en español se llamó JFourSquareEspanol, en este fólder se colocó el jar JFourSquareEspañol.jar; la imagen “index.gif” que identifica el tutor en español; el archivo JFourSquareEspañol.php; y finalmente el archivo JFourSquareEspañol.xml.

3.3 Liga de acceso para el nuevo tutor en el menú de tutores disponibles.

Como se puede observar en la figura 2, en el capítulo uno, el menú de tutores disponibles, muestra el icono que representa al tutor y dos ligas. La primera que es el nombre del tutor y la segunda las letras [MX], esto es debido a que los tutores pueden mostrar el material en inglés, pero usando instrucciones en inglés o español, así cuando se presionaba la liga que hace referencia al nombre del tutor, el usuario accesa al tutor completamente en inglés; de lo contrario si presionaba la liga que hace referencia a las letras [MX], entra al tutor que muestra material en inglés pero con instrucciones en español; éste ultimo con el objetivo de proporcionar un tutor de inglés a los alumnos que hablan español.

Debido a que nuestro tutor sólo es para usuarios que hablan español y que requieren aprender a escribir en español, las opciones mencionadas anteriormente no son necesarias. Por lo que fue necesario modificar este tipo de presentación para el nuevo tutor.

Para modificar la presentación del tutor en español, se realizaron los siguientes cambios a la página create_page.php, que se encuentra dentro de fólder C:/Program Files/Apache Group/Apache\htdocs/beginweb/scripts. Este archivo se encarga de crear la presentación del material disponible en la página inicial del usuario por lo que se le agregaron las siguientes líneas de código para poder mostrar sólo una liga que accesara al tutor en español

```
/************************************************************************** /
else if( $row['tutor_abbr'] == "JFourSquareEspanol"){ $arrayName[$currentIndex] =
"</A> <A href=\"javascript:void(0);\" class='NormalLink' onClick=\"openTutorIII('"
. $user_id . "','". $row['tutor_abbr'] . "');\">FourSquareEspanol</A>";
                        }
/************************************************************************** /
```

Estas líneas de código le indican que sólo se creará una liga con el texto “FourSquareEspanol”. Así la nueva presentación del icono para nuestro tutor es la siguiente, que se muestra en la figura 24.

