

Capítulo 2: Análisis del Sistema del CSLR

2.1 Estructura del sistema del CSLR antes de la migración

En el apartado de este capítulo se explicará la estructura que tiene el sistema del CSLR, y la base de datos:

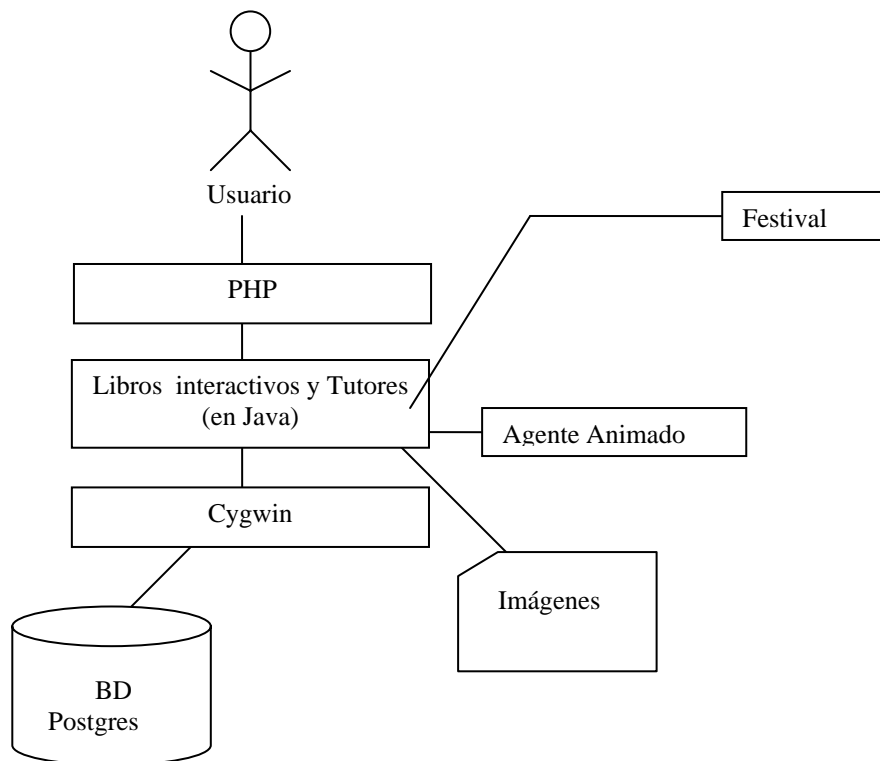


Fig 2.1.1 Estructura del Sistema del CSLR antes de la migración

En el diagrama de la Figura 2.1.1 podemos ver que el sistema utiliza Cygwin un emulador de Linux que sirve para poder tener la funcionalidad de Linux bajo el sistema operativo de Windows. En este caso, Cygwin es indispensable para poder acceder a la base de datos en Postgres. Postgres es el software para el manejo de la base de datos de toda la aplicación, los libros interactivos y tutores. Los programas en Java son los encargados de la administración de los datos obtenidos de la base de datos, el agente

animado, la voz del mismo; así como también la administración de las imágenes. Las imágenes que utilizan los tutores se encuentran almacenadas en directorios y algunas de las que utilizan los libros interactivos se encuentran almacenadas dentro de la base de datos. Los documentos en PHP se encargan del manejo de las sesiones y Festival es el sintetizador de voz, que proveer la funcionalidad de voz cuando no hay sonido pre-grabado; lo anterior requiere el JRE (Java Runtime Enviroment).

2.2 Análisis del funcionamiento interno del sistema

El funcionamiento del sistema se resume en el siguiente diagrama:

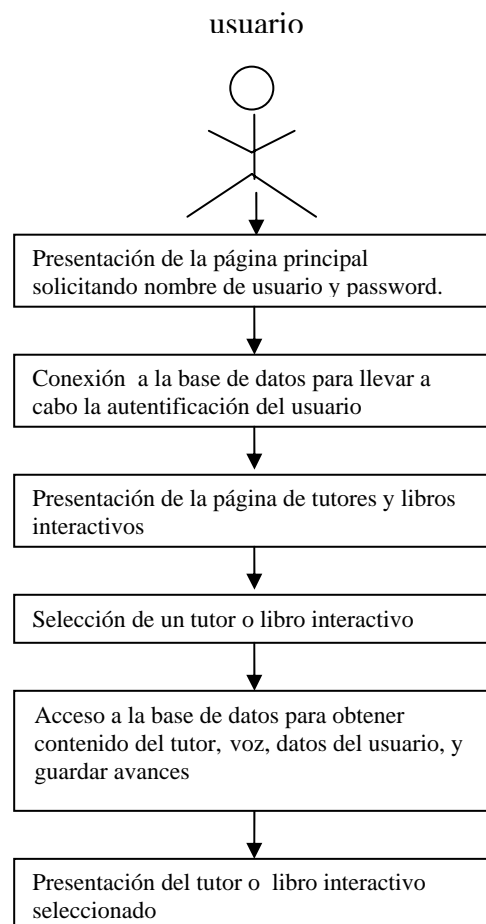
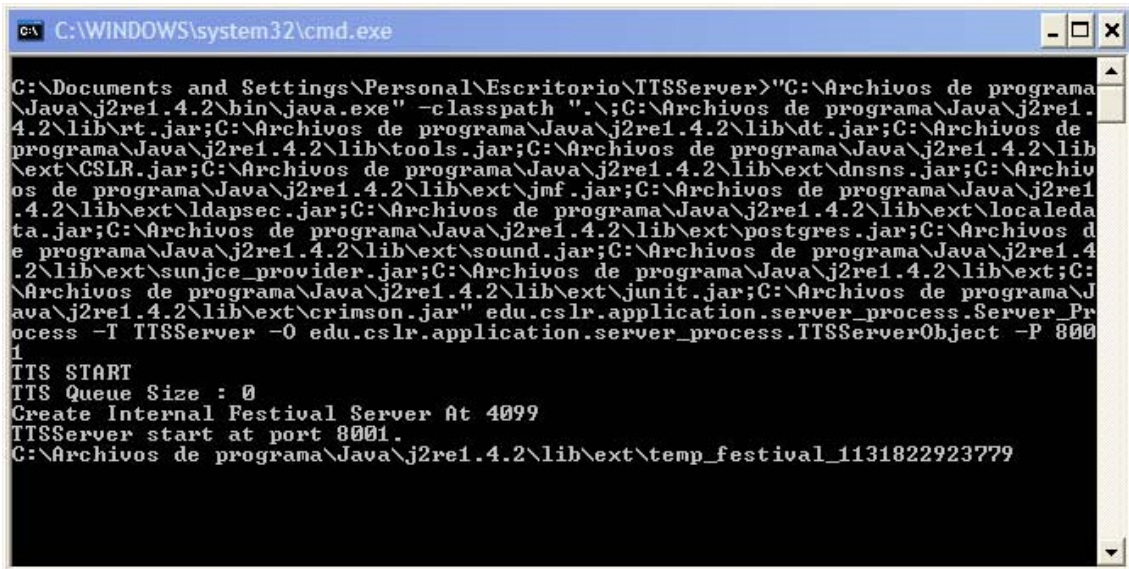


