

UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERIA ELÉCTRICA

PLATAFORMA DE APRENDIZAJE DE SEGUNDO IDIOMA CON ASR

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRICISTA

JOSÉ LUIS OLIVARES COTAL

PROFESOR GUÍA:

NESTOR BECERRA YOMA.

MIEMBROS DE LA COMISIÓN

JORGE WUTH SEPÚLVEDA.

CARLOS MOLINA SÁNCHEZ.

SANTIAGO DE CHILE

ENERO 2013

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: JOSE LUIS OLIVARES COTAL.
FECHA: 29 DE ENERO 2013.
PROF.GUÍA:SR. NESTOR BECERRA Y.

PLATAFORMA DE APRENDIZAJE DE SEGUNDO IDIOMA CON ASR

Las redes de datos que permiten acceder a la información desde cualquier lugar del mundo están cada vez más cerca del público en general. Este hecho, asociado al empleo de browser o navegadores ampliamente difundidos y de fácil uso, posibilita el desarrollo de sistemas de enseñanza en línea.

Esta memoria se centra en el desarrollo de una plataforma de enseñanza basada en las redes y sistemas Web. Se emplea la técnica de Reconocimiento Automático de Voz (*Automatic Speech Recognition, ASR*), una base de datos relacional llamada Postgresql y un servidor HTTP cuyo nombre es Tomcat. Para el desarrollo se emplearon los modelos Cliente Servidor, Vista Controlador y la Arquitectura AJAX. La estructura de la base de datos y las lecciones se diseñaron de manera que futuras revisiones se puedan realizar con facilidad. Se destaca el empleo de un formato ligero y compatible para la transferencia de datos por medio de *JavaScript Object Notation*. El programa permite el almacenamiento de las lecciones, el registro de la información de uso y el envío-recepción de datos asincrónicos entre el servidor y el cliente.

El trabajo finaliza con un análisis de estabilidad y robustez del sistema por medio de la carga de 150 usuarios simultáneos. Para realizar esta prueba fue empleado Jmeter, un software gráfico para el análisis de carga y desempeño de sistemas Web. Como trabajo a futuro se propone el empleo de HTML5 como lenguaje de reemplazo del Applet, para el envío de los datagramas de voz. El sistema al estar basado en servicios se puede configurar para trabajar en paralelo de modo que se está frente a un sistema escalable a nivel de servicios.

AGRADECIMIENTOS

A mis padres, quienes me han entregado su cariño y apoyo incondicional durante estos años.

A Muriel mi paciente pareja que me ha querido y soportado este tiempo.

A mi Hermanita, que sin su permanente apoyo no habría funcionado esta empresa.

Y a mis hijos, quienes han pagado al fin de cuenta los costos.

Tabla de contenido

1. INTRODUCCIÓN.....	1
1.1. Marco general.....	1
1.2. Presentación del Tema	2
1.3. Descripción de Contenidos.....	3
1.4. Objetivos	4
1)Objetivos Generales.....	4
2)Objetivos específicos.....	4
3)Metodología	5
2.CONTEXTUALIZACIÓN.	5
2.1. Modelos de referencia.....	6
2.2.Principios del modelo de capas.....	7
2.3.Definiciones del modelo de capas.....	7
1)Capa física.....	7
2)Capa de enlace de datos.....	7
3)Capa de red.....	8
4)Capa de transporte.....	8
5)Capa de sesión.....	8
6)Capa de presentación.....	8
7)Capa de aplicación.....	8
2.4.Transmisión de datos OSI.....	9
2.5.Protocolo TCP/IP.....	9
1)Modelo de Capas TCP/IP.....	10
2)Características del protocolo.....	11
2.6.Redes de computadoras.....	14
1) Dispositivos de redes.....	14
2) Tipos de redes.....	15
3)De acuerdo a su extensión.....	16
2.7.Arquitectura de Internet.....	17
1)Direccionamiento	17
2)Subredes.....	18
2.8.Servicios en Internet.....	19

1)Servicio de Resolución de Nombre (DNS).....	19
2)Servicio http (hypertext transfer protocol).....	19
2.9.Arquitectura Cliente-Servidor.....	20
2.10.Arquitectura Modelo Vista Controlador.....	21
2.11.Arquitectura Asynchronous JavaScript And XML (AJAX).....	22
HTML : Hyper text Marckup Language.....	23
JavaScript.....	23
DOM: Document Object Model.....	23
1) XML: Extensible Markup Language.....	24
2)Librería JQuery.....	24
a)Funciones y métodos básicos de JQuery.....	25
3)Funciones Ajax.....	26
4)Datos nativos JavaScript (JSON).....	27
2.12.Servidor Tomcat.....	28
1)Contenedores de Servlets Stand-alone (Independientes).....	29
2)Contenedores de Servlets dentro-de-Proceso.....	29
3)Contenedores de Servlets fuera-de-proceso.....	29
2.13.Servidor PostgreSQL.....	30
1)A continuación mostramos una lista con las principales características.....	30
2) Archivos de configuración.....	30
3) Algunas utilidades.....	31
2.14.Servidor ASR (Automatic Speech Recognition).....	31
3. IMPLEMENTACIÓN.....	32
3.1.Características generales del sistema.....	32
3.2.Componentes del Sistema.....	34
1)El Servidor de la plataforma.....	34
a)Servicio Http.....	34
b)Base de datos.....	35
c)Servicio ASR.....	35
d)Servicio de Dictado.....	35
2)Cliente.....	36
3)Base de Datos.....	36
4)Estructura de la base de datos.....	39

3.3.Aplicación WEB.....	40
1)Definición de usuarios de la plataforma.....	40
2)Implementación en la Base de datos.	41
3)Módulo del estudiante.....	41
4)El módulo del docente.....	44
5)El módulo del administrador de escuela “utp”	45
6)Administración general del sistema.....	46
3.4.Lenguajes y archivos de la aplicación	48
1)Lenguaje Html.	48
2)Lenguaje JavaScript.	48
3)Java Server page.	49
4)Servlet.....	49
5)Applet Java.	49
3.5.Implementación, funcionamiento y estructuras.....	49
1)Funcionamiento de la plataforma.	50
2)Código JavaScript	51
a)Identificación de los componentes.	51
b)Definición de evento.....	51
c)Estructuras y arreglos.....	53
d) La mnemotecnia del campo valor.....	56
e) Acceso a la BD desde JavaScript.	56
f)Acceso a la BD desde páginas JSP.....	57
4.PRUEBA DE CARGA.....	60
4.1.El software empleado en las pruebas.....	60
4.2.El Hardware para la realización de las pruebas.....	61
4.3. La prueba de rendimiento.....	61
4.4.Discusión.....	62
5.CONCLUSIONES.....	65
Bibliografía	66

Índice de figuras

Figura 1: Modelo de capas OSI.....	6
Figura 2: Modelo de capas OSI.....	9
Figura 3: Modelo de Capas TCP/IP.....	11
Figura 4: Tramas TCP/IP.....	13
Figura 5: Protocolos y redes Modelos TCP/IP.....	14
Figura 6: Topología de red. a) y b) red de bus. c) Estrella. d) topología de árbol.....	16
Figura 7: Modelo Cliente Servidor.....	21
Figura 8: Modelo Vista Controlador.....	22
Figura 9: Cliente servidor.....	34
Figura 10:: Esquema del Servidor.....	35
Figura 11:Descripción Funcional del cliente Web.....	36
Figura 12: Descripción General de Sistema.....	38
Figura 13:Esquema de la Base de datos.....	40
Figura 14:Pagina de Acceso.....	41
Figura 15:Modulo del estudiante.....	42
Figura 16: Módulo de administración docente, en la parte superior se puede ver la asignación de cursos y en la inferior el resumen de notas del alumno.....	44
Figura 17: Menú administrador de escuela "utp";a) lista de Docentes; b)cursos asignados al profesor; c) lista de alumnos por curso;d) icono para agregar alumnos.....	45
Figura 18: Menú de administración general y algunas opciones.....	46
Figura 19: Algoritmo de ingreso de usuarios.....	47
Figura 20: Diagrama de Flujo de información.....	50

Índice de tablas

Tabla 1: Rango de IP por sub red y máscara.....	18
Tabla 2: Estado de avance.....	43
Tabla 3: Hardware servidor y cliente, para prueba de rendimiento.....	61
Tabla 4: Tabla de resultados Jmeter. Los 4 totales están medidos en [ms] y el rendimiento se encuentra en [s].....	64

CAPITULO 1

1. INTRODUCCIÓN.

1.1. Marco general

Desde sus comienzos, la red Internet y los servicios asociados a World Wide Web, se han empleado para distribuir información. Las ventajas que presentan ambos sistemas los hacen muy apropiados para aplicarlos en la enseñanza y el aprendizaje en general. Si a esto se suma que las conexiones a Internet son cada vez más asequibles y los formatos presentados en la Web son más variados, permitiendo enriquecer el contenido de la información con la incorporación de voz y video en línea, se está frente al desarrollo de sistemas de enseñanza realmente útiles que potencian la capacidad de aprendizaje de los estudiantes.

La educación en línea no garantiza por sí sola el proceso de aprendizaje, sin embargo, es una poderosa herramienta de apoyo sobre la que se espera que continúe y amplíe su desarrollo. La presentación de la información es un factor clave, ya que, ésta permite capturar la atención del estudiante. Además cuenta con un formato adaptativo que es, hoy en día, una cuestión de mucha importancia en el desarrollo Web. Permitir una rápida actualización y un fácil mantenimiento es parte de los problemas que se tienen que resolver a la hora de enfrentar el diseño de una nueva plataforma.

En la actualidad, la educación en línea es un problema multidisciplinario, ya que, requiere un encuentro de distintos profesionales de áreas diversas como: diseñadores de software, psicólogos expertos en enseñanza, expertos en prácticas pedagógicas, etc. La concurrencia de los profesionales unidos al proceso evaluación deben permitir una evolución permanente en el software para que éste de cuenta de las necesidades concretas de los estudiantes cuyo aprendizaje debe ser el objetivo de esta plataforma.

En este contexto, desarrollar en conjunto con el Laboratorio de Procesamiento y Transmisión de Voz (en adelante LPTV) del departamento de Ingeniería Eléctrica de la Universidad de Chile, una plataforma de enseñanza de segundo idioma, que posee características

de registro y almacenamiento de datos generados por el usuario, parece ser un desarrollo adecuado a las tecnologías actuales. Permite además aumentar los conocimientos de los miembros del laboratorio, en el uso de bases de datos, la renovación de las librerías empleadas en el manejo de eventos y transmisión de datos asincrónica, es sin duda una contribución por parte de esta memoria.

El avance en el reconocimiento de palabras por métodos automatizados permite el desarrollo de un sin número de plataformas de enseñanza, ya se han estas en dispositivos móviles o en WEB. El LPTV ha desarrollado un programa a nivel de servicio TCP/IP que reconoce palabras provenientes de un diccionario de palabras correctamente pronunciadas, las cuales se integran de manera transparente a la plataforma de aprendizaje. Al ser este servicio un reconocedor de palabras al que se accede por red, la configuración de la plataforma puede contener cualquier idioma, siempre que este en concordancia a las lecciones que se encuentran almacenadas en la base de datos.

En el trabajo desarrollado se emplearon lecciones de la lengua inglesa por lo que el reconocedor de palabras posee un diccionario adecuado para las lecciones.

Hay en la actualidad una gran cantidad de empresas que venden el servicio de inglés en línea, sin embargo son pocas las que emplean técnicas de reconocimiento de voz y la principal diferencia con ellas es que basta cambiar las lecciones y el diccionario de palabras para que ya se pueda tener en funcionamiento para un idioma distinto o ampliado.

1.2. Presentación del Tema

En esta memoria se ha desarrollado una integración de sistemas de Base de Datos (BD), Procesamiento de Voz, Web y AJAX para desarrollar una plataforma orientada a la enseñanza de un segundo idioma. Para ello, se emplearon lecciones de inglés para los cursos de 5° a 8° básico. El servicio de procesamiento de palabras fue desarrollada por el LPTV y se integra al sistema por medio de socket TCP/IP que emplea los métodos de reconocimiento de patrones, extracción de parámetros y definición de clases previamente entrenadas. Junto con rutinas elaboradas en Java para la captura de datos por parte del cliente (Applet) y un servlet para el manejo de socket

en el servidor (Servlet). Este conjunto de tecnologías se integraron en el sistema denominado “English Loud” (Eloud), el cual fue desarrollado en esta memoria.

Es necesario mencionar que se debe conocer un sin número de lenguajes de programación para desarrollar el trabajo de integración, ya que cada parte constituyente del sistema se maneja en forma independiente. Para la base de datos se emplea SQL y PL/SQL en consultas y procedimientos de almacenado; en la parte WEB, HTML-JavaScript; JSP para el manejo dinámico de datos en el servidor y en los Servlet y Applet se emplea JAVA.

1.3. Descripción de Contenidos

En el Capítulo 2 se introduce al lector en la parte conceptual del desarrollo de redes, modelos de referencia y de capas. Se profundizan los conocimientos en los protocolos TCP/IP con la idea de entender la forma en que se transmite información por la red, a la vez de conocer los mecanismos y datagramas de transmisión. Se desarrolla en forma introductoria el tema de redes de computadoras, la distintas topologías, su clasificación y la arquitectura de Internet. Se verán los modelos conceptuales que sustentan el desarrollo del sistema: el Modelo Cliente Servidor (MCS) que tan extensamente se emplea; el Modelo Vista Controlador (MVC) que separa las distintas partes desde el punto de vista manejo lógico almacenamiento de datos y visualización. Posteriormente, se analizarán las diferencias entre los modelos de transmisión de datos Web tradicionales versus el modelo AJAX. Para terminar con una breve introducción a la librería JQuery. Se finalizará el capítulo con la presentación de los distintos servidores involucrados, sus principales archivos de configuración y las características que los destacan.

En el Capítulo 3 se aborda la implementación del sistema, la estructura de la BD, los distintos tipos de usuarios que pueden acceder al sistema, la estructura de las lecciones y partes del código empleado para una mejor comprensión de los lenguajes y librerías. Se analiza el funcionamiento lógico del sistema por medio de esquemas, finalizando con una muestra de las páginas principales de English Loud.

En el Capítulo 4 el análisis se dirige hacia los resultados de estabilidad y confianza para 150 usuarios simultáneos, los cuales fueron simulados por procesos en paralelo a partir de un software llamado Jmeter, el cual permite la medición de rendimiento de sistemas Web. Este

programa fue realizado por la Apache Foundation y es ampliamente utilizado para medir el desempeño de servidores Web. Es software libre y tiene que ser ejecutado en una computadora con la capacidad de mantener los distintos procesos simultáneos.

En el Capítulo 5 se realiza el análisis de los datos obtenidos, se presentan las principales conclusiones del trabajo y se sugieren innovaciones futuras para el software.

1.4. Objetivos

1) Objetivos Generales

Implementar y evaluar un sistema de enseñanza WEB empleando los modelos: cliente-servidor, vista-controlador, Ajax y el servidor de reconocimiento de voz del LPTV del Departamento de Ingeniería Eléctrica de la Universidad de Chile

2) Objetivos específicos

- Implementación y configuración de un servidor Web basado en Tomcat.
- Realización de la configuración y definición de una de base de datos Postgresql.
- Elaboración de un sistema de transmisión de datos asincrónicas con tecnología AJAX y JSON.
- Desarrollo de un sistema Web con JavaScript, Java , html y JQuery.
- Creación de estructuras de datos que permitan un sistema multi-usuario, multi-escuela y multi-lección.
- Disposición de un sistema Web de administración de escuelas y usuarios.
- Confección de un servidor prototipo que permita dar soporte a las lecciones de 5° a 8° básico con reconocimiento de voz basado en el servidor ASR, Tomcat, Postgresql y AJAX.
- Evaluación del sistema Web-Base de datos-Reconocimiento de Voz.
- Realización de pruebas de carga del servidor empleando la herramienta Jmeter de la Apache Software Foundation.