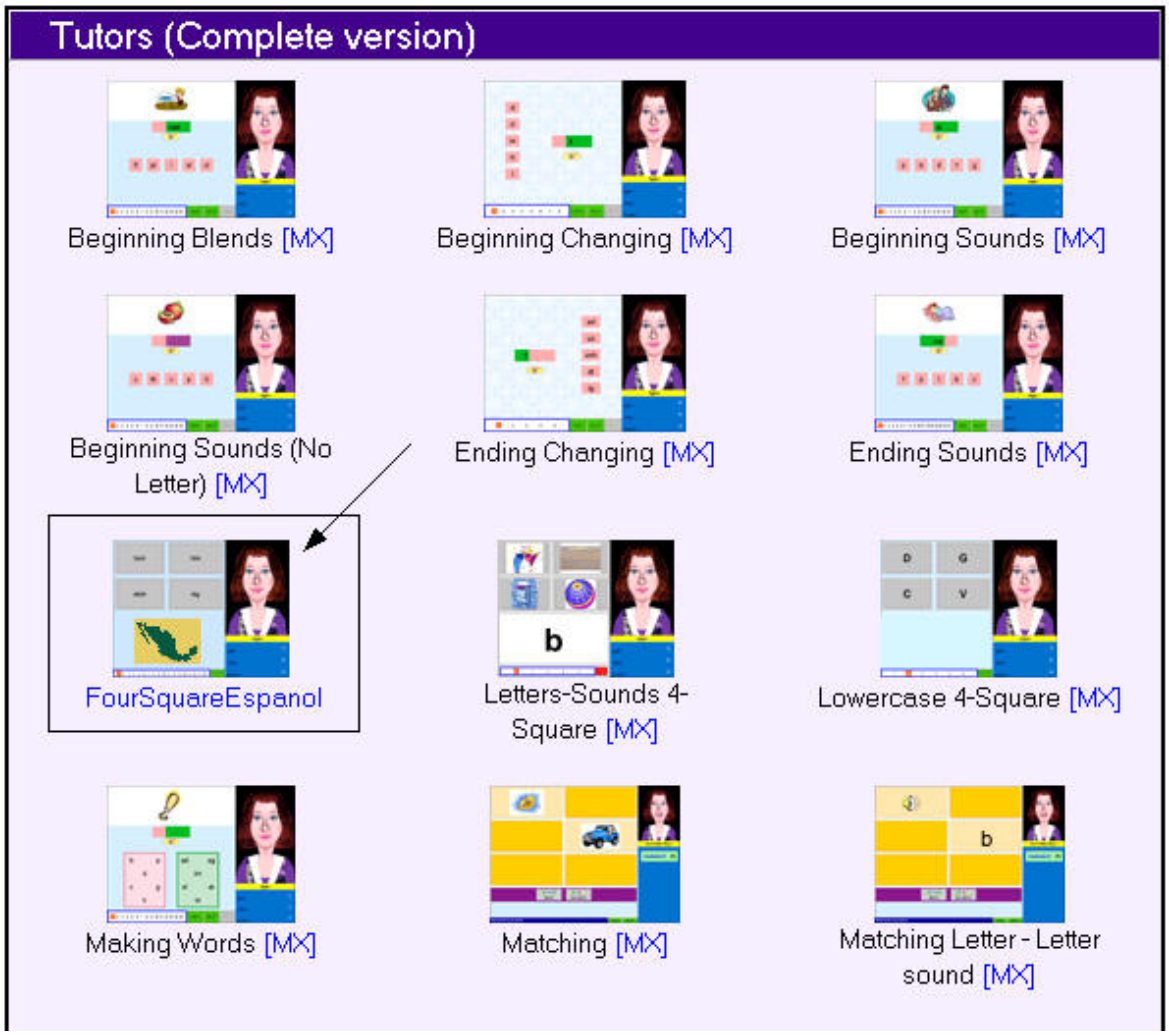

Figura 24. Nueva liga para acceder únicamente al tutor en español.

3.4 Crear tutor en español en el ambiente del sistema de Tutores

Una vez que se creo la carpeta del tutor en español, el siguiente paso es entrar al ambiente del tutor de lectura para dar de alta en el menú de tutores, el tutor en español.

La figura 25 muestra la página inicial de los tutores del CSLR. El login que se uso para entrar al ambiente es: student1 y el password: 1student1