Figura 2.2.1 Diagrama de funcionamiento interno del sistema del CSLR

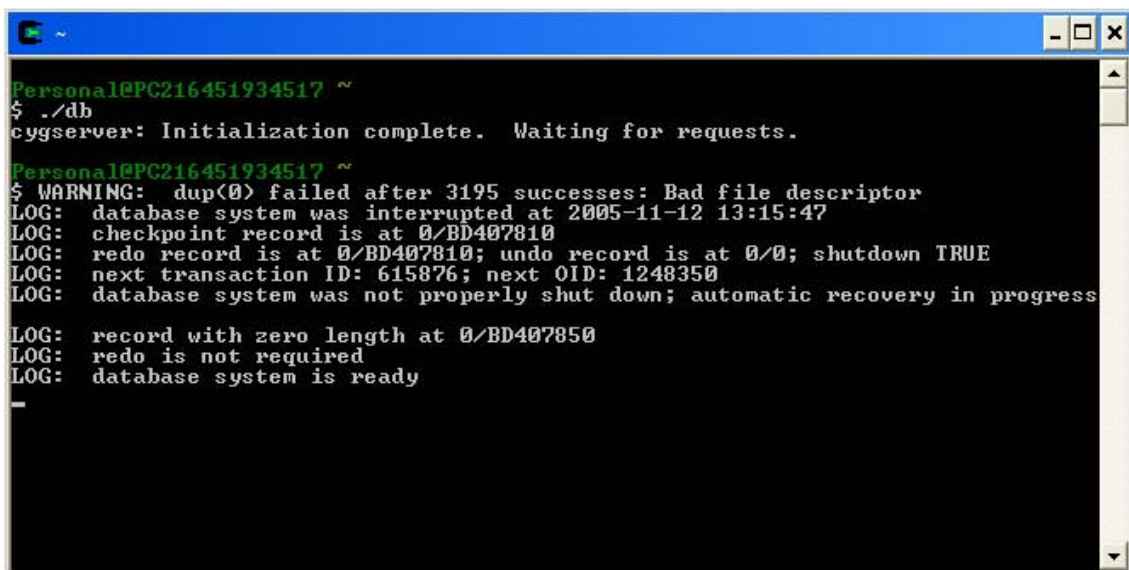
En primer lugar para poder echar a andar el sistema es necesario inicializar el TTSServer y la base de datos; para acceder a la base de datos es necesario abrir una

terminal de Cygwin; una vez abierta, es necesario teclear “./db” que inicializará la base de datos. Ambas acciones se muestran en la Figura 2.2.2 y 2.2.3 respectivamente.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Personal\Escritorio\TTSServer>"C:\Archivos de programa\Java\j2re1.4.2\bin\java.exe" -classpath ".\;C:\Archivos de programa\Java\j2re1.4.2\lib\rt.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\dt.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\tools.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\CSLR.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\dnsns.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\jmf.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\ldapsec.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\localedata.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\postgres.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\sound.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\sunjce_provider.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\junit.jar;C:\Archivos de programa\Java\j2re1.4.2\lib\ext\crimson.jar" edu.cslr.application.server_process.Server_Process -T TTSServer -O edu.cslr.application.server_process.TTSServerObject -P 8001
TTSS START
TTSS Queue Size : 0
Create Internal Festival Server At 4099
TTSServer start at port 8001.
C:\Archivos de programa\Java\j2re1.4.2\lib\ext\temp_festival_1131822923779
```

Figura 2.2.2 Pantalla de inicialización del TTSServer.