3) Metodología

- Primera etapa: puesta en marcha de los servidores Tomcat, Postgresql y ASR.
- Segunda etapa: definición de la estructura de la base de datos y el funcionamiento general del sistema.
- Tercera etapa: estructura de datos JSON para las lecciones y usuarios.
- Cuarta etapa: prueba mínima de funcionamiento del sistema; resumen de avance en las lecciones; notas; usuarios y procesamiento de voz en línea.
- Quinta etapa: almacenamiento de las actividades de los usuarios y de las señales de voz.
- Sexta etapa: prueba de 5º básico completa.
- Séptima etapa: sistema Web de administración de usuarios.
- Octava etapa: prueba completa de los cursos 5º a 8º básico y administración de usuarios.
- Novena etapa: prueba de carga y tiempos de respuesta.

CAPITULO 2

2. CONTEXTUALIZACIÓN.

En la red se encuentran una gran cantidad de cursos en línea de distintos idiomas, los cuales sin duda hacen uso de los elementos multimedia que poseen los navegadores. Estos cursos principalmente se basan en la escucha y repetición de palabras o frases, sin que la correcta pronunciación se pueda verificar. Una vez realizada la lección se asigna tutores para realizar la corrección del habla, esto encarece el sistema ya que requiere tener una persona atenta a las solicitudes de los usuarios. Otro tipo de sistemas son los que emplean técnicas de reconocimiento de voz, los cuales en general son programas que incluyen lecciones completas y se instalan a nivel de usuario. La plataforma que se presenta en esta memoria se caracteriza por realizar el reconocimiento de las frases y palabra en un servidor, al cual se accede por medio del protocolo TCP/IP en forma remota, este posee una base de datos donde se almacenan las lecciones, el

avance del usuario y las palabras pronunciadas. Adicionalmente se guardan las evaluaciones que obtiene el estudiante, por lo que estamos frente a un sistema bastante completo y extensible, ya que centralmente se pueden cambiar las lecciones o el idioma sin necesidad de intervención de los usuarios.

En este capítulo se verá los modelos de referencia, protocolos, conceptos de redes, servicios y el funcionamiento de las distintas partes que permiten establecer la comunicación en Internet. Se dará una explicación de los patrones de diseño que fueron empleados en la aplicación desarrollada en esta memoria, ya sea, su modelo conceptual como práctico.

2.1. Modelos de referencia

En el año 1984 se estableció un modelo de referencia para la interconexión de sistemas abiertos O.S.I (Open System Interconetion) el cual fue creado por la Organización Internacional para la Estandarización. Este modelo es muy similar a SNA de IBM que dominaba el mercado por aquel entonces. Posee 7 capas que, si bien es cierto, es un intento de estandarización de las redes, en la práctica sólo es usado para la enseñanza y permite mostrar la interoperatividad de distintos protocolos para establecer la comunicación. En la Figura 1 se puede apreciar cada una de las 7 capas del modelo OSI.

No es menester de esta memoria profundizar en el modelo OSI ni generar una discusión entorno a este, por lo que, sólo se definirán algunos principios del modelo de capas y se dará una breve descripción de cada una. La idea principal es entender como se puede resolver el problema de comunicación utilizando estas definiciones y principios.

2.2. Principios del modelo de capas

Cuatro son los principios fundamentales a tener en cuenta para la resolución de un problema mediante capas:

- El modelo debe tener tantas capas como nivel de abstracción se requieran.
- Cada capa realiza una función bien definida.
- La función debe estar implementada de acuerdo a protocolos estándares pre-definidos.

- Los límites de las capas deben minimizar el flujo de información entre ellas.

2.3. Definiciones del modelo de capas

1) Capa física

La capa física tiene que controlar la transmisión de los bits por el canal de comunicación, esto significa que debe verificar que cuando el emisor envíe un bit 1, el receptor lo reciba como bit 1 y no como bit 0.

- Se define el nivel de voltaje correspondiente al 0 lógico y al 1 lógico.
- El tiempo necesario para transmitir un bit.
- El tipo de transmisión simplex, duplex o halfduplex
- Cómo se establece la conexión inicial y cómo se finaliza.
- El tipo de conector a emplear, las especificaciones mecánicas, eléctricas, térmicas, etc. de las interfaces físicas que se utilizarán.

2) Capa de enlace de datos

La principal tarea de esta capa es tomar un medio de transmisión y transformarlo en una línea libre de errores. Esta capa define y transmite los frames en orden secuencial, procesa Acknowledgement y toma acciones cuando hay errores en la transmisión. Se preocupa de definir y detectar los límites de cada frame además de marcarlos para así poder reconstruir la información entregada a la capa de red.

Debe preocuparse de la recuperación de errores lo que se puede resolver mediante la re-transmisión de frames en caso de errores en la transmisión.

3) Capa de red.

Es responsable de controlar el funcionamiento de la subred de comunicaciones. Se preocupa del enrutamiento de los datos, es aquí donde se realiza el control de congestión de la red. En esta capa se realizan funciones de contabilidad, se pueden contar los paquetes, bytes o bits que un cliente transfiere por la red.

Es la que se preocupa de recuperar los errores de falla en los enlaces, realiza el balance de carga, etc.

4) Capa de transporte

La función básica de la capa de transporte es aceptar los datos de la capa de sesión, dividirlos en unidades más pequeñas si es necesario, pasarlos a la capa de red y asegurar que todos los trozos lleguen correctamente al otro extremo.

Es aquí donde los paquetes transmitidos por rutas distintas se ordenan y entregan a la capa de sesión.

5) Capa de sesión

Permite a los usuarios de máquinas diferentes establecer sesiones entre ellos. Establece mecanismo para controlar el diálogo, agrupa los datos de la manera que serán presentados y realiza la comprobación de estos en caso de tener que retransmitir una porción de ellos.



Figura 1: Modelo de capas OSI.

6) Capa de presentación

Realiza ciertas tareas como el manejo y adaptación de formatos de números enteros (complemento a uno o complemento a dos) o la representación de caracteres (ASCII o Unicode). La capa de presentación se preocupa de realizar automáticamente los cambios.

7) Capa de aplicación

Es la capa más cercana al usuario y no presta servicios a ninguna otra capa del modelo, en cambio, si presta servicios a las aplicaciones que están fuera del ámbito del modelo OSI. Estas aplicaciones pueden ser planillas de cálculo, procesadores de texto, terminales virtuales, etc.

2.4. Transmisión de datos OSI

En el esquema OSI se trata de resolver el problema de envío y recepción de información utilizando niveles que resuelven un problema particular, por ejemplo, dividiendo los datos en paquetes de determinado tamaño o asignándole un destinatario concreto. Para ello, se realiza comunicación en forma jerarquizada donde cada nivel agrega información de control que puede ser interpretada por un nivel equivalente en el receptor. Cada nivel es independiente de su predecesor, por lo que, la información que se recibe es tratada como dato y, por lo tanto, no se interpreta ni se modifica. En la Figura 2 se puede observar como los datos son ampliados por la información de control, la cual será suprimida en el receptor.

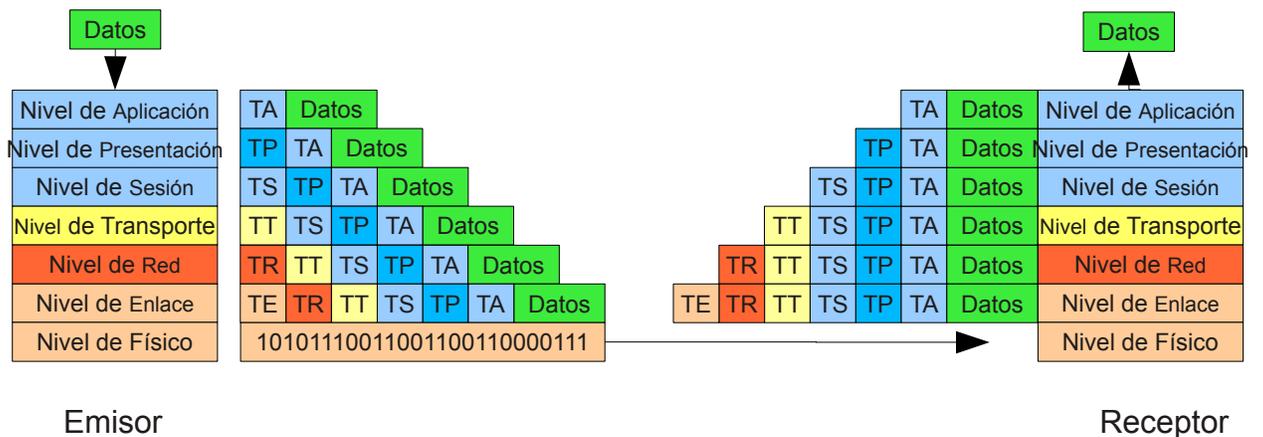


Figura 2: Modelo de capas OSI.

En la práctica, el modelo más ampliamente difundido es el TCP/IP razón por la cual, se analizará su modelo de funcionamiento, las capas en que está dividido, su definición y las aplicaciones en que se emplea.

2.5. Protocolo TCP/IP

En Internet hay un conjunto de protocolos que permiten la interconexión de redes heterogéneas y descentralizadas a nivel mundial. Este conjunto de protocolos son conocidos como TCP/IP y nacen a partir de la necesidad del departamento de defensa de EEUU de mantener las comunicaciones gubernamentales a pesar de que uno o varios nodos sufrieran un ataque nuclear. Junto a esto, se preveía la necesidad de poder enviar desde un archivo de texto hasta señales de voz en tiempo real.

Todos los requerimientos llevaron a que se adoptará una red de paquetes de datos no orientada a la conexión; esto es, que cualquier dispositivo conectado a la red pueda emitir paquetes dentro de ella y estos sean enviados a su destino. Así surge inter-red o Internet que define un protocolo oficial denominado IP (Internet Protocol). Es claro que este protocolo esta orientado al enrutamiento y entrega de paquetes.

Para permitir que dos dispositivos puedan establecer una comunicación se requiere un protocolo. La idea de este protocolo es que si un dispositivo emite un paquete a un destinatario, el destinatario pueda comunicar que se encuentra escuchando y entiende el mensaje. Los protocolos de este tipo son los denominados protocolos orientados a la conexión.

El protocolo TCP está orientado a la conexión y divide la información recibida en un flujo de Bytes que son traspasados a la capa IP y recibidos sin errores por el receptor TCP; junto a esto, se preocupa que la tasa de envío de paquetes no sature al receptor, por lo que, ajusta la velocidad a medida que envía la información. Este protocolo basa su funcionamiento en la fiabilidad del envío de datos, por lo cual, esta se emplea en aplicaciones donde la integridad de los datos es de primera importancia.

El protocolo de datos UDP (Protocolo de Datagrama de Usuario) es un protocolo diseñado para el rápido envío de información donde la integridad de los datos no es muy importante. Se emplea, por ejemplo, en transmisión de voz, ya que, el usuario final puede comprender el mensaje de voz aunque este no este completo.

Esta familia de protocolos es la que se encuentra más difundida en Internet, se centrará el análisis en TCP/IP, ya que, la aplicación hace uso extensivo de este procedimiento.

1) Modelo de Capas TCP/IP

Los protocolos TCP/IP posee, al igual que el modelo OSI, una forma modular pero basado en 4 capas, estas son:

- Capa de acceso a la red, Link o Física. Es la capa que se encarga del envío y recepción de los paquetes IP independiente de la red que se utilice.
- Capa de Internet. Es la encargada de la comunicación de datos entre máquinas. Se encarga de realizar la selección de la máquina hacia la que se envían los datagramas, maneja el flujo de datos, verifica la validez y responde a las peticiones de la capa de transporte.
- Capa de transporte. Es la encargada de entregar los mecanismos que permiten conocer el estado de la transmisión, permite la comunicación entre programas, se asegura que los datos lleguen sin errores y en la secuencia adecuada.
- Capa de aplicación. Es la encargada de interoperar con los programas del usuario, realiza peticiones a la capa de transporte para enviar y recibir datos en concordancia al tipo de petición de envío, ya sea, un mensaje individual o un flujo continuo de octetos.

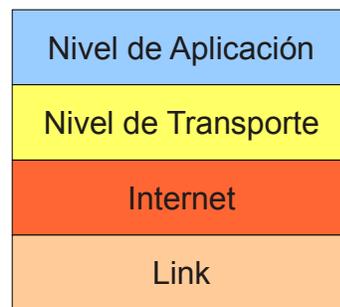


Figura 3: Modelo de Capas TCP/IP.

2) Características del protocolo

Al ser el protocolo IP orientado al envío de datagramas sin ninguna forma de verificación, el protocolo TCP lo dota de los elementos necesarios que una aplicación requiere para el envío fiable de datos. Estos elementos son:

- Protocolo orientado a la conexión. Permite la conexión de dos máquinas para envío y recepción de datos, el manejo de flujo de la información impidiendo de esta manera la saturación de la red por el envío excesivo de datagramas.
- Operación Full duplex. Quiere decir que la comunicación puede ser en ambas direcciones al mismo tiempo. Para ello, la capa de transporte puede manejar los datos enviados y el orden de llegada pidiendo un reenvío de la información si esta no pasa los mecanismos de verificación .
- Acknowledgements. Es el campo al que llega el receptor al recibir la información verificada de manera que el emisor pueda tomar decisiones de reenvío o continuación de la comunicación.
- Control de flujo. Si en el receptor se ve desbordado el buffer de datos por una transmisión demasiado rápida no llegarán de vuelta los ACK de los datos enviados, por lo que, se entenderá que es necesario ajustar el flujo hasta que todos los paquetes se reciba en ACK correcto.

Para un mejor entendimiento del protocolo TCP se puede ver a continuación el formato de segmentos TCP en el cual, se pueden apreciar los distintos campos necesarios para realizar la comunicación. En la Figura 4 se puede observar cómo cada capa empleada posee un segmento de la trama, el cual, permite comunicarse con su capa similar en el receptor como, por ejemplo, los puertos de envío y recepción, número de orden del segmento, etc.

	<i>Bits 0-3</i>	<i>4-7</i>	<i>8-15</i>	<i>16-31</i>
<i>0</i>	<i>Puerto de Origen</i>		<i>Puerto destino</i>	
<i>32</i>	<i>Número de secuencia.</i>			
<i>64</i>	<i>Número de acuse de recibo.</i>			
<i>96</i>	<i>Longitud cabecera TCP</i>	<i>Reservado</i>	<i>Flags</i>	<i>Ventana</i>
<i>128</i>	<i>Suma de verificación</i>		<i>Puntero</i>	
<i>160</i>	<i>Opciones + Relleno</i>			
<i>192</i>	<i>Datos</i>			

Figura 4: Tramas TCP/IP.

En la figura 5 se ven ejemplos de las distintas capas del protocolo y los distintos niveles de abstracción, se puede observar que no está especificada una capa física particular. Al analizarlo desde el punto de vista del modelo OSI se aprecia que no se adecua exactamente al número de capas. En OSI las capas de enlace, presentación y sesión no aparecen en TCP/IP, esto ha motivado una serie de discusiones sobre la validez del modelo de capas OSI, pero este tema se escapa del alcance de esta memoria.

Para mayor comprensión sobre el modelo OSI, el número de capas y su implementación se sugiere consultar el libro “Redes de computadoras de 4ª edición Andrew Tanenbaum”.

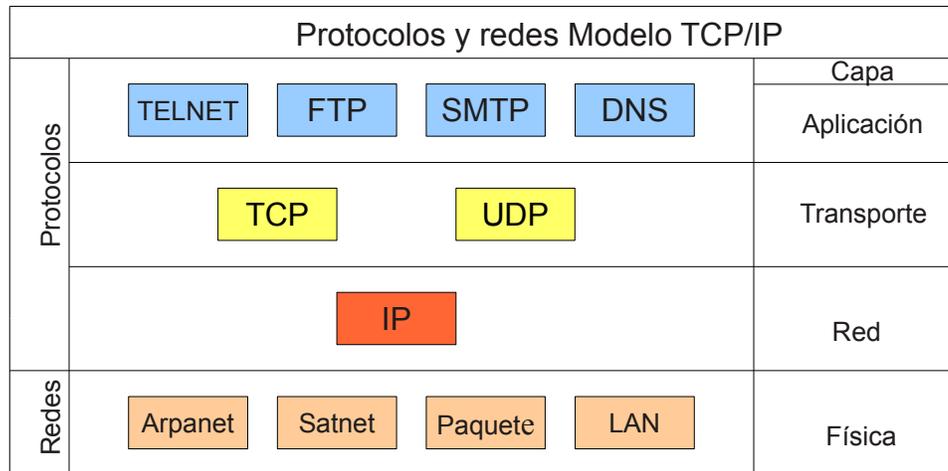


Figura 5: Protocolos y redes Modelos TCP/IP

2.6. Redes de computadoras

Se ha señalado la manera en que se establece la comunicación por medio de protocolos entre un emisor-receptor y cómo se pueden construir estos por medio de abstracción en capas. A continuación se mencionan los dispositivos que se emplean en la construcción de redes, su arquitectura y como se clasifican.