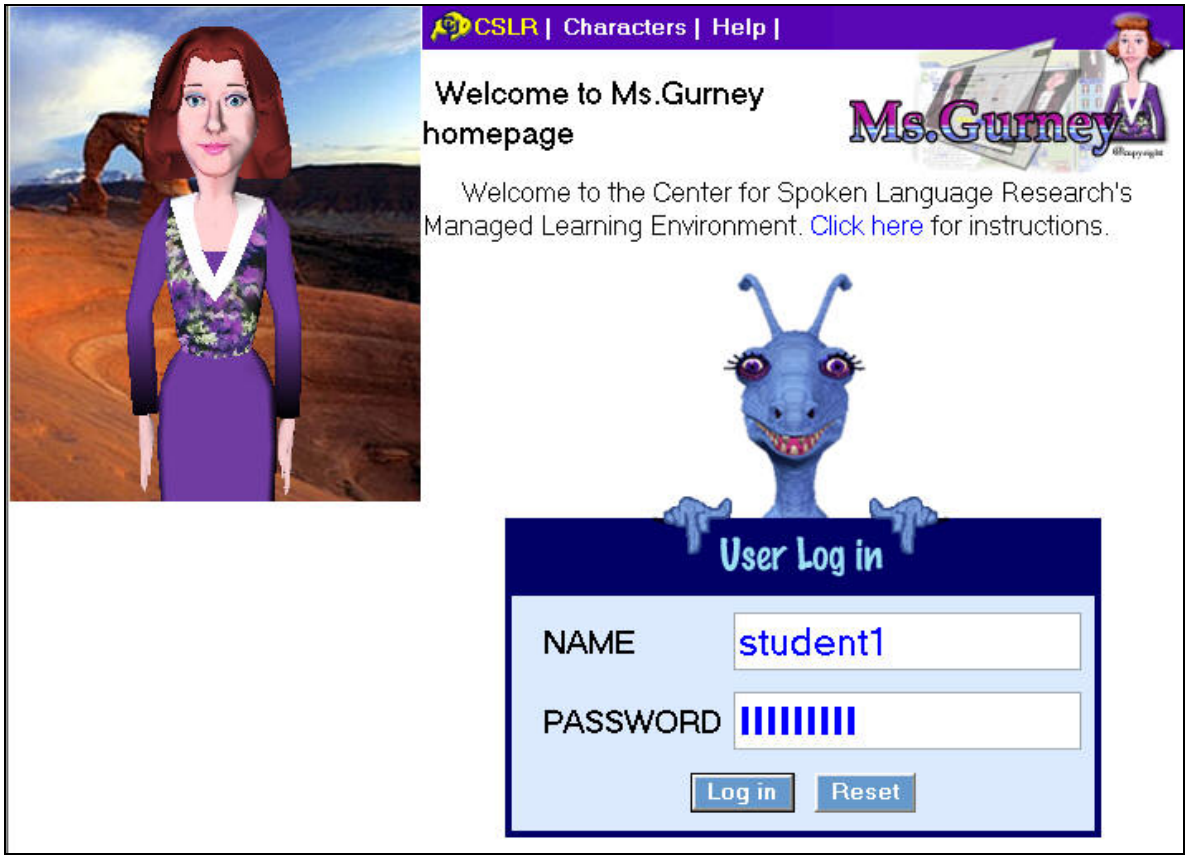

Figura 25. Pagina inicial de los tutores del CSLR.

Una vez dentro del ambiente, se accesa al menú de herramientas de configuración, “Tutors Settings” y en la parte inferior de la página existen dos botones “New” y “Delete”. Para dar de alta un nuevo tutor, se presiona el botón “New”, esta acción despliega una forma, cuyos campos se presentan en la figura siguiente:

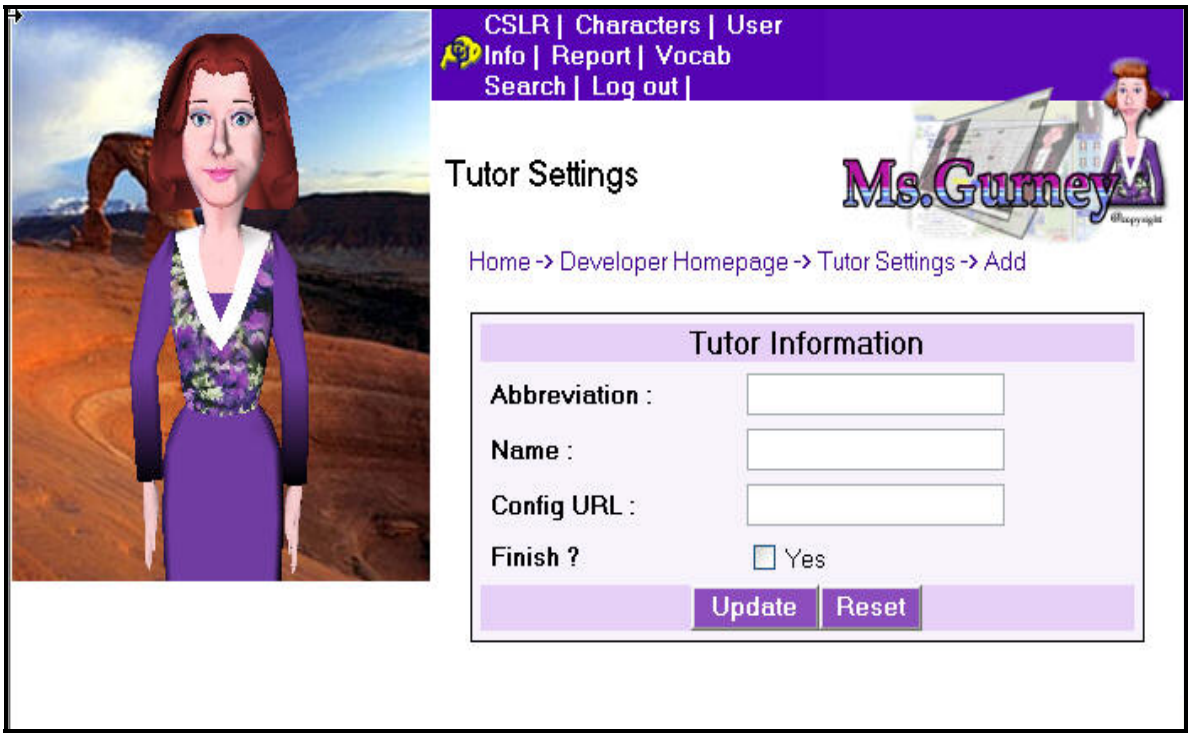

Figura 26. Forma para dar crear un tutor en el menú de tutores disponibles

En el campo “Abbreviation” se escribe el nombre de la carpeta del tutor que se da de alta, en este caso el nombre es: JFourSquareEspanol. En el campo “Name” se escribe el nombre con el cual aparecerá en el menú de tutores, por ejemplo: FourSquareEspanol. El campo “Config URL”, es opcional, en este se escribe alguna dirección URL, que contenga información adicional, como documentos o especificaciones del tutor. El campo “Finish?” se habilita para terminar la configuración y posteriormente se presiona el boton “update”

para terminar el proceso. Al momento de presionar este último botón la base de datos es actualizada y se insertan los datos del nuevo tutor en la tabla llamada tutor_program.

Una vez que se da alta el nuevo tutor, en la página de principal del usuario, aparecerá la opción de accesar al nuevo tutor FourSquareEspanol, como se muestra en la figura 27:

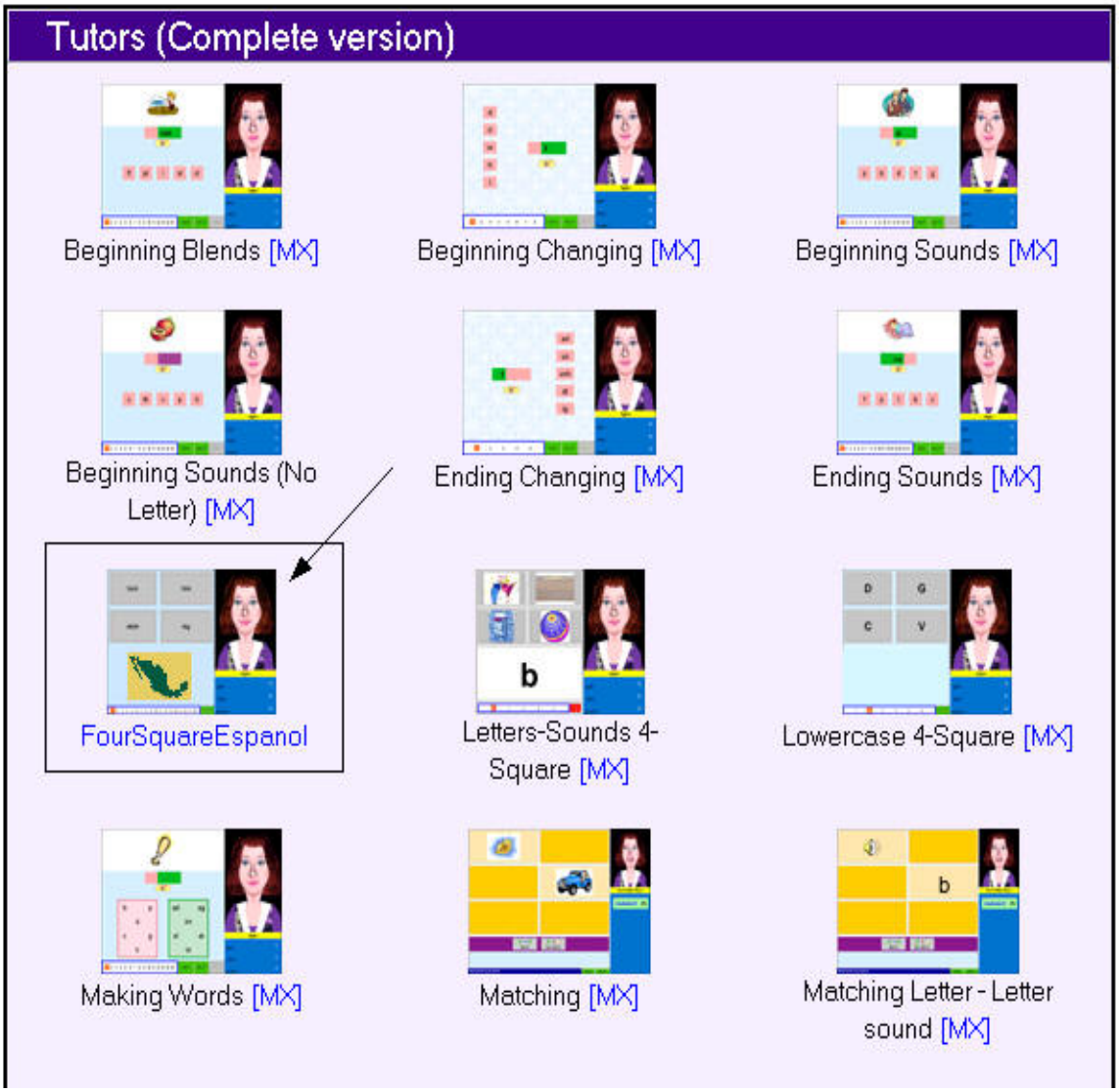

Figura 27. Tutor en Español dentro de los tutores disponibles

A partir de este paso se puede tener acceso al nuevo tutor, presionando la liga del icono del tutor FourSquareEspanol. Esta liga da completo acceso al tutor en español.

Los pasos descritos anteriormente son los que se llevaron a cabo para crear el tutor en español. En la siguiente figura se muestra la interfaz del tutor en español, en la que se pueden notar los cambios, tanto en contenido didáctico como en imágenes.

