

Diseño e implementación de una aplicación web orientada al análisis sintáctico de sentencias SQL -DML en base de datos MySQL.

Documento proyecto de grado

Autor:

Víctor Julio Escobar Safar

2014



**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB
ORIENTADA AL ANÁLISIS SINTÁCTICO DE SENTENCIAS SQL-
DML EN BASE DE DATOS MYSQL.**

Trabajo presentado por:

VICTOR JULIO ESCOBAR SAFAR

**Trabajo presentado al Departamento de Ingeniería de Sistemas y
Telecomunicaciones, en cumplimiento parcial de los requisitos para
obtener el título de Ingeniero de Sistemas.**

DIRECTOR: ING. JAVIER PENICHE PADILLA

**UNIVERSIDAD DE CÓRDOBA
FACULTAD DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES
MONTERÍA**

2014

NOTA DE ACEPTACIÓN

DIRECTOR

JURADO 1

JURADO 2

DEDICATORIA

A Dios, por permitirme llegar a este momento tan especial en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más. A mi madre por ser la persona que me ha acompañado durante todo mi trayecto estudiantil y de vida. A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

A mis profesores, gracias por su tiempo, por su apoyo así como por la sabiduría que me transmitieron en el desarrollo de mi formación profesional.

Victor Julio Escobar Safar

AGRADECIMIENTOS

En primer lugar a Dios por haberme guiado por el camino correcto en segundo lugar a mi madre que me dio fuerzas para salir adelante, mi familia y a los que confiaron en mí que me apoyaron sin ningún compromiso y que con su cariño y amor me han motivado, este proyecto es el esfuerzo y estudio por eso agradezco a todos mis profesores, mis compañeros que estuvieron ahí incondicionalmente y quienes a lo largo de este tiempo han puesto a prueba mis capacidades y conocimientos en el desarrollo de esta gran meta, finalmente agradezco a esta universidad la cual abrió sus puertas a jóvenes como yo, preparándonos para un futuro competitivo y formándonos como personas de bien.

Muchas gracias

Victor Julio Escobar Safar

TABLA DE CONTENIDO

| | |
|--|-----------|
| 1. OBJETIVOS | 12 |
| 1.1 OBJETIVO GENERAL | 12 |
| 1.2 OBJETIVOS ESPECÍFICOS | 12 |
| 2. INTRODUCCION | 13 |
| 2.1 AMBIENTACIÓN | 13 |
| 2.2 PROBLEMÁTICA | 14 |
| 2.3 ANTECEDENTES | 15 |
| 2.4 JUSTIFICACION | 17 |
| 3. MARCO TEORICO Y MARCO CONCEPTUAL | 18 |
| 3.1 SOFTWARE LIBRE | 18 |
| 3.2 LENGUAJE SQL | 19 |
| 3.2.1 LENGUAJE DE DEFINICIÓN DE DATOS (DDL). | 20 |
| 3.2.1.1 CREATE CREAR | 20 |
| 3.2.1.2 ALTER MODIFICAR | 21 |
| 3.2.1.3 DROP ELIMINAR | 21 |
| 3.2.1.4 TRUNCATE BORRAR TABLA | 21 |
| 3.2.2 LENGUAJE DE MANIPULACIÓN DE DATOS (DML). | 21 |
| 3.2.2.1 SELECT | 22 |
| 3.2.2.2 INSERT | 22 |
| 3.2.2.3 UPDATE | 23 |
| 3.2.2.4 DELETE | 23 |
| 3.3 ANALIZADOR SINTACTICO | 24 |
| 3.4 DEPURACIÓN DECLARATIVA | 25 |
| 3.5 APPLETS JAVA. | 26 |
| 3.5.1 ARQUITECTURA DE UN APPLLET. | 26 |
| 3.5.2 ESTRUCTURA DE UN APPLLET. | 27 |
| 3.6 APLICACIONES WEB | 28 |
| 3.6.1 ARQUITECTURA DE APLICACIONES WEB | 29 |
| 3.7 SERVIDORES WEB | 30 |
| 3.7.1 SERVIDOR WEB APACHE | 30 |
| 3.8.1.1 CARACTERÍSTICAS DEL SERVIDOR APACHE | 31 |
| 3.8 WEB SERVICE | 31 |
| 3.8.1 SERVIDORES DE APLICACIONES PARA SERVICIOS WEB: | 32 |