1) Dispositivos de redes.

En las redes de computadores hay una serie de dispositivos encontrando entre otros: hubs, repetidores, routers, bridges, switches.

- Hubs. Permiten la conexiones de varios dispositivos a una misma red propagando en forma de repetidor las señales recibidas, opera a nivel de capa 1. A diferencia de un repetidor que posee pocas puertas el hub posee varias, desde 5 a 24.
- Bridges- Este dispositivo realiza la separación lógica de 2 redes, opera a nivel de capa 2. Se emplea, en general, para poder traducir tecnologías de capas como puede ser Ethernet con Tokenring.
- Switches. Al igual que el Bridges, permite la separación lógica de redes, realiza micro segmentación manejando un segmento de red por puerta disminuyendo la colisión de paquetes de red. Opera a nivel de capa 2.
- Routers. Estos elementos trabajan a nivel de capa 3, son dispositivos diseñados para separar redes y permiten mejorar el empleo del ancho de banda, ya que, los paquetes internos de una red sólo pueden ser vistos por ésta no afectando al resto de las redes del entorno. Los routers son dependientes del protocolo.
- Gateways. Son equipos que permiten interconectar redes y arquitecturas distintas a todo nivel de comunicación.

2) Tipos de redes.

Hay distintas formas de clasificar las redes de comunicación, pero en general, se pueden clasificar de 3 formas: de acuerdo a la tecnología para realizar la transmisión, a su topología o a su extensión. De acuerdo a la tecnología de transmisión se clasifican en redes de difusión o punto a punto.

- Redes de difusión (broadcast). Las redes de difusión se encuentran conectadas a 1 solo canal de comunicación, esto permite que si una máquina envía un mensaje todas lo pueden escuchar, pero lo recibe sólo la máquina a la cual va dirigido.

- Redes punto a punto. Estas redes son las que en general conectan 2 máquinas entre sí o las que permiten la conexión entre muchos pares de máquinas. Esto significa que, para llegar de una máquina a otra, se puede eventualmente pasar por 1 o más máquinas intermedias.

De acuerdo a su topología pueden ser redes de bus, redes en estrella, redes en árbol, etc. Las topologías básicas se encuentran en la Figura 6.

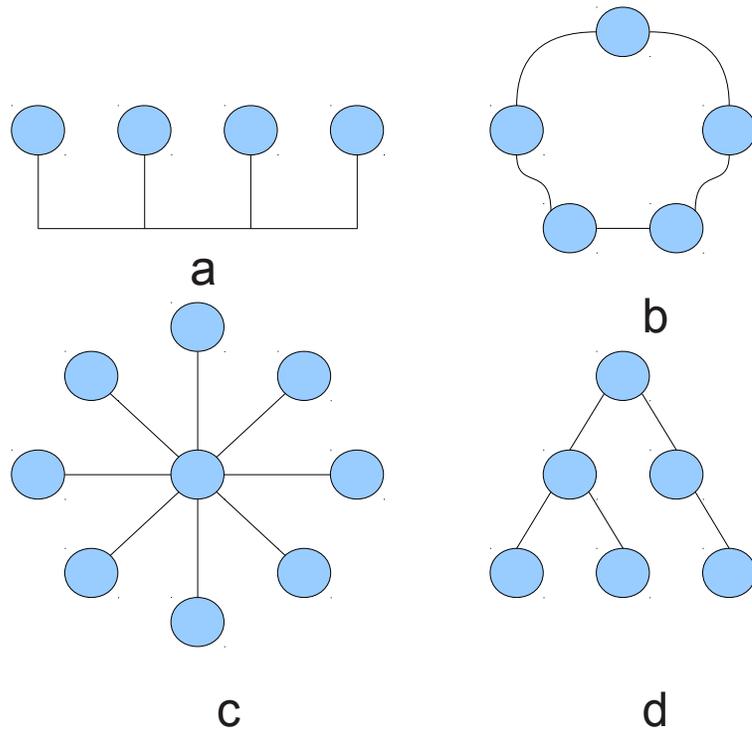


Figura 6: Topología de red. a) y b) red de bus. c) Estrella.
d) topología de árbol.

3) De acuerdo a su extensión

Una forma distinta de clasificar redes de datos es por medio de la extensión que poseen. Se pueden encontrar redes de ámbito personal hasta Internet. A continuación se enumeran redes por extensión:

- PAN (Personal Area Network). Las redes de área personal o PAN son las redes que se establecen en torno a una persona. Ejemplo de ella son las redes bluetooth, iR, etc.

- LAN (Local Area Network). Estas son las redes que se establecen en torno a una casa, oficina, un avión, etc.
- CAN (Campus Area Network). Son redes que se establecen en áreas determinadas como puede ser un campus universitario o una base militar.
- MAN (Metropolitan Area Network). Son redes de una cobertura geográfica amplia, suelen ser ciudades.
- WAN (Wide Area Network). Son redes informáticas que se extienden en forma extensa abarcan regiones o continentes.

2.7. Arquitectura de Internet

La Internet es un conjunto de protocolos con funciones determinadas, posee varias especificaciones tecnológicas, políticas y económicas que condicionan el uso de la red. Una de sus principales características es que es descentralizada lo que la hace más inmune a una serie de errores de funcionamiento o ataques. Al ser una red heterogénea que se dedica al envío y recepción de información la convierte en una red neutra, ya que, se limita al transporte de paquetes y no al análisis de contenido. Esta neutralidad de la red ha permitido su rápido desarrollo, puesto que, empresas y personas independientes colaboran permanentemente en investigación de nuevas aplicaciones y servicios.

1) Direccionamiento

Para poder comunicarse en Internet los equipos deben poseer un identificador único en la red, este identificador se denomina IP. Esta dirección está constituida por 4 números separados por un punto y van desde 0 a 255, esto es un número del tipo w.x.y.z. La IP esta organizada en 5 clases que determinan el tamaño y el tipo de red que se encuentra. Estas clases son:

- Clase A. Las direcciones de esta clase pueden identificar 16.777.214 máquinas en la red, por lo que, se emplea en grandes redes. Está representada por direcciones del tipo w.0.0.0 donde w, se encuentra entre 0 a 126.

- Clase B. Estas direcciones se encuentran en el rango de redes medias y pueden identificar 16.384 redes y emplea la nemotecnia de w.x.0.0 con w entre 128 a 192. Los 2 números restantes se emplean para identificar 65.534 computadores por red.
- Clase C. Esta clase se emplea para redes pequeñas y pueden identificar 2.097.152 redes con valores de w que va de 192 hasta 223, usando los valores w.x.y.0 e identifica hasta 255 computadores por red.
- Clase D y E. La clase D es empleada para multidifusión y la E para desarrollo futuro.

En la red de clase A, el primer número de la IP representa a la red. En las redes de clase B el primer y segundo número son los que representan la red. La red de clase C está representada por los 3 primeros números. Los números restantes de cada clase representas el computador correspondiente.

2) Subredes

Una red que posee mucho tráfico de sus computadores puede ver disminuido su rendimiento producto de que aumenta la colisión de paquetes presentes en ella, para ello, puede ser segmentada en redes más pequeñas o subredes. Para dividir una red en forma fisica se emplean routers, pero hay que identificar el segmento de red al que se pertenece. Para ello, se emplea una máscara de red que corresponde a un número w.x.y.z para el cual existe una nemotecnia.

Se conservan los valores numéricos referidos a la red con valores 11111111 equivalente a 255.

Se define el número de bits, los cuales tienen correspondencia con el número de redes, por ejemplo, si nuestra red es de clase C 193.144.238.0 y se quieren 8 redes entonces se necesitan 3 bit esto es 111. La máscara corresponde a 255.255.255.224 y los rangos de IPs de cada subred se observan en la Tabla 1.

1	193.144.238.0	193.144.238.1 a 193.144.238.30
2	193.144.238.32	193.144.238.33 a 193.144.238.62
3	193.144.238.64	193.144.238.65 a 193.144.238.94
4	193.144.238.96	193.144.238.97 a 193.144.238.126
5	193.144.238.128	193.144.238.129 a 193.144.238.158
6	193.144.238.160	193.144.238.161 a 193.144.238.190
7	193.144.238.192	193.144.238.193 a 193.144.238.222
8	193.144.238.224	193.144.238.225 a 193.144.238.254

Tabla 1: Rango de IP por sub red y máscara.

2.8. Servicios en Internet

Como se puede apreciar en la Figura 5 en la capa de aplicación aparecen una serie de programas que emplean el protocolo TCP/IP para entregar servicios a la red. Estos servicios son accedidos por medio de llamada a puertos determinados donde se encuentra la aplicación en cuestión. Ejemplos de estos servicios son DNS (53), SMTP(25) , HTTP(80), etc. Para acceder a ellos los programas realizan una llamada a un Socket que no es más que una abstracción del sistema operativo para poder comunicar 2 computadoras en Internet. Para el llamado de un Socket es necesario definir un par de direcciones IP, el puerto de llamada, destino y el protocolo a emplear.

En la presente memoria se emplearán una serie de servicios que se relacionan, ya sea, para poder resolver los nombres de servidores, para entregar el contenido o para analizar los datos.

1) Servicio de Resolución de Nombre (DNS)

Inicialmente Internet se componía de pocas máquinas a las cuales se les asignaba una dirección IP y un nombre, esto permitía referenciar rápidamente una máquina por su nombre a través de un listado que existía en el computador llamado host. A este listado accedían las aplicaciones para encontrar la dirección IP de computador. A medida que Internet fue creciendo, el número de máquinas aumentó y mantener en cada computador el listado se hacia inmanejable, por lo que, apareció la necesidad de tener un servicio que permitiera resolver los nombres. El servicio en la actualidad corresponde a una base de datos distribuida y jerarquizada, esto quiere decir, que se encuentra en forma parcial en cada servidor. Que sea jerarquizada quiere decir que hay máquinas que tienen mayor importancia y son las que se encargan de resolver un dominio en particular como, por ejemplo, el dominio .cl en Chile.

2) Servicio http (hypertext transfer protocol).

Es un servicio que implementa alguna de las versiones del protocolo, la más importante es la que se describe en document RFC 2616. Habitualmente se encuentra escuchando en el puerto 80 y es donde los navegadores lo buscan por defecto. Algunos de estos servidores son: APACHE,

TOMCAT, IIS, etc. Este servicio es el que más ampliamente se ha difundido, ya que, de un formato estático para la publicación de documentos se ha transformado en documentos dinámicos que, junto a los applet y plugins, amplían día a día en contenido de información y la forma de visualizarlo.

Desde el punto de vista de páginas dinámicas, se han sistematizado algunas arquitecturas de diseño como son la cliente servidor y la modelo vista controlador, arquitecturas que se emplean ampliamente en el desarrollo de la aplicaciones.

2.9. Arquitectura Cliente-Servidor

La arquitectura cliente servidor es un modelo de desarrollo de sistemas de información que pretende manejar en forma eficiente los servicios, recursos o procesos que requieren ser compartidos por un cierto número de usuarios. Estos recursos son accedidos en forma remota por la máquina cliente, la cual realiza la petición de los recursos disponibles y el servidor se encarga de:

- Validar el acceso.
- Generar consultas sobre base de datos.
- Entregar informes.
- Gestionar los periféricos compartidos.
- Acceder a otras redes,etc.

Las principales características de una arquitectura cliente servidor son:

- Presentan todos los clientes una interface única.
- El cliente no necesita conocer la lógica del servidor para acceder a los datos.
- Grandes cambios en el servidor no requieren cambios en el cliente.
- El cliente no depende de la ubicación física del servidor.

En la Figura 7 podemos ver el modelo lógico de funcionamiento.

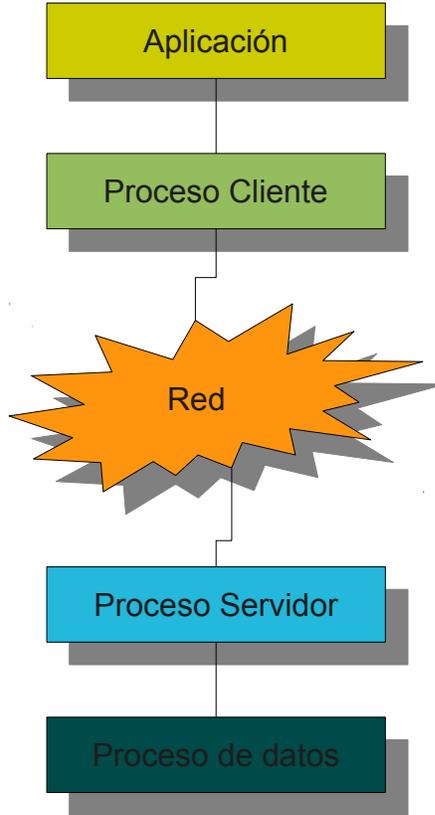


Figura 7: Modelo Cliente Servidor

2.10. Arquitectura Modelo Vista Controlador

El modelo vista-controlador (MVC) es una arquitectura de software que separa los datos de una aplicación, de la interface del usuarios y de la lógica de control. De estos tres componentes el que sufre más frecuentemente cambios es la interfaz de usuario, en especial en los sistemas Web. Fue descrito por primera vez por Trygve Reenskaug en 1979 quien trabajaba para los Laboratorios Xerox. Esta arquitectura permite dotar de mucho dinamismo a los sistemas, ya que, se encuentran modularizados cada uno de sus componentes y deja realizar cambios en sus distintas partes sin que el resto de los módulos se vea afectado. En general, los datos de una aplicación se encuentran almacenados en una Base de Datos la cual puede acceder rápidamente a ellos.

La visualización es provista por los servidores de interface que pueden ser web o aplicaciones visuales como librerías Qt, ncurses, etc.

El modelo lógico representa la forma en que se va a interactuar con los datos o dicho de otra manera, representa el diagrama secuencial con el que se operará la interface de usuario para acceder a ellos, esto se controla por lenguajes incorporados al cliente o por las aplicaciones que se desarrollan en el servidor. Un ejemplo típico de esta arquitectura es la de los sistema web en los cuales la representación se realiza por medio de html, la lógica de control por Javascríp-Ajax-Servlet que acceden a los datos.

En la figura 8 siguiente se puede apreciar el MVC

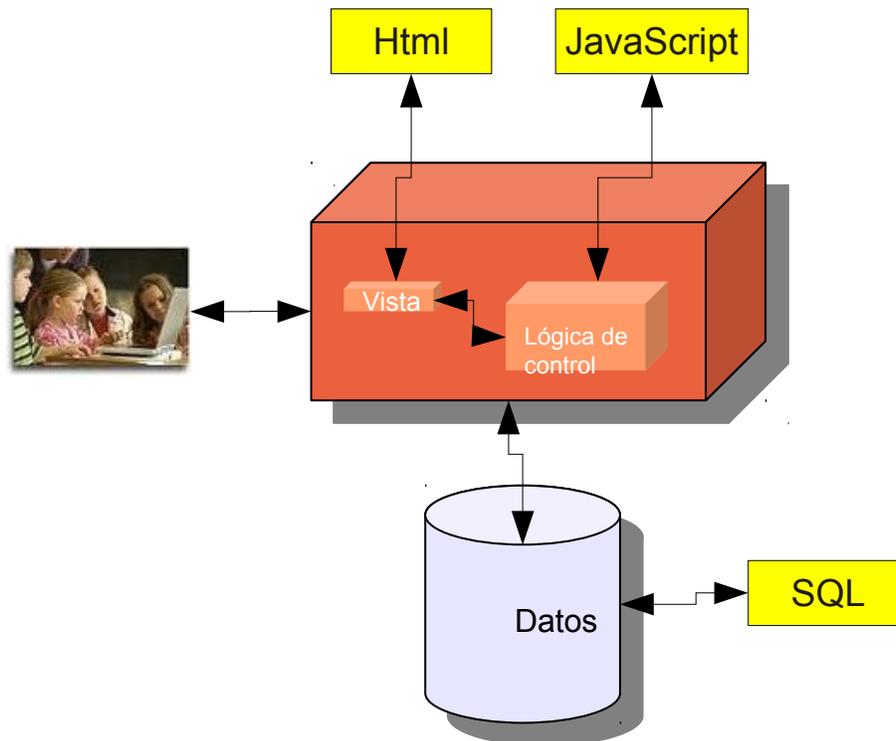


Figura 8: Modelo Vista Controlador.