```
Personal@PC216451934517 ~
$ ./db
cygserver: Initialization complete. Waiting for requests.
Personal@PC216451934517 ~
$ WARNING: dup(0) failed after 3195 successes: Bad file descriptor
LOG: database system was interrupted at 2005-11-12 13:15:47
LOG: checkpoint record is at 0/BD407810
LOG: redo record is at 0/BD407810; undo record is at 0/0; shutdown TRUE
LOG: next transaction ID: 615876; next OID: 1248350
LOG: database system was not properly shut down; automatic recovery in progress
LOG: record with zero length at 0/BD407850
LOG: redo is not required
LOG: database system is ready
```

Figura 2.2.3 Pantalla de inicialización de la base de datos.

Una vez que se inicializó la base de datos podemos acceder a la página principal del sistema a través del navegador Internet Explorer; página que nos solicita un nombre de usuario y un password (Fig 2.2.4).

Esta página es index2.php. De manera mas detallada podemos decir que el archivo de index2.php es un archivo PHP que es el encargado de conectar el sistema y cargar la página principal del sistema, con la ayuda de otros archivos de configuración, tales como system.inc, forms. inc y sql.inc.

En caso de que el password sea incorrecto nos mostrará un mensaje de error y solicitará nuevamente la información; pero si dicha autenticación se llevó a cabo con éxito, entonces nos muestra la pantalla de los tutores y libros interactivos Figura 2.2.6.

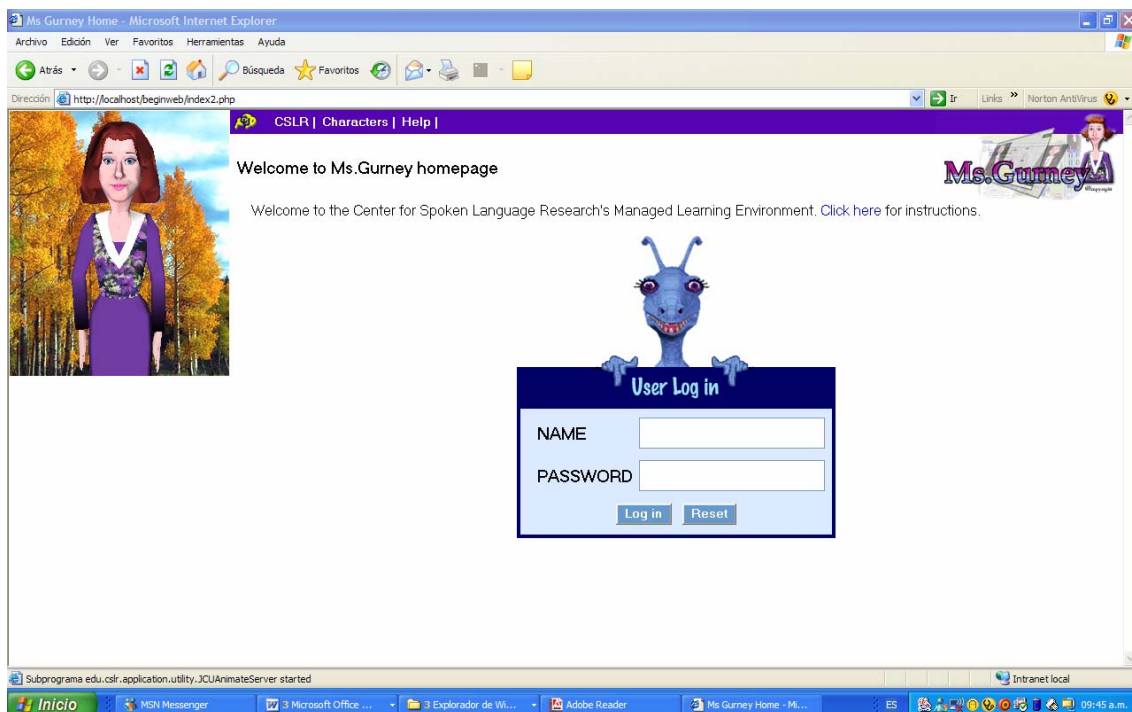


Figura 2.2.4 Index2.php

Los archivos de configuración son invocados desde el archivo index2.php mediante las instrucciones:

```
include "./lib/inc/system.inc";
```

```
include "./lib/inc/forms.inc";
```

```
include "./lib/inc/sql.inc";
```

y que se explicarán a continuación:

system.inc: Es un archivo de configuración que contiene funciones que se encargan de llevar a cabo una autenticación del usuario que desea entrar al sistema; así como determinar el estado de las sesiones de cada usuario, hacer consultas a las múltiples tablas, todo esto en el lenguaje de PHP.

forms.inc: Archivo en lenguaje PHP encargado de desplegar las ventanas, menús y submenús de la página principal, encargados de permitir al usuario la selección del personaje con el cual trabajará durante todo el tiempo que dure su sesión y de recuperar la página de presentación de los tutores.

Sql.inc :En este archivo se encuentran funciones propias del lenguaje sql que son utilizadas para el manejo interno de la base de datos y que son invocadas en el archivo system.inc

La página principal del sistema Index2.php contiene un menú en donde el alumno puede escoger el personaje con el que desea trabajar durante esa sesión, así como un cuadro de texto en donde el alumno debe introducir su nombre de usuario y su password para que el sistema lo identifique. Una vez que se llevó a cabo la autenticación del usuario, se carga la página de los tutores y libros interactivos de acuerdo al historial del usuario(véase Figura 2.2.5 y 2.2.6).

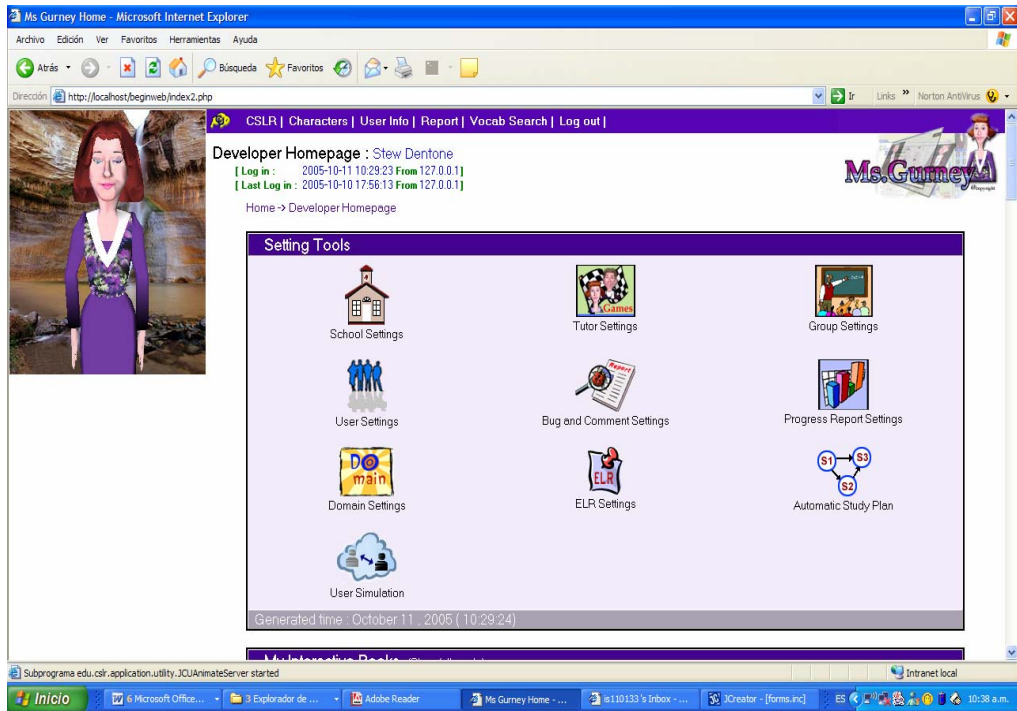


Figura 2.2.5 Pantalla de Presentación de los tutores y libros interactivos.
(vista del desarrollador)

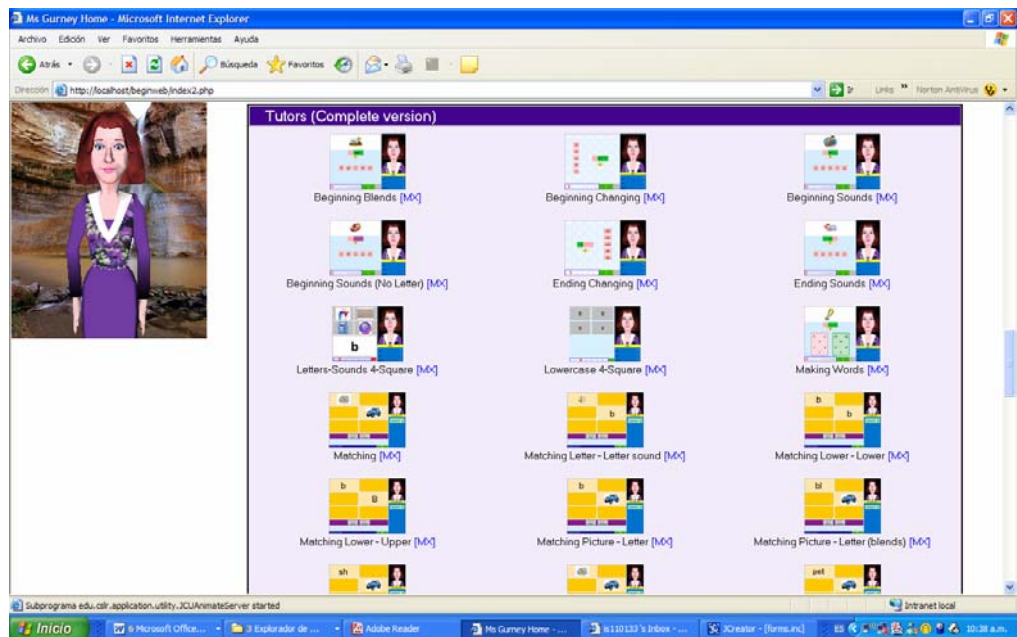


Figura 2.2.6 Pantalla de Presentación de los tutores y libros interactivos.

Cuando el usuario hace la selección de un tutor, el sistema carga los archivos correspondientes a dicho tutor, los cuales se componen de:

- a) Un archivo JAR que contiene todos los códigos que el tutor necesita, los cuales se encargan de controlar la información que presentará en cada actividad, dibujar la pantalla de la actividad, cargar las imágenes que utilizará etc;
- b) Uno o varios archivos xml que contienen los comandos que serán enviados al sintetizador de voz para que el tutor pueda a través del agente animado darle al alumno una retroalimentación, tanto en sus aciertos, como en sus errores.
- c) Un archivo PHP para el manejo de la sesión con el usuario.