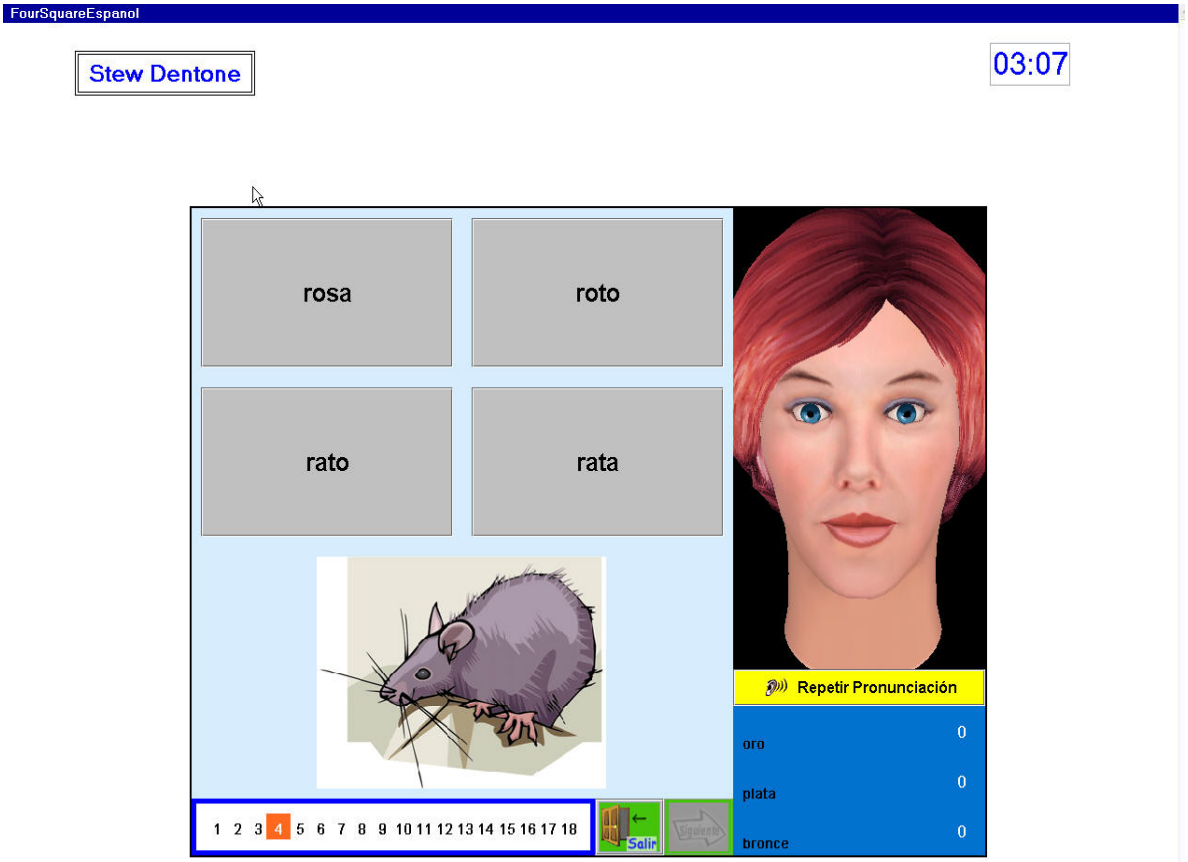

Figura 28. Interfaz del tutor de lecto-escritura del español

Es necesario mencionar que durante el desarrollo del tutor de lecto-escritura en español, la Universidad de Colorado proporcionó las nuevas versiones del agente animado, por lo que se actualizó el contenido en el ambiente de JRE, instalando las últimas versiones de los siguientes archivos y carpetas:

C:\j2sdk1.4.2_04\jre\lib\ext\ JCUAnimate_1.2

C:\j2sdk1.4.2_04\jre\bin\JCUAnimate_1.2.dll

C:\j2sdk1.4.2_04\jre\lib\ext\ CSLR.jar

Los botones, las modificaciones al código, el panel de monedas y el contenido didáctico de la interfaz en español se muestran el las figuras siguientes:

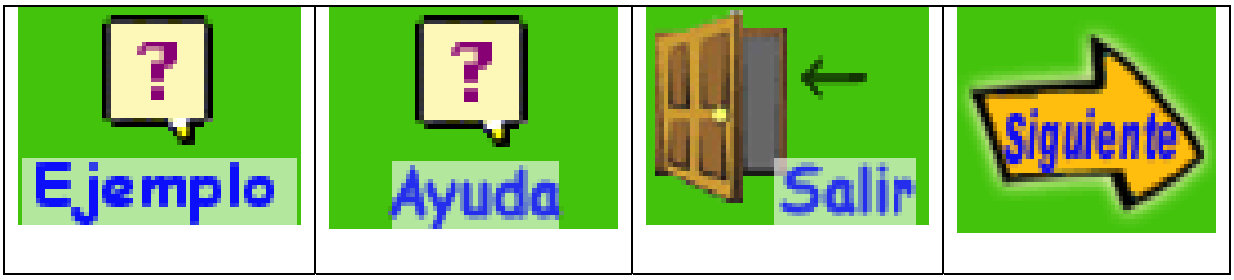

Figura 29. Imágenes que presentan los botones del tutor de lecto-escritura del español

En la figura 31, se puede observar la oración de retroalimentación y el letrero de la imagen de regalo, en la parte superior. Ambas enunciados están en español para facilitar al usuario el uso del tutor