2.11. Arquitectura Asynchronous JavaScript And XML (AJAX)

La arquitectura Ajax es una mezcla de 4 tecnologías: JavaScript, XML, el objeto XMLHttpRequest y el Modelo de Objetos de Documento (DOM). La característica principal de esta arquitectura es poder actualizar parte de una página con información proveniente desde el servidor

sin tener que actualizar por completo la página. Esto dota a la tecnología web de mayor interactividad y tiene como desventaja el tener que dominar cada una de las tecnologías por separado.

- HTML : Hyper text Marckup Language

Es usado para describir en formato de texto la estructura, el contenido y los objetos de una pagina web. Este lenguaje describe cada campo u objeto del documento por medio de etiquetas que se encuentran entre los símbolos “<>”, como por ejemplo: <head>, que indica el principio de la cabecera del documento y </head>, que indica el final. La documentación del lenguaje es numerosa y se sugiere ver la bibliografía

- JavaScript

Es un lenguaje interpretado cuyo motor se encuentra en el Navegador y que generalmente no se sale de este ámbito. Mejora la interactividad empleando los Modelos de Objeto del Documento (DOM) al cual hace referencia y modifica.

- DOM: Document Object Model

Es un API que proporciona un conjunto de objetos y una interface estándar para acceder a ellos. Define la manera que objetos y elementos de una página se relacionan entre sí y pueden ser referenciados de manera única. Por ejemplo, al definir el objeto <div> y con el identificador “juan”, este puede ser referenciado por medio de DOM de la siguiente manera:

<div id="Juan">...</div>

document.div[0]

document.div["Juan"]

document.div.Juan

donde “juan” es el primer elemento del vector de elementos “<div>”. Este elemento también puede ser referenciado por medio de la función “getElementById”. De la siguiente manera:

document.getElementById("Juan")

1) XML: Extensible Markup Language

Es un lenguaje que permite definir tipos de etiquetas para el principio y el fin de un campo general. Es ampliamente empleado en Internet y pretende ser el estándar para el intercambio de información. Se puede usar en bases de datos, planillas de cálculo, etc. XML permite definir documentos e interpretarlos por cualquier analizador sintáctico debido a que emplea definiciones jerárquicas para un documento cualquiera y contiene una sola raíz de donde proviene todo el resto de las definiciones.

2) Librería JQuery

Esta librería es una implementación de funciones JavaScript para acceder a objetos DOM de manera simplificada, desarrollada inicialmente por John Resig. Nace para facilitar la programación de aplicaciones, ya que, en muy pocas líneas se pueden implementar efectos visuales, drag, drop, autocompletar y animar con mucha facilidad.

Otra de las principales ventajas es que permite despreocuparse por la compatibilidad del código con los distintos navegadores, ya que, es la librería la que se encarga de ello.

La mención de algunas de las funciones de esta librería se hace para que el lector pueda entender los casos tratados con posterioridad donde se hace uso del código que se presenta. No se pretende abordar en extenso todas las propiedades de la librería, puesto que, se sale del alcance de esta memoria.

La librería JQuery posee las siguientes características

- Ahorra muchas líneas de código.
- Hace transparente el soporte de nuestra aplicación para los navegadores principales.
- Provee de un mecanismo para la captura de eventos.
- Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- Integra funcionalidades para trabajar con AJAX.

Para emplear esta librería debemos poner dentro de la página el siguiente código.

```
<script type="text/JavaScript" src="jquery.js"></script>
```

Esta librería de funciones posee una amplia documentación la cual se encuentra difundida en Internet.

a) Funciones y métodos básicos de JQuery.

La función básica de JQuery y, a la vez, una de las más útiles, es:

`$()`

Corresponde a una implementación de

`document.getElementById(), getElementByAttr(), etc.`

Para referenciar uno o más elementos basta con incluirlo como cadena de texto en los paréntesis, incorporando la sintaxis que lo diferencia. Por ejemplo: si se quiere referenciar un elemento por su "id", se antepone el carácter "#". Supóngase que se tiene definido un campo por una id= "primero", entonces la forma de acceder a él y sus clases es:

`$('#primero')`

También se pueden referenciar de acuerdo al tipo de elemento, ya sean, todos los "<td>" de una tabla, un enlace "<a>", etc. Si se quisiera referenciar todos los enlaces de una página la función será llamada de esta manera.

`$("a")`

Una cuestión importante que se realiza con esta función es la captura de eventos y la posterior ejecución de funciones JavaScript.

Para capturar el evento "onclick" veremos el siguiente ejemplo.

`$('#primero').click(alert("Salir"))`

Se llama a la función "`alert()`" que entrega en una página emergente con el mensaje "Salir".

Una característica importante de esta librería es que de una manera muy intuitiva se modifican los estilos de los elementos. Para cambiar el comportamiento de un elemento "<div>" desde un estado visible a no visible se emplean las funciones "`hidden()`" y "`show()`".

```
<div id= "primero"></div>
```

Ahora para cambiar el estado de visible a oculto empleamos la función *“hidden()”*.

```
$('#primero').hidden()
```

Para recuperar la visibilidad del elemento *“<div>”* simplemente llamamos a la función *“show()”*.

```
$('#primero').show().
```

Todas las funciones relacionadas con los efectos visuales permiten indicar dos parámetros opcionales: el primero es la duración del efecto y el segundo es la función que se ejecuta al final del efecto.

Otros efectos visuales a los que se pueden acceder, entre otros son:
fadeIn(), fadeOut(), fadeTo(): para visualizar y ocultar en forma progresiva, etc.

3) Funciones Ajax

La función que realiza Ajax se llama *\$.ajax()* y es la de más alto nivel que se encuentra en la librería, sin embargo, hay otras implementaciones que se construyen a partir de esta, las cuales cumplen funciones más específicas.

La sintaxis de la función es:

```
$.ajax(opciones);
```

con opciones iguales a

```
{  
  
    url: '/ruta/de/pagina.jsp',  
    type: 'POST',  
    async: true,
```

```
data: 'parametro1=valor1&parametro2=valor2',
success: Procesa Respuesta,
error: Para recuperar el error
}
```

Otras funciones implementadas a partir *\$.ajax()* son:

\$.get(opciones);

con opciones iguales a

```
{
  url: '/ruta/de/pagina.jsp',
  { dato:valor},
  function respuesta(datos){ alert(datos)}
}
```

\$.post(opciones);

con opciones igual a

```
{
  url: '/ruta/de/pagina.jsp',
  { dato:valor},
  function respuesta(datos){ alert(datos)}}
```

4) Datos nativos JavaScript (JSON)

JSON es un acrónimo de JavaScript Objeto Notación. La razón principal por la cual se implementa esta notación para el intercambio de datos y no se emplea XML es la sencillez con que pueden ser interpretados los datos provenientes desde el servidor, ya que, basta con usar la

función `eval()` de JavaScript para que los datos puedan ser usados en forma nativa en el lenguaje, ya sea que provienen de un string, un arreglo o una estructura.

La función para el intercambio de información en el diseño del sistema será la función Ajax llamada:

- `$.getJSON(opciones)`

con opciones iguales a

```
{  
  url: '/ruta/de/pagina.jsp',  
  { dato:valor},  
  function respuesta(datos){ alert(datos)}  
}
```

La razón principal por la que se empleará la función `$.getJSON()` es que las lecciones, datos de usuario y el resto de la información se pueden emplear sin necesidad de procesamiento en el lado del cliente, siempre que sean pre formateadas en alguna estructura de datos nativa de JavaScript.

2.12. Servidor Tomcat.

El servidor Tomcat es un servidor web con soporte de Servlets y páginas jsp, es mantenido por la Apache Software Foundation. A menudo, este servidor es presentado en combinación con el servidor Web apache pero puede funcionar en forma autónoma. Todos los usuarios pueden acceder a su código fuente y modificarlo si lo desean en concordancia con “ Apache Software Licencia”.

La principal características de Tomcat es que permite ejecutar código Java en las páginas que se encuentren en el servidor. Este código compilado a bytecode es denominado Servlets y se pueden emplear todas las librerías provistas por Java y, por lo tanto, posee una infinidad de aplicaciones. Los servlets se pueden clasificar en:

1) Contenedores de Servlets Stand-alone (Independientes)

Se define este tipo de servidor cuando está basado íntegramente en código Java. Este es el modo por defecto usado por Tomcat. Sin embargo, la mayoría de los servidores, no están basados en Java lo que lleva a los dos siguientes tipos de contenedores.

2) Contenedores de Servlets dentro-de-Proceso

El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java. El plugin del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debiera ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento, pero está limitado en escalabilidad.

3) Contenedores de Servlets fuera-de-proceso

El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debiera ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc).

Las páginas JSP son escritas en html en el cual se incluye lenguaje java por medio de la directiva “<% >”, lo cual dota de mucho dinamismo a las páginas html permitiendo incluir complejas soluciones matemáticas o resultados del acceso a base de datos.

El servidor tomcat posee una estructura de directorios que se detalla a continuación:

- Bin. Donde se encuentran una serie de script de arranque, cierre y ejecutables
- Common. Se encuentran una serie de clases o librerías que pueden ser usadas por los servlets o JSP.
- Conf. Lugar donde se almacenan los archivos de configuración y están escritos en xml

- Logs. Archivos de sucesos del servidor y sus aplicaciones.
- Server. Clases utilizadas sólo por servidor.
- Shared. Clases compartidas sólo por las aplicaciones web.
- Webapps. Directorio donde se encuentran las aplicaciones web.
- Work. Es donde se almacenan temporalmente archivos y directorios.

2.13. Servidor PostgreSQL

Postgresql es un sistema de gestión de base de datos relacional orientada a objetos y libre, (publicado bajo la licencia BSD). El desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

1) A continuación mostramos una lista con las principales características.

- Alta concurrencia.
- Alta variedad de tipos de datos nativos.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte transacciones distribuidas.
- Replicación.
- Alta disponibilidad.
- Funciones en una gran variedad de lenguajes.
- Lenguaje nativo (Plsql).

2) Archivos de configuración

- postgresql.conf. Es el archivo donde se configura las conexiones remotas y el comportamiento global del servidor.

- pg_hba.conf. Es el archivo donde se configura la lista de acceso al servidor, ya sea, por usuario, red ,etc.

3) Algunas utilidades

- PgAdminIII. Utilidad gráfica para la configuraciones y mantenimiento de la BD.
- PgDump. Utilidad encargada de hacer respaldos de la BD.
- Psql. Utilidad cliente por línea de comandos.
- PgBench. Utilidad para hacer pruebas de stress.

2.14. Servidor ASR (Automatic Speech Recognition) .

El servidor ASR es el encargado de realizar el procesamiento de la voz. Es un servicio que consiste en convertir una señal acústica en clases conocidas y previamente definidas. A grandes rasgos se puede dividir en 4 áreas, capturar los datos, extraer las características, comparar las clases obtenidas con las previamente definidas y entregar una evaluación a la respuesta por voz.

Para la captura de datos se emplea un micrófono y un convertidor análogo-digital, este puede ser una tarjeta de sonido, un teléfono celular, etc.

La extracción de características se realiza por medio de una serie de procedimientos matemáticos. En procesamiento de voz se acostumbra a usar los coeficientes cepstrales. Estos coeficientes son los más usados en técnicas de procesamiento de voz. Los MFCC son Mel Frequency Cepstral Coefficients (Coeficientes Cepstrales en las Frecuencia de MEL), son coeficientes para la representación del habla basados en la percepción auditiva humana. Se derivan de la Transformada de Fourier (TF) o de la Transformada de Coseno Discreta (TCD), su principal característica es que las bandas de frecuencia se encuentran espaciadas en forma logarítmica. El procedimiento para obtener los MFCC es el siguiente.

1. Se calcula la TF a una porción de la señal.
2. Usando una función triangular se mapea la energía del espectro en escala MEL.
3. Se calcula el logaritmo de la energía de cada frecuencia MEL.
4. Se determina la TCD de cada de los datos anteriores como si fueran una señal.

5. Los MFCC son las amplitudes del espectro resultante.

El vector obtenido a partir de los MFCC constituyen las características acústicas de la voz, el cual se compara con los vectores o clases previamente definidos en el sistema. Para realizar el procedimiento de comparación se emplean técnicas estadísticas. El teorema de Bayes permite modelar el reconocimiento de voz como un problema de maximización a posteriori, empleando la palabra y cada una de las secuencias de vectores obtenidas, las cuales son modeladas como una secuencia de estados conectados que poseen probabilidades de transición. Cada estado posee distribuciones de probabilidad de la cual se obtiene su verosimilitud. Este método emplea modelos ocultos de Markov de primer orden (HMM, “*Hidden Markov Models*”) esto significa que el estado actual depende del anterior. Para realizar la búsqueda de las distintas clases que componen la palabra se hace uso del Algoritmo de *Viterbi*. El método consiste en realizar optimizaciones a nivel local, con lo que se logra reducir el campo de búsqueda y genera un algoritmo viable desde el punto de vista computacional. Posteriormente los resultados son entregados de acuerdo a la mejor secuencia conseguida del reconocimiento realizado. Es necesario señalar que el procedimiento empleado para el procesamiento de voz posee un 98% de resultados asertivos.

CAPITULO 3

3. IMPLEMENTACIÓN.

En este capítulo se verán las características del sistema; la forma en que fue implementado, las estructuras necesarias para su funcionamiento; la definición de los distintos tipos de usuarios, como se relacionan y el hardware donde se instaló.

3.1. Características generales del sistema

Con el diseño y el desarrollo de esta plataforma se desarrolló un completo sistema de seguimiento, registro de notas y administración de usuarios. Lo que unido a la capacidad de alma-

cenamiento remoto de datos de navegación y voz, permitirán mejorar permanentemente esta herramienta de enseñanza dado que a partir de esta información se puede determinar el comportamiento y los resultados de los usuarios con el sistema.

La siguiente lista enumera los principales atributos de la plataforma de enseñanza que se caracteriza por ser.

- Liviano desde el punto de vista de la ejecución en el cliente.
- Posee la capacidad de ejecutarse en los navegadores mas comunes.
- No usa plugins adicionales de los que vienen incluidos en el navegador.
- Emplea lenguajes ampliamente difundidos.
- Es multiusuario, multinivel y jerarquizado.
- Almacena remotamente todas las actividades que realicen los alumnos.
- Puede administrar usuarios desde la aplicación.
- Es paralelizable a nivel de servicios.

Estas características se traducen en usar un Applet Java para la transmisión de voz, JavaScript para el manejo de los eventos y HTML para la transmisión de datos de manera que el sistema esté plenamente funcional en una configuración estándar. El modelo vista-controlador más Ajax permite que sea usado en máquinas con pocos recursos en hardware y software, debido a que la cantidad de memoria y procesamiento necesario se reduce al mínimo por lección; sin embargo, esto redundante en que será necesario recibir la información desde el servidor para cada lección que se acceda. El almacenamiento de las lecciones y la información de usuarios quedan registradas en la Base de datos Postgresql a la cual se accede por medio de consultas en lenguaje SQL. Es importante mencionar que al ser este un sistema de enseñanza de un segundo idioma, que emplea herramientas de procesamiento de señales, en particular de voz, necesita de los siguientes requisitos de hardware que son :

- Un micrófono.
- Parlantes o audífonos.
- Tarjeta de sonido de 16 bits.
- Red Lan o Internet.