| | | |
|-------------|---|-----------|
| 3.9 | AUTÓMATAS FINITOS | 33 |
| 3.9.1 | FUNCIONAMIENTO | 34 |
| 3.10 | AUTÓMATA DE PILA | 34 |
| 3.11 | BBDD: MySQL. | 35 |
| 3.12 | LENGUAJE DE PROGRAMACIÓN: JAVA. | 36 |
| 3.13 | PHP | 38 |
| 3.14 | HTML | 39 |
| 3.15 | DREAMWEAVER | 40 |
| 3.16 | NETBEANS | 41 |
| 4. | METODOLOGIA | 42 |
| 4.1 | FASES DEL PROYECTO | 42 |
| 4.2 | METODOLOGIA DE DESARROLLO DEL PRODUCTO | 55 |
| 5. | DESARROLLO | 57 |
| 5.1 | ARQUITECTURA DEL SISTEMA | 57 |
| 5.1.1 | MODELO MVC | 57 |
| 5.2 | DISEÑO DEL SISTEMA | 60 |
| 5.2.1 | MODELO ENTIDAD RELACIÓN | 60 |
| 5.2.2 | DIAGRAMA DE CLASE | 61 |
| 5.2.3 | DEFINICIÓN DE USUARIOS DE SISTEMA. | 62 |
| 5.2.3.1 | USUARIOS DEL SISTEMA | 62 |
| 5.2.4 | DIAGRAMAS DE CASOS DE USOS | 63 |
| 5.2.4.1 | DIAGRAMA DE CASO DE USO (ACCESO AL SISTEMA) | 63 |
| 5.2.4.2 | DIAGRAMA DE CASO DE USO (ADMINISTRADOR) | 64 |
| 5.2.4.3 | DIAGRAMA DE CASO DE USO (USUARIO) | 65 |
| 5.2.4.4 | DIAGRAMA DE CASO DE USO (REGISTRO DE USUARIO) | 66 |
| 5.2.4.5 | DIAGRAMA DE CASO DE USO APLICACIÓN | 67 |
| 5.2.5 | DIAGRAMAS DE SECUENCIAS | 68 |
| 5.2.5.1 | DIAGRAMA DE SECUENCIA ACCESO DEL SISTEMA. | 68 |
| 5.2.5.2 | DIAGRAMA DE SECUENCIA DEL ADMINISTRADOR | 69 |
| 5.2.5.3 | DIAGRAMA DE SECUENCIA REGISTRAR USUARIO | 70 |
| 5.2.5.4 | DIAGRAMA DE SECUENCIA USUARIO | 71 |
| 5.2.5.5 | DIAGRAMA DE SECUENCIA APLICACION | 72 |
| 5.2.6 | DIAGRAMAS DE COLABORACION | 73 |
| 5.2.6.1 | DIAGRAMA DE COLABORACION ACCESO AL SISTEMA | 73 |
| 5.2.6.2 | DIAGRAMA DE COLABORACION ADMINISTRADOR | 74 |
| 5.2.6.3 | DIAGRAMA DE COLABORACION REGISTRO DE USUARIO | 74 |
| 5.2.6.4 | DIAGRAMA DE COLABORACION USUARIO | 75 |
| 5.2.6.5 | DIAGRAMA DE COLABORACION APLICACION | 76 |
| 6. | CONCLUSIONES | 77 |

| | |
|--|------------|
| 7. BIBLIOGRAFÍA | 78 |
| 8. ANEXOS | 80 |
| 8.1 DOCUMENTACION DE DIAGRAMA DE CASOS DE USO | 80 |
| 8.2 DIAGRAMAS DEL SISTEMA | 85 |
| 8.2.1 ARQUITECTURA DEL SISTEMA | 85 |
| 8.3 MANUAL DE USUARIO | 86 |
| 8.4 MANUAL DE INSTALACION | 104 |
| 9. EVIDENCIAS | 109 |