Como ejemplo se muestra el tutor JMatchingWords. Este cuenta con dos archivos llamados intro.xml y JMatchingWords.xml que contienen los comandos que serán enviados al sintetizador de voz, un archivo llamado JMatchingWords.php para el manejo de la sesión con el usuario y un archivo llamado JMatchingWords.jar que contiene los archivos .class necesarios para el funcionamiento del tutor.

En la Tabla 2.2.7 se muestran los archivos contenidos en el JMatchingWords.jar; el cual se encuentra firmado. La firma de archivos .JAR se explica en el tema “3.4 Generación de un archivo JAR y proceso para firmar el mismo.” del capítulo 3 de éste documento.

| | |
|--------------------------|--------------------------------------|
| alphabetData.class | JMatchingWords\$25.class |
| JBasicEvent.class | JMatchingWords\$26.class |
| JBasicLog.class | JMatchingWords\$27.class |
| JCoinObj.class | JMatchingWords\$3.class |
| JMatchingWords\$1.class | JMatchingWords\$4.class |
| JMatchingWords\$10.class | JMatchingWords\$5.class |
| JMatchingWords\$11.class | JMatchingWords\$6.class |
| JMatchingWords\$12.class | JMatchingWords\$7.class |
| JMatchingWords\$13.class | JMatchingWords\$8.class |
| JMatchingWords\$14.class | JMatchingWords\$9.class |
| JMatchingWords\$15.class | JMatchingWords\$demoThread.class |
| JMatchingWords\$16.class | JMatchingWords\$JAnswerLabel.class |
| JMatchingWords\$17.class | JMatchingWords\$JButtonDrag.class |
| JMatchingWords\$18.class | JMatchingWords\$JLabelBut.class |
| JMatchingWords\$19.class | JMatchingWords\$JWordBut.class |
| JMatchingWords\$2.class | JMatchingWords\$mouseControl.class |
| JMatchingWords\$20.class | JMatchingWords\$playbackThread.class |
| JMatchingWords\$21.class | JMatchingWords.class |
| JMatchingWords\$22.class | JWordLevel\$wordSequence.class |
| JMatchingWords\$23.class | JWordLevel.class |
| JMatchingWords\$24.class | wordData.class |

Tabla 2.2.7 Lista de archivos del JMatchingWords.jar

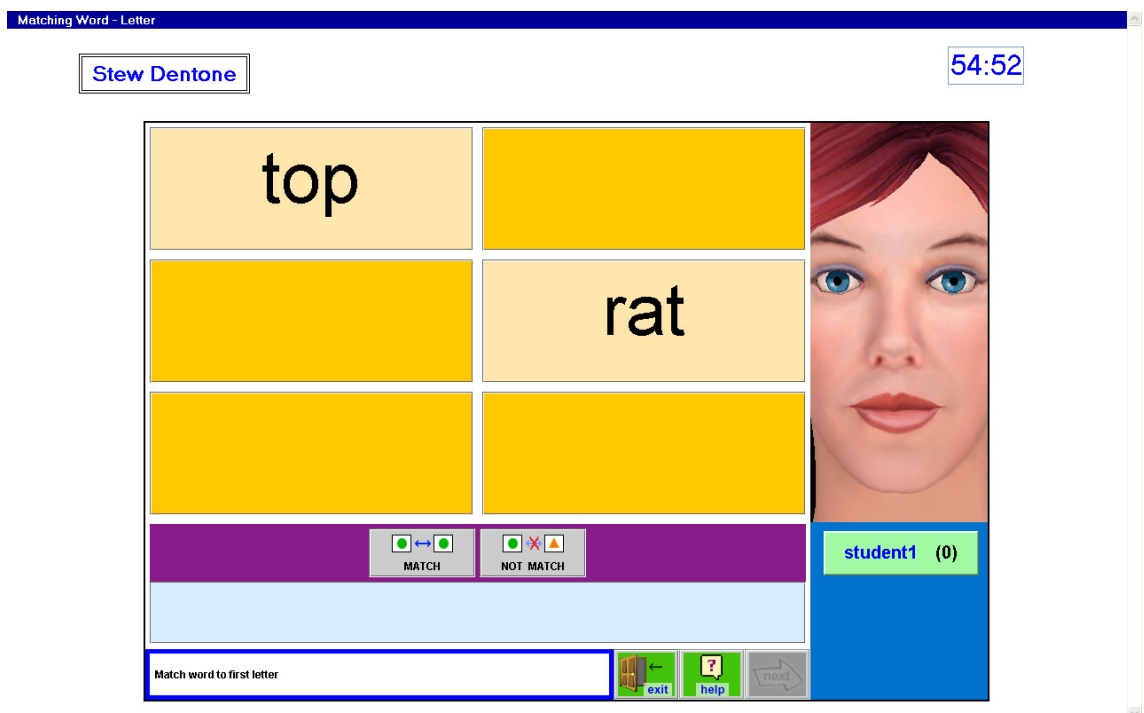


Figura 2.2.8 Interfaz del juego JMatchingWords.

La figura 2.2.8 muestra la interfaz del tutor JMatchingWords; cabe señalar que todos los tutores hacen llamados a un archivo llamado CSLR.jar que se encarga de

realizar la conexión de la base de datos de Postgres, así como de controlar el funcionamiento del TTSServer. En total el sistema del CSLR cuenta actualmente con aproximadamente 32 libros y 55 tutores interactivos.