En la Figura 9 vemos un diagrama general del cliente servidor.

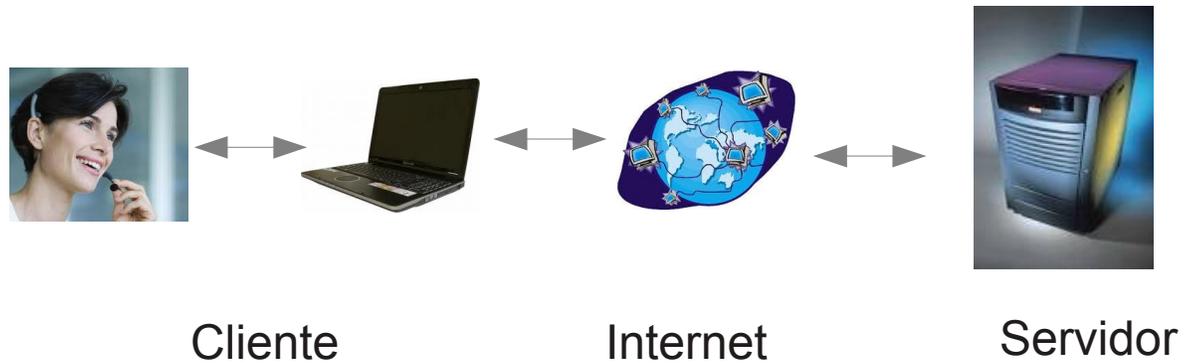


Figura 9: Cliente servidor.

3.2. Componentes del Sistema

1) El Servidor de la plataforma.

El Servidor es la parte fundamental del sistema y consta de 4 servicios que pueden ejecutarse en una o varias computadoras, cada una de sus partes se comunica por medio de Socket, los cuales pueden ser direccionados por IP. Los servicios que componen el servidor son: HTTP, Base de datos, motor de evaluación de pronunciación y respuesta por voz, en adelante (ASR), junto al servidor de Dictado son los que en conjunto constituyen la plataforma de enseñanza.

a) Servicio Http.

Este servicio establece comunicación por medio del puerto 80, que es el puerto por defecto y es por donde se realiza todo el tráfico de datos e información con el cliente. Este servicio realiza conexiones del tipo local para comunicarse con ASR, Dictado y base de datos. Estas comunicaciones se realizan usando Socket a nivel local, los cuales son accedidos por medio de páginas Servlet que son ejecutadas desde el cliente en el servidor. Estas páginas son implementadas en java y se encuentran compiladas en un archivo jar. La información necesaria para estas páginas se recibe por el Método POST desde el cliente y es procesada para ser enviadas al servicio respectivo.

El servicio http esta implementado por medio del servidor Tomcat, que provee de la transferencia de información por medio del protocolo http entre la máquina cliente y la máquina servi-

dor. Junto a esto, Tomcat provee de unas páginas dinámicas, JSP “Java Server Pages” y servlet, las primeras permiten incluir código java al interior de las páginas html, las segundas son implementadas en lenguaje Java y puede contener respuestas en html dentro del código. Este tipo de paginas son empleadas para realizar una comunicación al puerto 5432, para el envío y recepción de señales de voz, peticiones de dictado, direccionado de servicios y las respuestas respectivas.

b) Base de datos.

La base de datos se comunica por el puerto 5432 el que se puede modificar en el archivo de configuración. Es una base de datos relacional, orientada a objetos que implementa el estándar de consultas SQL. Es donde quedan almacenadas las lecciones, los datos de acceso y la información de navegación.

c) Servicio ASR.

Se encuentra implementado para comunicarse por socket con un protocolo propietario del LPTV; es el encargado de procesamiento de voz y sus respectivos resultados.

d) Servicio de Dictado.

Es el encargado de procesar la información provenientes de la página jsp respectiva, entregando evaluaciones de las lecciones. Esta comunicación es la que se realiza por medio del servlet.

El esquema de funcionamiento del servidor se puede ver en la figura N°10.

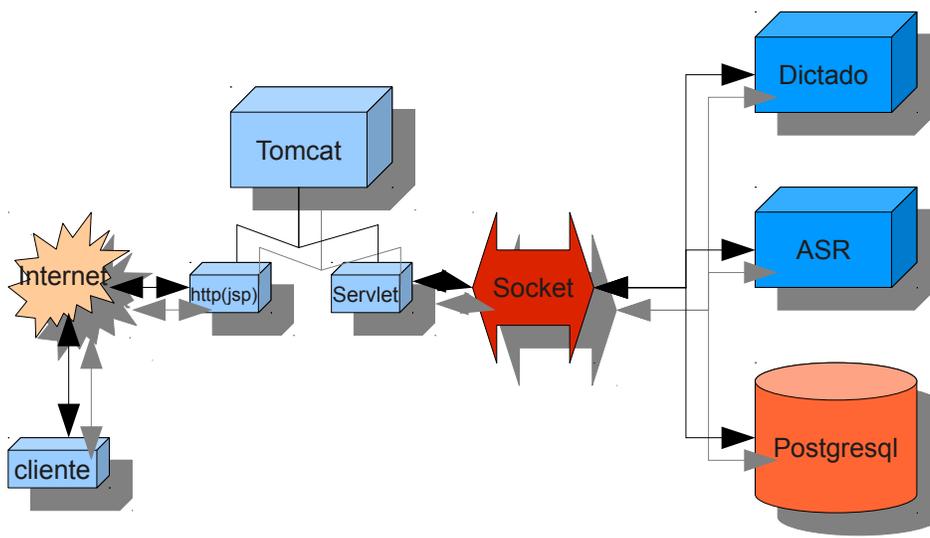


Figura 10:: Esquema del Servidor.

2) Cliente.

El cliente un computador que posee la capacidad de comunicarse por medio del protocolo http y puede interpretar el lenguaje Html, JavaScript y applets en java. Estas características se encuentran en la mayoría de los navegadores actuales, sin embargo, el navegador Explorer de microsoft realiza implementaciones propietarias del estándar html y JavaScript, por lo que, al realizar un servicio web es necesario tener cuidado con el navegador. Para resolver la compatibilidad con los distintos navegadores se empleo la librería JavaScript JQuery la implementa las funciones con compatibilidad para los distintos navegadores. En la Figura N° 11 se puede observar un diagrama funcional del cliente.

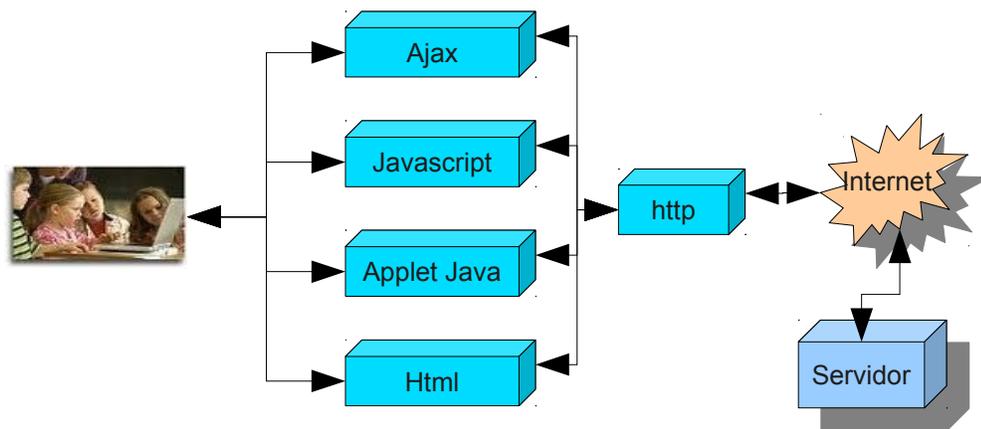


Figura 11: Descripción Funcional del cliente Web.

3) Base de Datos

El nombre de la Base de datos que se implemento en esta memoria se llama `bd_eloud`, es única en el sistema y se accesa por medio de un usuario y un password. La BD posee tablas, procedimientos de almacenado, índices y estadísticas, las cuales permiten un rápido acceso a los datos que se encuentran guardados. La BD optimiza el acceso a la información no sólo por medio de índices sino también por análisis estadísticos internos a ella. Los procedimientos de almacenado son funciones que pueden ser desarrolladas por el administrador de la BD en adelante DBA y tienen por objeto realizar operaciones complejas. Son programadas en un lenguaje propio deno-

minado PL/SQL y se almacenan en un espacio dentro de la BD. También pueden ser ejecutadas como cualquier función.

En el sistema hay consultas simples, como la consulta para una lección que se implementan por SQL y otras complejas, como por ejemplo, el avance de un alumno en las lecciones. Las consultas complejas se encuentran implementadas en PL/SQL, ya que, es este lenguaje el que permite realizar funciones para la consulta de múltiples tablas y campos.

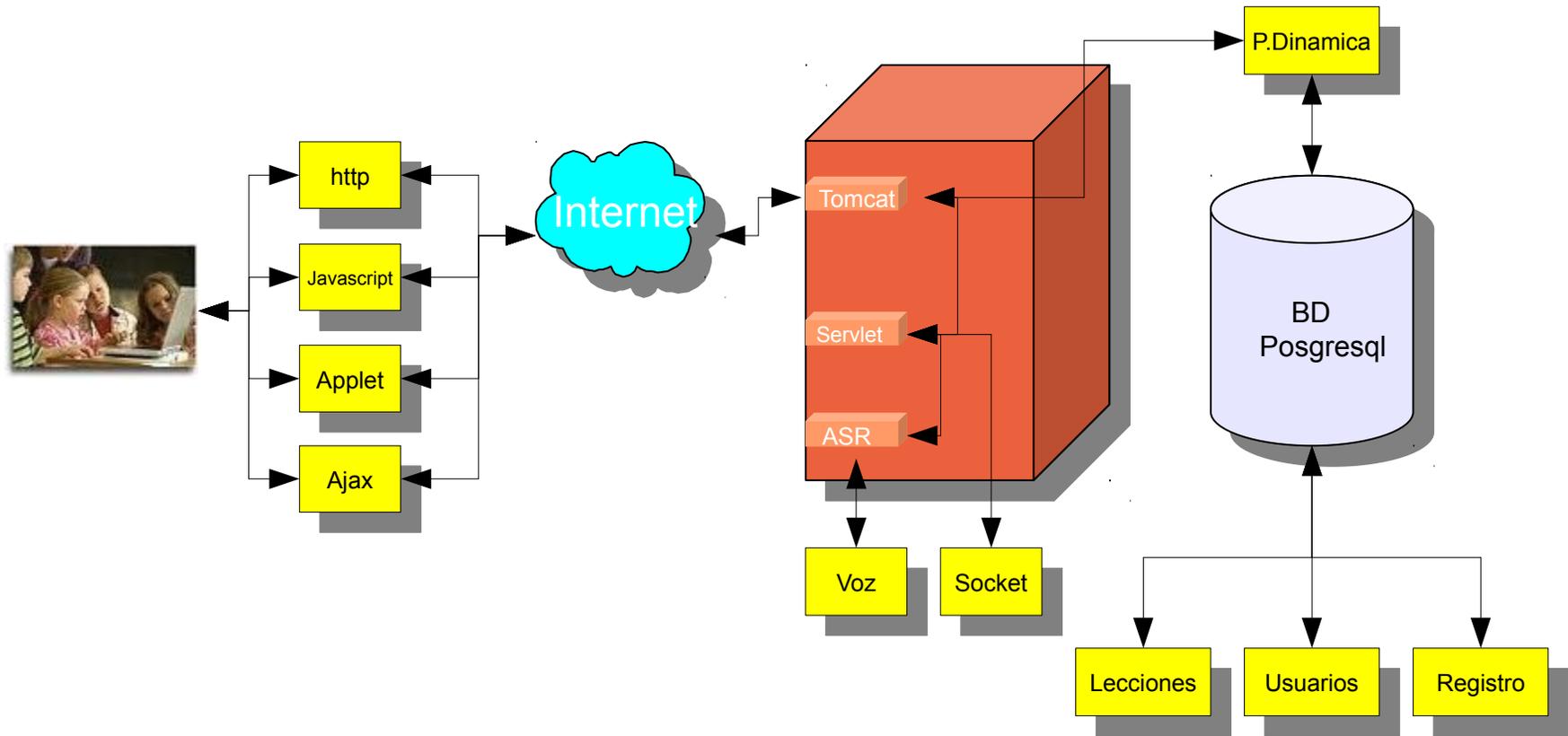


Figura 12: Descripción General de Sistema.

4) Estructura de la base de datos

La base de datos posee 5 tablas en las cuales se almacenan los datos de usuarios, lecciones y las estadísticas, estas se relacionan por medio de un identificador de usuario, llamado “id”, que es un número entero único en el sistema. Para asegurar la unicidad de este número se definió como incremental, esto quiere decir que cada vez que se ingrese un nuevo usuario el número se incrementara automáticamente. En la Figura 13 se puede apreciar la relación de las distintas tablas y campos.

La tablas son:

- TABLA USUARIOS. En esta tabla se almacenan todos los datos de usuarios como son, el nombre, login, password, colegio, correo, etc.
- TABLA ESCUELA. En ella se relaciona los usuarios docente, UTP con los cursos asignados y las escuelas respectivas.
- TABLA CONTENIDO. En esas tablas se almacenan las distintas lecciones, las cuales se encuentran ordenadas por curso, lección, actividad. En el campo valor, se encuentra codificada una estructura de JavaScript, que contiene los ejercicios de cada actividad esto ultimo, debido a que se puede ingresar esta estructura directamente en la memoria del cliente y acceder a los ejercicios por medio de un índice del arreglo.
- TABLA NOTAS. En esta tabla se almacenan las notas de los ejercicios realizados quedando guardadas las evaluaciones junto a la hora de realización. Esto significa que puede haber más de una nota por actividad, dependerá de la cantidad de veces que se realice el ejercicio en cuestión.
- TABLA ESTADISTICAS. En esta tabla se almacenan todas las interacciones realizadas por el usuario. Esto significa registrar la cantidad de botones que se presionan para llegar a enviar una evaluación y la ubicación del archivo con la voz asociado al ejercicio correspondiente. El nombre asignado al archivo es un string que contiene el “timestamp” y la dirección física donde se encuentra el archivo. El timestamp es un numero que denota la hora, la fecha en que ocurrió un evento.

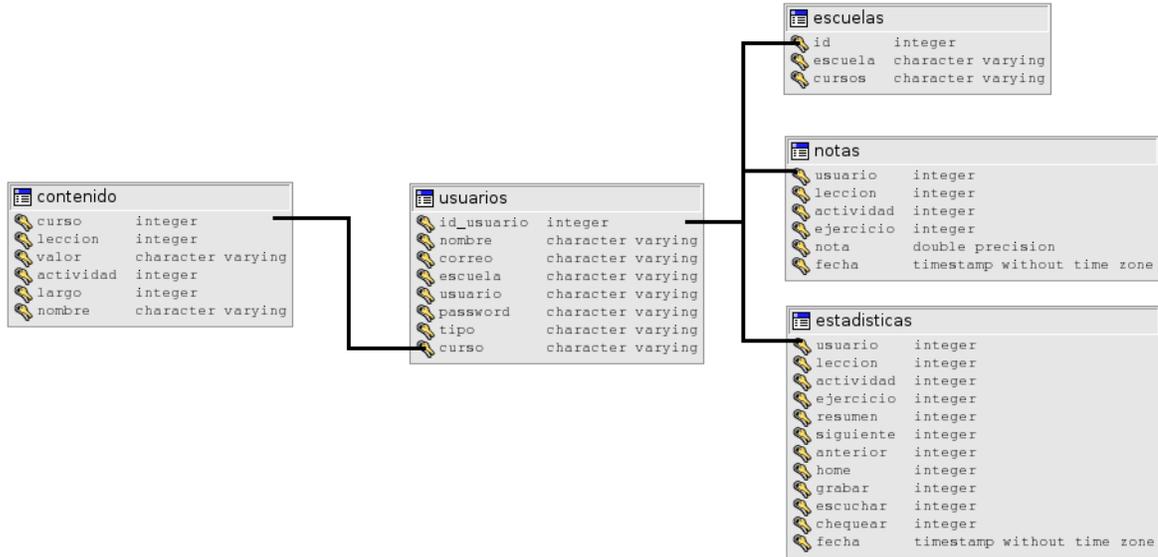


Figura 13:Esquema de la Base de datos.