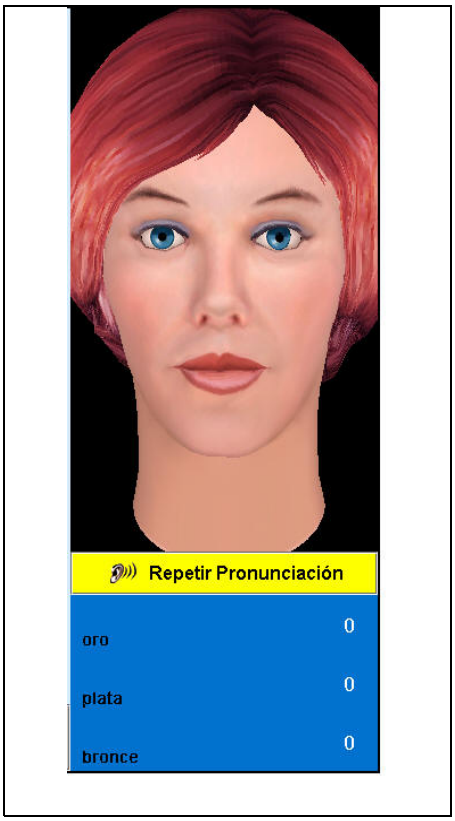

Figura 30. En esta se muestra los cambios hechos al código de JFourSquare.java para que el panel de monedas usara las palabras: “oro” “plata” y “bronce”. También se puede notar los cambios en la etiqueta del botón, que tiene la función de repetir la pronunciación de las palabras. Originalmente el botón mostraba la etiqueta “Again”. Otra característica en esta imagen es la actualización del nuevo agente animado que muestran los tutores.

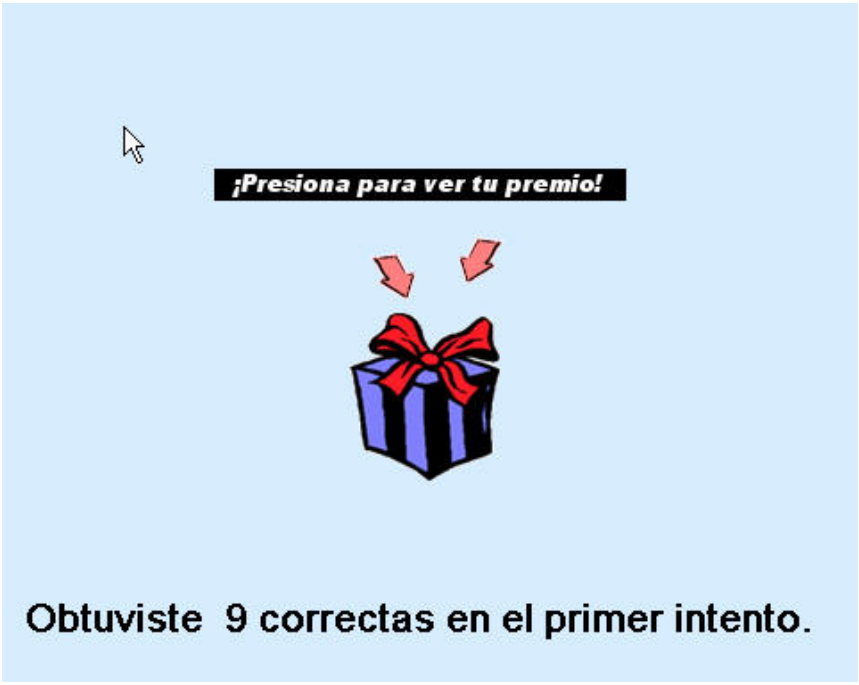

Figura 31. Oraciones de retroalimentación al final de cada nivel.

Las modificaciones al código de JFourSquare y JTutorFeedBack, así como el material insertado en la base de datos se pueden ver en el resultado de la interfaz. En cuanto a la implementación del sintetizador de voz en español, podemos decir que se logró el objetivo de que el tutor construyera las palabras que no se encuentran grabadas y almacenadas en la base de datos. Es decir existen palabras como “cable” cuya pronunciación en inglés está grabada en un archivo de sonido que se encuentra almacenado en la base de datos y que el tutor recupera para reproducirla, sin embargo esta misma palabra existe para el material en español, pero dado que si existe en la base de datos, el tutor no la manda al TTS para construirla. Así el tutor reproduce las grabaciones en inglés que recupera de la base de datos y sólo aquellas palabras que no le es posible encontrar como: México, tapete, Susana, queso, entre otras son construidas a partir del sintetizador de voz en español.

Para implementar de mejor manera el tutor en español y que éste reproduzca la pronunciación de las palabras completamente en español, es necesario grabarlas para posteriormente almacenarlas en la base de datos en una tabla exclusiva para sonidos del español. De tal manera que el tutor pueda recuperarlas y reproducirlas, sin necesidad de construirlas a partir del sintetizador de voz, lo cual es efectivo pero no optimiza tiempo, ya que el hecho de construir palabra por palabra requiere de diversos procesos que le restan velocidad al tutor.