2.3 Descripción de la base de datos

La base de datos cuenta con 104 tablas y 169 vistas cuales contienen las diferentes vocales y palabras utilizadas para las actividades de los tutores interactivos.

En el anexo de éste documento se incluye un diagrama entidad-relación de las tablas involucradas en el sistema; también el anexo se da una descripción de los atributos de cada una de las tablas, en donde se muestran los campos que las componen, cuales son sus llaves, y sus índices.

Debido a la complejidad de la base de datos se trató de identificar las tablas mas importantes y describir la función que desempeñan dentro del sistema.

Se pudieron identificar las tablas en donde son almacenados los sonidos que utilizarán tanto los libros como los tutores, las cuales son:

assess_sound_default,

book_sound_default,

global_book_sound_detail,

global_sound_detail,

nword_sound_default,

nword_sound_mx,

phoneme_sound_default,

phoneme_sound_mx.

Los tutores recuperan las imágenes de carpetas, sin embargo los libros interactivos si recuperan imágenes de la base de datos; las tablas en donde se encuentran almacenadas éstas imágenes, junto con otra información de los tutores son object_book, story_info y las vista all_stream y full_story_chapter_page.

La información del usuario correspondiente a su historial de acceso a los tutores así como información personal es almacenada en las tablas user_login, user_login_history, user_type.

Las tabla principal de donde los libros interactivos obtienen la información de las lecciones y las preguntas para verificar la comprensión de las mismas es comprehensive_dialog.

Los tutores requieren de conjuntos de palabras para presentarlos en cada actividad, y este conjunto de palabras es recuperado de las tablas:

words_cmu,

words_four_square,

words_level_detail,

words_rhyme_changing,

words_tutor,

words_vowel_finding,

y todas las vistas a excepcion de:

log_tutor_full_detail,
full_story_chapter_page y
all_stream.

Las imágenes y los sonidos de las tablas se encuentran almacenados de forma binaria; no los almacena directamente en las tablas, sino que en vez de eso los almacena en otra tabla aparte y utiliza un identificador también llamado OID (Object Identifier) para referirse a ellos.