3.3. Aplicación WEB

El sistema Web desarrollado consta de diversos módulos relacionados por medio de la base de datos los cuales se adecuan al rol que cumple cada usuario en el sistema, por lo que, se tendrá tantos módulos como tipos de usuarios. Cada modulo consiste en una agrupación de funciones que se relacionan con un tipo de usuario en general. Los usuarios son admin, utp, docente y alumno.

1) Definición de usuarios de la plataforma.

- El tipo de usuario admin. En el sistema pueden haber un sin número de escuelas, por lo que, este usuario es el encargado de la definición de éstas y de la asignación de uno o más encargados por escuela (UTP).
- El usuario UTP. Este es el encargado de ingresar las listas de los cursos que acceden al sistema y la asignación de los docentes respectivos pudiendo ingresar, borrar y asignar a los alumnos como a los docentes.

- El usuario Docente. Administra avance académico de los alumnos.
- El usuario Alumno. Es a quien van dirigidas las lecciones, envía su voz por medio del micrófono, digita la información requerida por cada lección, etc.

2) Implementación en la Base de datos.

Para poder diferenciar las distintas funciones que desarrolla el sistema se definió un campo “tipo de usuario” en la BD, con esto cada tipo de usuario accede a las funciones definidas para su nivel de acceso, por ejemplo: que los alumnos pueden acceder a las lecciones, los docentes a las notas y los supervisores a las listas de alumnos y docentes. A continuación se definirá cada tipo de usuario, sus funciones y el nivel de acceso que poseen en la plataforma.

Todos los usuarios acceden por la misma página al sistema, por lo que, una vez autenticado el usuario, se define su tipo y se ejecuta el módulo correspondiente. La página de acceso al sistema es única y se puede apreciar en la figura 14.



Figura 14:Pagina de Acceso.

3) Módulo del estudiante.

El módulo principal y el más importante es el del estudiante, ya que, en este módulo se desarrolla toda la actividad de aprendizaje. Están implementadas las lecciones, ejercicios, avance, el resumen de notas, etc. El módulo del estudiante se puede ver en la figura 15 b), c), d) y e).



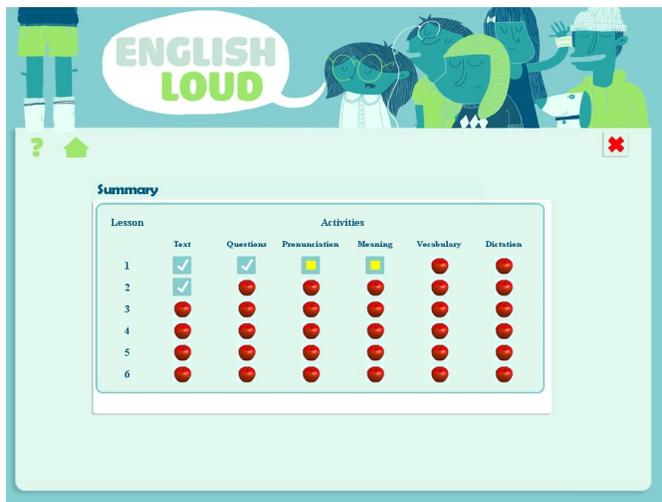
a) Ingreso al Eloud.



b) Grabar el texto.



c) Escribir dictado.



d) Resumen avance.



e) Resumen de Notas.

Figura 15: Modulo del estudiante.

Una vez ingresado el alumno, la primera página que ve es la de resumen del avance figura 15 d) y es donde se ve en qué lección se encuentra trabajando. Cada lección posee 3 estados que son: no realizada, en proceso y terminada. Éstas se pueden ver en la tabla 2.

<i>Estado de la lección</i>	<i>Figura</i>
<i>No realizada.</i>	
<i>En proceso pero no terminada.</i>	
<i>Terminada.</i>	

Tabla 2: Estado de avance.

Para acceder a los ejercicios de la lección se realiza un click sobre las imágenes. Esto dirige al primer ejercicio de la actividad.

Las actividades son:

- Text. Es la primera parte de la lección y corresponde a una lectura introductoria de la lección, donde se encuentran las palabras y el contexto de la lección.
- Questions. Son preguntas relacionadas con el texto introductorio las cuales se responden por voz, son evaluadas la pronunciación y la respuesta.
- Pronunciation. Corresponden a frases que se deben pronunciar en el micrófono las cuales son valoradas en concordancia con los aciertos de la pronunciación.
- Meaning. De una lista de palabras que son escuchadas por el alumno se debe seleccionar una y digitar la letra correspondiente a la respuesta.
- Vocabulary. Se debe escuchar la palabra las veces que sea necesario para posteriormente pronunciar frente al micrófono correctamente la palabra de la lección.

- Dictation. Sin leer la palabra se repite el vocablo escuchado, esto se puede realizar varias veces por problema.

4) El módulo del docente.

Este módulo permite ver:.

- Los cursos asignados al docente.
- La lista de los alumnos.
- El resumen de notas del alumno seleccionado.

En la figura N° 16 se aprecian las distintas partes del módulo y los sub-menus asociados, junto con ver el resumen de notas de un alumno.

The image shows two screenshots of a software interface. The top screenshot displays a table for course assignment and a list of students. The bottom screenshot shows a detailed view of a student's grade summary across six lessons.

Escuela	Curso	Curso	Nombre	Username	E-mail	Evaluation
Escuela1	5A	5A	alumno1	alumno1	alumno1@d.cl	View
		5A	alumno2	alumno2	alumno2@e.cl	View

Lesson	Text	Questions	Pronunciation	Meaning	Vocabulary	Dictation
1	-	-	-	-	-	-
2	-	-	-	-	-	-
3	-	-	-	-	-	-
4	-	-	-	-	-	-
5	-	-	-	-	-	-
6	-	-	-	-	-	-

Figura 16: Módulo de administración docente, en la parte superior se puede ver la asignación de cursos y en la inferior el resumen de notas del alumno.

5) El módulo del administrador de escuela "utp"

La administración de los establecimientos educacionales es llevada a cabo en forma interna, razón por la cual se define tipo de usuario "utp", de manera se pueden resolver rápidamente la falta de docentes, cambio de alumnos, etc. Para ello, se ha creado un menú orientado al usuario tipo "UTP", en el que se pueden agregar o borrar cursos, administrar la listas de alumnos, asignar docentes, etc. Para un mejor entendimiento del menú se ha incorporado éste y los sub-menús en la figura 17. En ella se puede apreciar:

- a) Lista de Docentes.
- b) Cursos asignados al profesor.
- c) Lista de alumnos por curso.
- d) Icono para agregar alumnos al curso.

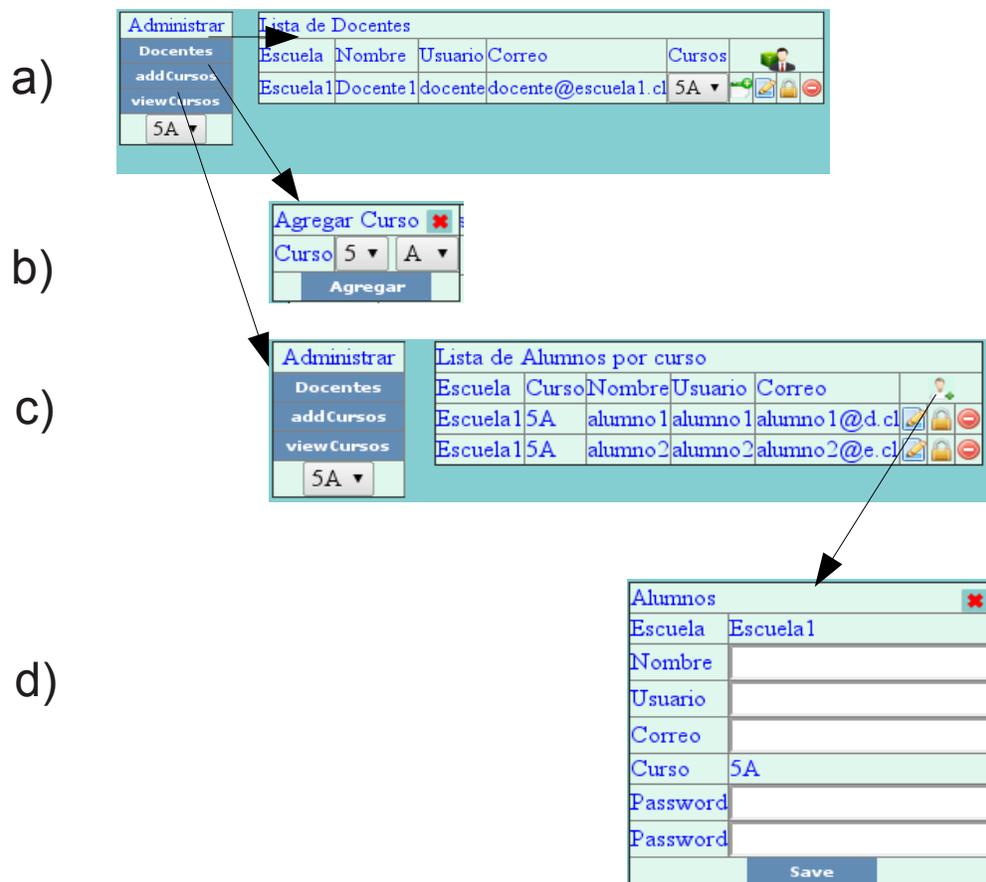


Figura 17: Menú administrador de escuela "utp"; a) lista de Docentes; b) cursos asignados al profesor; c) lista de alumnos por curso; d) icono para agregar alumnos.

6) Administración general del sistema

La administración general del sistema se realiza por los usuarios que poseen el tipo “admin”. Este tipo de usuario posee el menú de ingreso de escuelas y usuarios tipo “utp”. Para ingresar un nuevo usuario “utp” al sistema es necesario ingresar primero la escuela a la que pertenece. En el menú aparecerá como opción desplegada la elección de la escuela para los usuarios.

Una vez ingresada la escuela se procede a ingresar el usuario, el menú de “admin” lo podemos ver en la Figura 18. Un algoritmo de ingreso de usuarios al sistema se encuentra en la Figura 19.

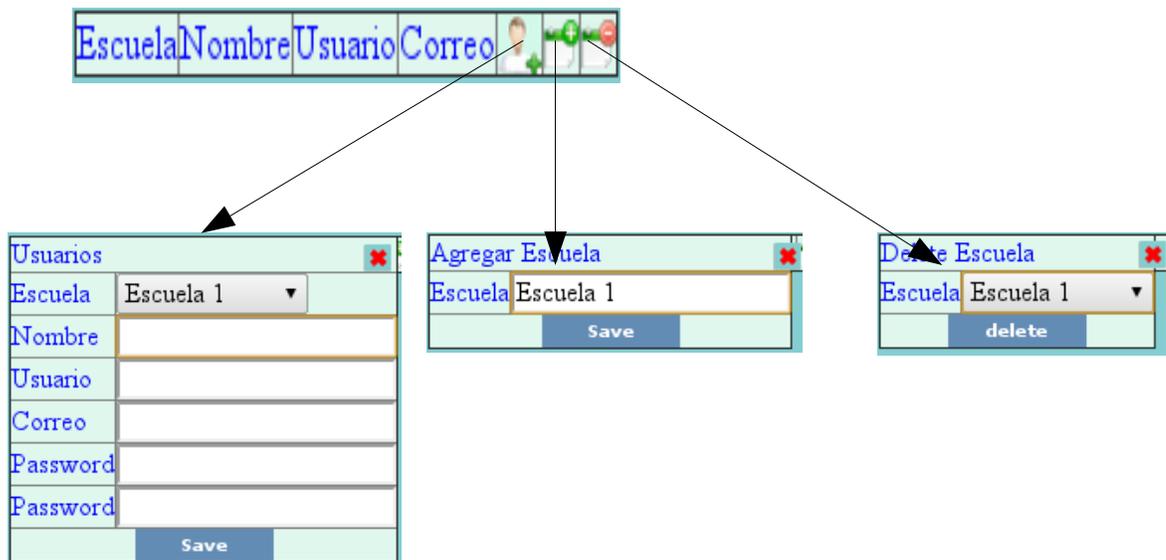


Figura 18: Menú de administración general y algunas opciones.

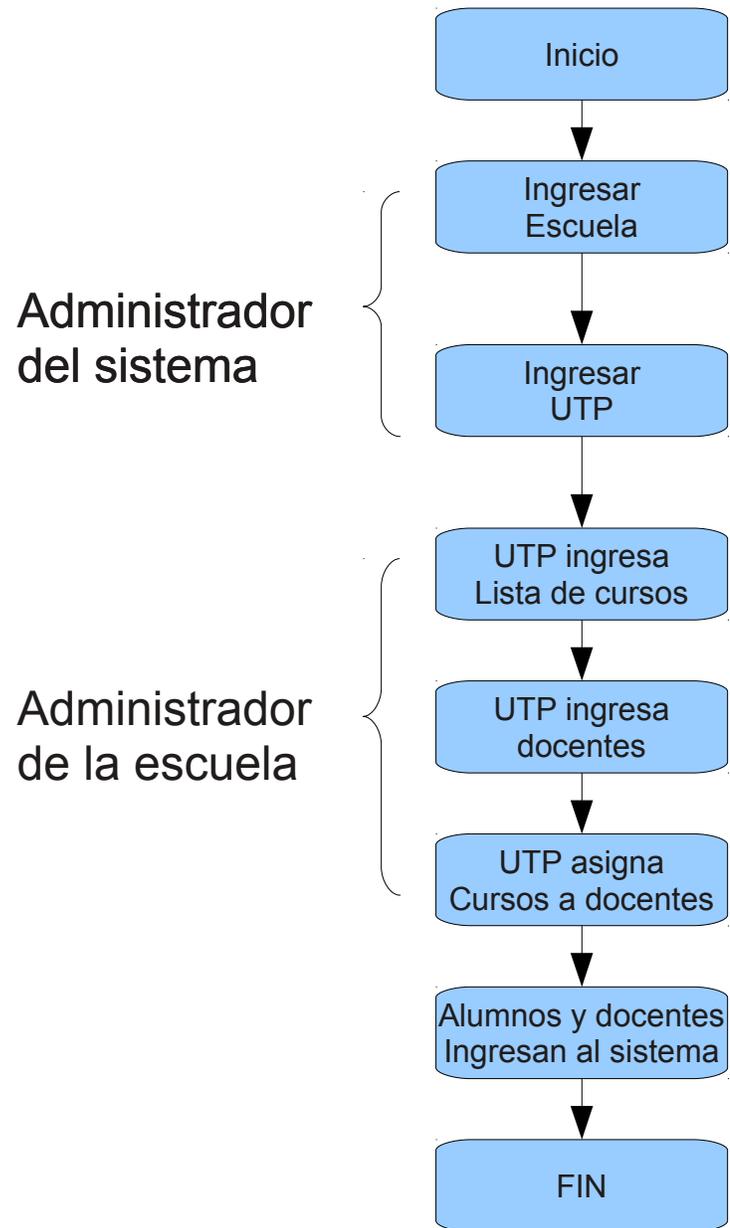


Figura 19: Algoritmo de ingreso de usuarios.

Como se ha visto, hasta el momento hay 2 tipos de usuarios administradores: los “admin” que cumplen la labor de administración general del sistema y los usuarios “utp”, los cuales administran el sistema por escuela. Esta definición se realizó con la intención de adecuar el sistema al trabajo que se realiza en los establecimientos educacionales.

3.4. Lenguajes y archivos de la aplicación

Como ya se ha mencionado, la plataforma de enseñanza Eloud es un programa que permite la interactividad de los usuarios con el sistema, para ello, se creó un software utilizando como base html-JavaScript-ajax-jsp-servlet junto a un applet java . Este software posee diversas funciones las cuales se encuentran divididas según el lenguaje empleado.

1) Lenguaje Html.

Este lenguaje se emplea como patrón de diseño. Es el encargado de dimensionar y posicionar los distintos elementos gráficos, por lo que, se encontrará como una página estática la cual es modificada en forma dinámica en concordancia con los requerimientos del usuario.

2) Lenguaje JavaScript.

Son las funciones que realizan la interactividad de botones, imágenes y la actualización del código html. Junto con proveer de Ajax embebido, este lenguaje se encuentra implementado en los archivos llamados jquery.js, jpage.js, admPage.js, junto a otros que entregan funciones particulares.

Jquery.js. Este archivo corresponde a una librería de funciones.

Jpage.js. Aquí se encuentra todo el código de interacción lógica del sistema, esto es, el manejo de la interactividad, el envío y recepción información, la forma en que se actualizan y visualizan las lecciones, etc. Este archivo es el que emplea preferentemente el usuario alumno.

AdmPage.js. Es donde se encuentran las funciones de actualización de las páginas de administración, utp y docente.

3) Java Server page.

Estos archivos son los poseen código ejecutable por el servidor tomcat y se encuentran escritos en html y java.

4) Servlet

Es un programa escrito en Java que gestiona la búsqueda y envío de información de voz, direccionando a los distintos servicios provenientes del cliente. Por ejemplo: Si desde el cliente se envía una petición de análisis de voz es el servlet el que se encarga de interpretar las peticiones y enlazar el puerto correspondiente a la lección respectiva.

5) Applet Java.

El Applet Java es el que toma las muestras de voz y se envían encapsuladas al Servlet donde se procesa y se le entrega el socket de destino para posteriormente recibir los resultados del servicio ASR o dictado.

3.5. Implementación, funcionamiento y estructuras.

El curso de inglés implementado en la plataforma English Loud se divide en 5 lecciones, 6 actividades y entre 8 a 10 ejercicios por actividad. Este curso es multinivel y puede ser ampliado a cuantos niveles se desee, ya que, el sistema considera un campo denominado curso. Si bien es cierto, está pensado entre 5º y 8º año de enseñanza básica, esto es sólo referencial.

Por otra parte, las actividades pueden tener cuantos ejercicios se quiera, sólo basta con que esté en la base de datos para que los ejercicios sean mostrados por el sistema.

Como se ha dicho con anterioridad, las lecciones se encuentran almacenadas en la BD y para acceder a ella se realiza una petición por medio de un evento JavaScript el cual se conecta a una página jsp por medio de Ajax, lo que permite que el sistema se actualice rápidamente sin mostrar mayores cambios al cargar los datos.

1) Funcionamiento de la plataforma.

El funcionamiento se debe a la interacción de distintos lenguajes que se traspasan información donde cada uno cumple una función específica, ya sea interactividad con el usuario o conexión con la BD. El diagrama de Flujo de la Figura 20 se puede apreciar apreciar el diagrama de flujo de la información en la Plataforma.

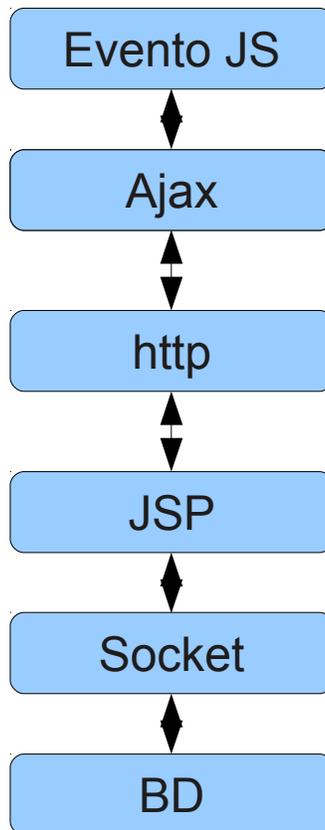


Figura 20: Diagrama de Flujo de información.

Para entender el diagrama de flujo explicaremos cada uno de los campos.

- El evento JS corresponde a alguna actividad realizada por el usuario, como por ejemplo la pulsación de una imagen o botón.
- Ajax nos provee del formato de transmisión de la información y de las funciones necesarias para que esta sea asíncrona, esto significa que los datos de actualización se realizan

sin necesidad de cargar nuevamente la página y en segundo plano, de manera que el usuario no nota este proceso.

- Http nos proporciona los métodos por los cuales el servidor recibe la información, estos métodos son POST o GET según sea el caso.
- JSP son páginas dinámicas que se cargan en concordancia con la petición de las lecciones, esto quiere decir que cada vez que el usuario quiera una nueva lección esta petición será atendida por una de estas páginas en el servidor.
- Socket son empleados para realizar comunicación entre los distintos servicios que se encuentren en el servidor, como por ejemplo ASR o la BD.
- BD corresponde a la Base de datos de la plataforma y es donde se encuentran las lecciones y los datos de actividad del usuario.

2) Código JavaScript

JavaScript es un lenguaje interpretado que se ejecuta en general en el navegador web, permite mejorar la interfaz del usuario incorporando dinamismo en los distintos componentes en concordancia con la actividad que realiza el usuario. Para ello se requiere identificar los componentes y definir los distintos eventos.

a) Identificación de los componentes.

Los distintos componentes de una página Html son: imágenes, botones, tablas, texto, párrafos etc. Se identifican por medio de id o class, palabras reservadas del lenguaje HTML. Las class se emplean habitualmente en las definiciones de estilo, las cuales se pueden modificar por medio del lenguaje.

b) Definición de evento

Un evento es una acción que realiza el usuario ante el programa navegador. Algunos tipos de eventos son: Onclick, mousedown, mouseup. Para click, apretar el botón, soltar el botón, etc.

Para realizar un evento se requiere determinar qué componente realizó para determinar cual es el elemento involucrado se busca el identificador id o class. En el siguiente ejemplo se puede ver como se relaciona un id o class con un evento.

Ejemplo 1: Para una imagen con id="img1", que requiere que se ejecute la función funcion1() cuando se realice un click sobre ella queda escrita de la siguiente manera.

```
$("#img1").onclick(funcion1)
```

Cabe destacar que la función “#” es la que permite identificar el llamado a un “id”. Para señalar que se emplea un componente tipo “class” se emplea un punto antes de la palabra reemplazando a “#”, como se indica a continuación.

Ejemplo 2: Para el elemento class="im1", ejecutar la función 1 para el evento onclick.

```
$(".im1").onclick(funcion1)
```

Un ejemplo de una animación. En este caso los eventos son mousedown y mouseup que crean la idea de un botón apretado y suelto. Para ello, tenemos el id= “leccSig” y las funciones que ejecuta el evento LeccSig() e img_ cuyo orden muestran como se presiona el botón.

Ejemplo 3:

```
x=$("#leccSig");  
x.mousedown(img_).mouseup(LeccSig);
```

La función img_() es la encargada de cambiar la imagen 1 por la imagen 2.

imagen 1



imagen 2



La función `LeccSig()` modifica los índices y el código html que se muestra junto con volver la imagen mostrada a imagen 1.

Para modificar el contenido del código html referenciado por un id o una class se realiza como se ve el siguiente ejemplo.

Ejemplo 4: Para el código html:

```
<div id="ID1">Hola</div>
```

Pero ahora se quiere modificar el contenido por Chao, para ello se empleará el siguiente código:

```
x=$("#ID1");  
x.html("Chao");
```

El ejemplo 4 es muy importante dentro del código, ya que, esto permite que una página estática se pueda modificar, dependiendo de la necesidad del usuario. Esto permite que toda el contenido de la lección se pueda visualizar sin necesidad sin realizar una llamada al servidor.

c) Estructuras y arreglos

Parte importante del empleo del lenguaje JavaScript es la definición de estructuras y arreglos, dado que se emplearán para la transferencia de lecciones como un sistema ordenado que contiene la información. Esta definición permite referenciar una lección o ejercicio por medio de un índice.

Una estructura en JavaScript se representa por un campo nombre y un campo valor separado por 2 puntos:

{nombre: valor}

Una arreglo en el lenguaje se representa por paréntesis cuadrados separados por comas:

[1,2,3,4]

Entonces, una estructura que posee un arreglo esta representada por:

{nombre:[1,2,3,4]}

Ahora, una estructura típica de lección es la siguiente:

```
{  
  "read":{  
    "inst": "Read the following text.",  
    "titl": "Let's talk about my classroom",  
    "lecc": ["Pedro: Hi, Tomás. How are you?",  
            "Rosita: Fine. Look"]  
  }  
}
```

donde:

“read”, es el nombre de la estructura que corresponde a la actividad lectora.

“inst”, son las instrucciones.

“titl”, es el título.

“lecc”, el contenido que se encuentra en un arreglo, por lo que, se puede referenciar por medio de un índice.

Ejemplo de referenciación es:

```
read.lecc[0]
```

Esta estructura incorpora diversos campos entre los que se encuentra el campo Test. Este incorpora el nombre de los archivos de audio que se usan en la lección respectiva los cuales se encuentran indexados de igual manera que ellas. Al igual que el resto de la estructura es un arreglo de JavaScript.

Ejemplo del código de la estructura es:

```
"Test":[  
    "C5_L1_A2_E1",  
    "C5_L1_A2_E2",  
    "C5_L1_A2_E3",  
    "C5_L1_A2_E4",  
    "C5_L1_A2_E5",  
    "C5_L1_A2_E6",  
    "C5_L1_A2_E7",  
    "C5_L1_A2_E8"]
```

El valor Test[0] corresponde al nombre del archivo de audio que se encuentra en un directorio denominado audio que se encuentra en el PATH del Applet.

d) La mnemotecnia del campo valor.

- CX. Curso al que pertenece la lección, donde X corresponde a un número entero.

- LX. Lección, número.

- AX. Actividad, número.

- EX. Ejercicio, número.

e) Acceso a la BD desde JavaScript.

Uno de los principales avances de la programación web es que se puede modificar el código HTML por medio de distintas funciones como ya se ha visto. Sin embargo, el poder acceder a datos, imágenes, sonidos, en formas asincrónicas entrega una gran versatilidad y fuerza a este tipo de sistemas. El avance continuo es lo que permite la gran popularidad de estos sistemas de manera que se pueden encontrar desde simples páginas html a sofisticados sistema de control industrial y, por supuesto, cursos de enseñanza de un segundo idioma.

Para acceder a la BD desde JavaScript y poner a disposición del cliente las distintas lecciones, actividades y ejercicios se utiliza la función.

```
$.getJSON("jsp/Leccion.jsp",{Leccion:leccion,Actividad:actividad},llegadaDatos);
```

donde se pueden identificar 3 campos:

- dirección url: "jsp/Leccion.jsp".
- Variables de datos: {Leccion:leccion,Actividad:actividad}.
- La función que recibe los datos: llegadaDatos(datos).

Como ésta es una función embebida en una página web, el campo url no tiene que estar completo, ya que, se encuentra definido en la cabecera de la página.

`<head>`

`<base href="http://localhost/EnglishWeb/">`

`</head>`

Es claro que las variables enviadas a la función jsp viajan en una estructura JavaScript que puede tener tantos campos como se quiera. La función a la que llegan los datos recibe el resultado en sus variables de entrada, en este caso la variable datos.

f) Acceso a la BD desde páginas JSP

Distintas implementaciones de lenguaje se han realizado para dotar de dinamismo a las páginas html, desde los antiguos CGI de apache que han evolucionado hasta el desarrollo completo de un lenguaje como el caso de PHP o implementaciones propietarias como es el caso de ASP. En el servidor http está basado en Tomcat, posee un interpretador de código java que permite realizar una gran cantidad de procesos, análisis, creación de imágenes, todo en el tiempo en que se está navegando y con todas las librerías que posee java.

Estas páginas denominadas JSP pueden encontrarse embebidas en lenguaje html y para ello, existen unos delimitadores de código. Este delimitador comienza por `<%` y termina con `%>`, de manera que, lo que se encuentra entre estas delimitaciones es código java en html, que puede ser desde un simple `out.print` hasta complejos programas de análisis.

Como lenguaje requiere de algunas definiciones de librería, estas se realizan de la siguiente manera:

```
<%@ page contentType="text/html; charset=iso-8859-1" language="java"
import="java.sql.*" errorPage="" %>
```

Junto a esto, se pueden cargar en memoria librerías específicas. Como por ejemplo:

```
<%@ page import="java.net.*" %>
```

```
<%@ page import="java.util.*" %>
```

```
<%@ page import="java.io.*" %>
```

```
<%@ page import="javax.sql.*" %>
```

```
<%@ page import="java.sql.*" %>
```

```
<%@ page import="javax.naming.*" %>
```

Las librerías fundamentales son: java.net, java.util y java.io. Sin ellas, no se puede hacer uso de la gran potencialidad de las páginas JSP.

Como lo importante de esta implementación es el acceso a la BD, se ve a continuación un ejemplo en que se acceda a ella.

Ejemplo 5:

```
<%@ page contentType="text/html; charset=iso-8859-1"
language="java" import="java.sql.*" errorPage="" %>
<%@ page import="java.net.*" %>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<%@ page import="javax.sql.*" %>
<%@ page import="java.sql.*" %>
<%@ page import="javax.naming.*" %>
<%
String curso=(String)session.getAttribute("curso");
/* Esto es un comentario */
try
{
    try {
        Class.forName("org.postgresql.Driver");
    } catch (ClassNotFoundException e) {
        out.println("<h1>No se encuentra el Driver: " + e.getMessage() + "</h1>"
);
    }

String driver = "org.postgresql.Driver";
String connectString = "jdbc:postgresql://localhost/bd_eloud";
String BDuser = "user_eloud";
String BDpasswd= "lptv_nbecerra";
```

```
String Query = "select valor from contenido where curso="+curso+"
and leccion="+leccion+" and actividad="+actividad;
```

```
Class.forName(driver).newInstance();
Connection conn = DriverManager.getConnection(connectString,
BDuser,BDpasswd);
ResultSet rs = conn.createStatement().executeQuery(Query);
String colName = rs.getMetaData().getColumnLabel(1);
rs.next();
String pass=rs.getString("valor");
out.println(pass );
}
```

```
catch (Exception e)
{
out.print("Excepcion : 2" + e.getMessage() + "Fin\n");
} %>
```

La línea `Class.forName("org.postgresql.Driver");` verifica la existencia del controlador de la BD.

El código `String driver = "org.postgresql.Driver";` define el controlador que se empleará y `String connectString = "jdbc:postgresql://localhost/bd_eloud";` entrega la dirección de la máquina, donde se encuentra y el nombre de la BD, en nuestro caso "bd_eloud".

Una cuestión importante de la conexión a BD es la necesidad de autenticar al usuario, el cual posee un password y permisos de conexión que se define como un "grand" dentro de la BD o al momento de su creación.

En la línea `Connection conn = DriverManager.getConnection(connectString, BDuser,BDpasswd);` se incorpora el nombre de usuario y el password al conector.

Una consulta en sql se puede enviar a la BD y la forma de enviarla se ve en el punto es la siguiente `ResultSet rs = conn.createStatement().executeQuery(Query);`. Donde la consulta en este

caso es `String Query = "select valor from contenido where curso="+curso+" and leccion="+leccion+" and actividad="+actividad;`

El conector devuelve los resultados por medio de un puntero desde el cual se obtiene la respuesta a la consulta. Para acceder a este resultado se debe llamar a la clase `rs.next()`;

Para acceder a un campo de la consulta se emplea `rs.getString("valor")`;

Para que el resultado de la consulta se envíe a la página web se aplica `out.println(valor)`;

CAPITULO 4

4. PRUEBA DE CARGA

En este capítulo se da a conocer el software y el hardware que se empleó para las pruebas de robustez, estabilidad del sistema y los resultados de las mismas. El principal requisito que se le exige al sistema es que permita la interacción simultánea de 150 usuarios.

4.1. El software empleado en las pruebas

El software que se empleó en las pruebas de carga es Jmeter. Este software está realizado en JAVA y fue diseñado para medir el rendimiento de los servidores Web. Es un software libre, por lo que, no tiene costo de adquisición, posee una interface gráfica que lo hace fácil de usar, entrega los resultados en forma de gráficos o por medio de un archivo con datos separados por coma.

Este programa permite simular el comportamiento de un usuario al enfrentar un sitio web cualquiera, ya que posee un proxy interno que guarda los datos de navegación. Estos datos son repetidos en forma paralela hasta completar el límite de usuario simulados.

Es importante mencionar que el rendimiento de un sistema está íntegramente relacionado con el hardware del computador que funcionará como servidor, ya que, la capacidad de proceso, almacenamiento y memoria son los límites reales a los que se enfrenta el sistema.

4.2. El Hardware para la realización de las pruebas

Para la realización de las pruebas de rendimiento se emplearon dos computadoras, una de las cuales correspondía a un servidor y la otra como cliente. Las características de los computadores se pueden ver en la Tabla 3:

	Servidor	Cliente
CPU	Intel Core i5	Intel Core 2 Quad
Memoria RAM	2G Byte	2G Byte
Vel. Procesador	3,2 Ghz	2,4 Ghz
Sistema Operativo	Windows Xp	Windows Xp
Disco Duro	320 G Byte	320 G Byte

Tabla 3: Hardware servidor y cliente, para prueba de rendimiento.

Ambos computadores se encontraban en redes distintas separados por un switch, por lo que, las pruebas se realizaron en torno a la carga más que a los retrasos en la red.

4.3. La prueba de rendimiento

La prueba de rendimiento consistió en realizar 150 conexiones simultáneas, simulando el comportamiento de los usuarios que acceden por medio de un login y password a la página de resumen, se realizan los 8 ejercicios de la lección 1, se consultaron las notas y posteriormente abandonan el sistema. Esto mostrará el comportamiento en demanda del servidor y los servicios asociados.

El resultado de estas pruebas se encuentra en la tabla N° 4. Las filas representan los objetos involucrados en una petición y las columnas las medidas dentro de la prueba. A continuación se explica el contenido de cada columna de la tabla de respuesta.

- Label. Representa el objeto involucrado en la petición.
- #Muestra. El número de veces que se solicitó el objeto al servidor.
- Media. Corresponde al promedio o media aritmética medido en [ms].
- Mediana. Medida en [ms].
- Mín. Tiempo mínimo medido en [ms].
- Máx. Tiempo máximo medido en [ms].
- %Error. El porcentaje de errores ocurridos en el requerimiento.
- Rendimiento. Medido en peticiones por segundo.
- Kb/sec. Medida de velocidad en Kb/Seg.

4.4. Discusión

Las Pruebas de carga se realizaron con computadores de similares características, como se expuso en la tabla N° 3 y se realizaron dentro de una Lan de 100MB/s con otros usuarios conectados a la red. La conexión de las computadoras se realizó por medio de un switch de igual velocidad. En la tabla N° 4 se pueden apreciar en la fila Totales el desempeño del servidor y sus componentes considerando 150 usuarios simultáneos. Cabe señalar que el tiempo máximo de respuesta se encuentra en la transmisión de los archivos de audio y fue de 3.177 [ms], el mínimo para igual elemento fue de 16 [ms]. Estos resultados mejoran si consideramos la línea del 90% de los usuarios ya que el tiempo máximo es de 1.125[ms].

Desde el punto de vista de la BD, el desempeño lo podemos observar en las funciones *avance.jsp*, *Leccion.jsp* y *notas.jsp*, que poseen una complejidad similar y tiempos de igual magnitud. El tiempo máximo observado fue de 1.900[ms] y el mínimo fue de 18[ms]. Si se considera para este caso la línea del 90% encontramos que el tiempo máximo fue de 1.420[ms]. Otra cuestión importante a destacar es que en la función *checkLogin.jsp* posee tiempos de respuesta máximo de 933[ms] y mínimo de 256[ms] con línea del 90% de 815[ms]. Tiempos relativamente altos para la complejidad de la función, esta función es la encargada de verificar el login y el password. Por lo que podemos suponer que el tiempo adicional se debe al tiempo que toma realizar una conexión vía socket entre el servidor web y la BD.

Los tiempos de respuesta se encuentran bastante bien, vistos desde la experiencia web, esto dado que la mediana es de 215[ms] y el limite del 90% esta en 1.055[ms].

El desempeño global del sistema lo podemos observar en la fila totales y la columna rendimiento y corresponde de 295 peticiones por segundo.

Tabla de Resultados entregada por JMETER									
Label	# Muestras	Media	Mediana	Linea de 90%	Min	Max	% Error	Rendimiento	Kb/sec
/EnglishWeb/	150	170	139	340	7	386	0	54,6249	149,9518
/EnglishWeb/img/pronunciacion_r1_c2.gif	300	255	251	481	10	566	0	33,7610	1299,6656
/EnglishWeb/img/pronunciacion_r3_c4.gif	150	209	171	386	7	425	0	45,6621	325,3871
/EnglishWeb/img/pronunciacion_r14_c2.gif	300	77	37	241	1	350	0	26,6999	19,3991
/EnglishWeb/img/pronunciacion_r5_c5.gif	150	208	209	353	14	430	0	40,8719	60,1903
/EnglishWeb/img/pronunciacion_r2_c2.gif	300	168	154	309	8	431	0	31,8674	20,4773
/EnglishWeb/img/pronunciacion_r4_c22.gif	300	87	66	246	1	329	0	26,9107	15,0584
/EnglishWeb/img/pronunciacion_r2_c23.gif	300	154	145	297	1	393	0	30,4878	18,3105
/favicon.ico	450	77	22	216	2	296	1	10,8719	0,0000
/tbproxy/af/query	300	374	298	593	188	1821	0	7,2539	0,8855
/EnglishWeb/jsp/checkLogin.jsp	150	571	559	815	256	933	0	29,4060	21,5089
/EnglishWeb/course.jsp	150	238	232	326	80	395	0	29,8745	21,8516
/EnglishWeb/javascript/jquery.alerts.js	150	334	354	482	119	565	0	28,9911	213,8377
/EnglishWeb/css/jquery.alerts.css	150	235	217	351	95	416	0	27,9070	30,4960
/EnglishWeb/javascript/jquery.js	150	729	764	917	444	1002	0	24,5419	1729,7715
/EnglishWeb/javascript/jquery.ui.draggable.js	150	375	365	496	223	605	0	25,3079	294,8965
/EnglishWeb/javascript/jpage.js	150	392	389	507	237	600	0	25,5145	695,7447
/EnglishWeb/css/EnglishLoud.css	150	356	342	525	174	609	0	25,8264	36,6715
/EnglishWeb/img/leccAnt.gif	150	218	202	366	93	426	0	27,7213	16,6220
/EnglishWeb/img/home.gif	150	213	197	371	83	416	0	28,2592	17,8276
/EnglishWeb/img/pronunciacion_r2_c3.gif	150	175	160	285	78	365	0	28,5225	11,1973
/EnglishWeb/img/home1.gif	150	183	167	342	62	423	0	28,8184	14,9439
/EnglishWeb/img/Salir.gif	150	179	161	323	44	392	0	29,4002	25,4668
/EnglishWeb/img/summary_r5_c7.gif	150	156	113	298	33	431	0	31,7393	20,3330
/EnglishWeb/img/pronunciacion_r6_c7.gif	150	148	110	313	2	399	0	32,9815	9,3083
/EnglishWeb/img/leccSig.gif	150	133	98	342	1	453	0	33,4747	19,1891
/EnglishWeb/img/pronunciacion_r6_c18.gif	150	115	91	283	1	353	0	35,9885	6,6073
/EnglishWeb/img/pronunciacion_r6_c8.gif	150	110	84	310	1	415	0	38,6698	6,4953
/EnglishWeb/img/msg_dictado.png	150	156	114	495	2	619	0	37,9747	136,6199
/EnglishWeb/img/chequearDict.gif	150	87	61	202	1	421	0	41,0734	22,7428
/EnglishWeb/img/dictation_r6_c2.png	150	67	50	165	1	262	0	44,3787	11,9181
/EnglishWeb/img/escucharDict.png	150	87	39	271	2	300	0	46,1965	114,4085
/EnglishWeb/img/pronunciacion_r10_c8.gif	150	50	15	157	1	203	0	48,9077	10,0299
/EnglishWeb/img/dictation_r7_c7.png	150	41	8	113	1	180	0	50,9511	10,0011
/EnglishWeb/img/dictation_r7_c2.png	150	37	9	111	1	160	0	53,3618	10,6307
/EnglishWeb/img/dictation_r8_c2.png	150	33	9	107	1	134	0	56,0748	16,4829
/EnglishWeb/img/escuchar.gif	150	27	4	95	1	158	0	58,4340	29,9018
/EnglishWeb/img/escuchartexto.gif	150	24	3	91	1	152	0	60,7780	47,2454
/EnglishWeb/img/grabar.gif	150	22	3	88	1	151	0	63,3446	35,8788
/EnglishWeb/img/chequear.gif	150	13	3	55	1	92	0	73,9280	40,9348
/EnglishWeb/jsp/avance.jsp	1050	944	984	1388	55	1900	0	36,2256	2,7753
/EnglishWeb/applet/English.jar	150	14	11	32	8	49	0	57,7367	2483,9194
/EnglishWeb/jsp/Leccion.jsp	1500	989	993	1420	35	1841	0	52,8318	4,1095
/EnglishWeb/jsp/notas.jsp	750	716	981	1223	18	1875	0	23,9395	3,9804
/EnglishWeb/applet/au/lesson1_tr_18.au	150	409	306	838	13	1818	0	16,5089	1873,2459
/EnglishWeb/applet/au/lesson1_tr_17.au	150	592	650	979	15	2555	0	14,6915	1878,2998
/EnglishWeb/applet/au/lesson1_tr_16.au	150	658	896	1125	17	2044	0	13,1648	1767,2490
/EnglishWeb/applet/au/lesson1_tr_15.au	150	636	786	1010	16	3007	0	11,5128	1324,7341
/EnglishWeb/applet/au/lesson1_tr_14.au	150	592	771	1011	17	2531	0	11,1924	1341,5092
/EnglishWeb/applet/au/lesson1_tr_13.au	150	509	629	796	16	3177	0	11,0375	1251,2288
/EnglishWeb/applet/au/lesson1_tr_12.au	150	555	543	999	20	2130	0	10,9123	1604,3588
/EnglishWeb/applet/au/lesson1_tr_11.au	150	310	243	637	13	1241	0	10,7751	1033,2510
/EnglishSocket/EnglishSocket	300	912	1009	1472	4	2225	0	19,9203	0,7667
/EnglishWeb/jsp/Salir.jsp	150	61	4	124	2	615	0	11,7463	0,0574
Peticion Java	150	218	213	320	97	353	0	11,6090	0,0000
TOTAL	12450	422	215	1055	1	3177	0,0361	295,2476	4416,1204

Tabla 4: Tabla de resultados Jmeter. Los 4 totales están medidos en [ms] y el rendimiento se encuentra en [s]

CAPITULO 5

5. CONCLUSIONES.

La plataforma desarrollada ha demostrado cumplir con los distintos objetivos planteados, en especial desde el punto de vista del rendimiento. Presenta además una ventaja en la incorporación de nuevos servicios al estar estos comunicados por socket. Puede responder rápidamente a un aumento de la demanda y posee una arquitectura que facilita la rápida actualización del sistema. El empleo del modelo MVC permiten realizar modificaciones en forma segmentada sin que se vean afectadas las otras partes, esto es, que se puede modificar la presentación, el funcionamiento lógico o el contenido indistintamente. Por otro lado, que el sistema se encuentre en línea permite incorporar rápidamente innovaciones, nuevos módulos o lecciones sin tener que recurrir en forma física donde se encuentre el usuario.

Que el sistema posea diversos tipos de usuarios ha permitido personalizar el acceso, brindando a las distintas partes de la estructura educativa una interface. Ello permite que los distintos involucrados en el proceso educativo puedan retroalimentar el programa, lo que redundará en una mejora de éste en la medida que se analicen las necesidades de los distintos actores.

La incorporación del registro de actividades de los usuarios y el almacenamiento de voz, brinda grandes posibilidades desde el punto de vista del administrador, ya que, permiten mejorar la usabilidad de la plataforma, encontrar cuellos de botella o tamaños no adecuados de los objetos. Esta información permite no sólo mejorar el desempeño sino también determinar posibles errores en la programación. El almacenamiento de voz permite ampliar los datos para el análisis de señales y mejorar el desempeño del ASR en especial, en lo que se refiere a las lecciones que se encuentran ingresadas.

El sistema presenta una alta escalabilidad, ya que, está basado en servicios, los cuales pueden habitar en una u otra máquina indistintamente, por lo que, se propone que el servlet permita en una futura versión que posea también esta capacidad de direccionar mediante IP en una forma más transparente, lo que redundará en poder satisfacer grandes demandas de ser el caso.

Se sugiere para trabajo a futuro la incorporación de HTML 5, en especial las librerías JavaScript socket.io.js y node.js, ya que, permite trabajar con elementos multimediales en forma nativa dentro del código de la página web, incorporando las tramas de voz por medio de JavaScript y el envío al servidor directo a puertos específicos. Sin embargo será necesario mantener compatibilidad con navegadores antiguos por medio del applet.

Bibliografía

- **Tanenbaum Andrew, 2003.** “Redes de computadoras 4ª edición”.
- **Molina C., Becerra Yoma N., Wuth, J., and Vivanco, H., (2009),** ASR based pronunciation evaluation with automatically generated competing vocabulary and classifier fusion. *Speech Communication, Elsevier.* Vol. 51 (6), Pages 485-498, June.
- **Molina C, 2010,** “Investigación en tecnología de reconocimiento de voz aplicada a enseñanza de segundo idioma con adaptación y cancelación no-supervisada de ruido”.
- **Jorge Wuth, 2007,** “Desarrollo de aplicación de reconocimiento de voz para telefonía celular”.
- **Tanenbaum Andrew,2000.** “Organización de Computadoras:un enfoque estructurado”.
- **Morris M, 1994.** “Arquitectura de Computadoras”.
- **Sanders William B, 2001.** “JavaScript Design”.
- **Mahemoff Michael, 2006.** “AJAX Design Patterns”.
- **Brown Tracy Steven, 2006.** “Dynamic Apache with AJAX”.
- **The PostgreSQL Global Development Group, 1996-2010.** “Manual PostgreSQL 8.4”.
- **Resig John, 2010.** “jQuery Cookbook: Solutions & Examples for jQuery Developers (Animal Guide).”
- **Kurniawan Budi. 2002.** “Java for Web, with Servlets, JSP and EJB: A Developers Guide with J2EE Solutions”.
- **Subramaniam Venkat. 2010.** “Servlets, JSP, Struts and MVC”. [en línea], <<http://www.agiledeveloper.com/articles/JSPMVC.pdf>>, [consultado junio 2010].

- **Hall Marty,2010.** “Integrating Servlet and JSP:The Model View Controller Architecture.” [en línea],<<http://pdf.coreservlets.com/MVC.pdf>>, [consultado junio 2010].
- **Márquez Avendaño,Bertha Mariel . Tesis 2004.** “Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español”. [en línea], <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/>
- **Anastacio Velásquez, Miguel.** “ Model View Controller”. [en línea], http://www.informatizate.net/articulos/model_view_controller_mvc_20040324.html, [Consultado en agosto 2010].
- **Stonebraker, Michael. 1987.** “The Design of de The Postgres Storaje System”.
- **Sánchez, Jorge.2004.** “Diseño conceptual de Base de datos”.
- **Rob, Peter/Coronel, Carlos. 2004.** “Sistemas de bases de datos: Diseño, implementación y Administración”.
- **Bravo Crecencio, 2005.** “Sistemas interactivos y colaborativos en la WEB”.
- **Márquez Santiago,2007.**“La WEB Semántica”.
- **Sanchez José/Ruiz Julio,** “Manual de GIMP”.
- **Meyer Eric, 2006.** “ CSS:The Definitive Guide”.