LISTA DE FIGURAS

| | |
|--|----|
| FIGURA 1 ESTRUCTURA DEL ANALIZADOR SINTÁCTICO | 25 |
| FIGURA 2 APPLETS JAVA | 26 |
| FIGURA 3. ESQUEMA NIVELES DE APLICACIONES WEB MVC..... | 29 |
| FIGURA 4 AUTÓMATA FINITO | 33 |
| FIGURA 5: ICONO MYSQL | 35 |
| FIGURA 6: ICONO JAVA..... | 36 |
| FIGURA 7: ICONO PHP..... | 38 |
| FIGURA 8: ICONO HTML | 39 |
| FIGURA 9 ICONO DREAMWEAVER..... | 40 |
| FIGURA 10 ICONO NETBEANS..... | 41 |
| FIGURA 11 NETBEANS APPLET..... | 44 |
| FIGURA 12 WEB SERVICE..... | 45 |
| FIGURA 13 ENCUESTA PREGUNTA # 1 | 47 |
| FIGURA 14 ENCUESTA PREGUNTA # 2 | 47 |
| FIGURA 15 ENCUESTA PREGUNTA # 3 | 48 |
| FIGURA 16 ENCUESTA PREGUNTA # 4 | 48 |
| FIGURA 17 ENCUESTA PREGUNTA # 5 | 49 |
| FIGURA 18 ENCUESTA PREGUNTA # 6 | 49 |
| FIGURA 19 ENCUESTA PREGUNTA # 7 | 50 |
| FIGURA 20 ENCUESTA PREGUNTA # 8 | 50 |
| FIGURA 21 DIAGRAMA DE RESULTADOS EJERCICIOS..... | 52 |
| FIGURA 22 MODULO REPORTE DE USUARIO | 53 |
| FIGURA 23 APLICACIÓN APPLET ANALIZANDO | 55 |
| FIGURA 24 ARQUITECTURA DEL SISTEMA..... | 59 |
| FIGURA 25 MODELO ENTIDAD RELACIÓN..... | 60 |
| FIGURA 26 DIAGRAMA DE CLASE..... | 61 |
| FIGURA 27 USUARIOS DEL SISTEMA | 62 |
| FIGURA 28 DIAGRAMA DE CASO DE USO ACCESO AL SISTEMA..... | 63 |
| FIGURA 29 DIAGRAMA DE CASO DE USO ADMINISTRADOR | 64 |
| FIGURA 30 DIAGRAMA DE CASO DE USO USUARIO | 65 |
| FIGURA 31 DIAGRAMA DE CASO DE USO REGISTRO DE USUARIO | 66 |
| FIGURA 32 DIAGRAMA DE CASO DE USO APLICACIÓN | 67 |
| FIGURA 33 DIAGRAMA DE SECUENCIA ACCESO DEL SISTEMA | 68 |
| FIGURA 34 DIAGRAMA DE SECUENCIA ADMINISTRADOR | 69 |
| FIGURA 35 DIAGRAMA DE SECUENCIA REGISTRO DE USUARIO..... | 70 |
| FIGURA 36 DIAGRAMA DE SECUENCIA USUARIO | 71 |
| FIGURA 37 DIAGRAMA DE SECUENCIA APLICACIÓN..... | 72 |
| FIGURA 38 DIAGRAMA DE COLABORACIÓN ACCESO AL SISTEMA..... | 73 |
| FIGURA 39 DIAGRAMA DE COLABORACIÓN ADMINISTRADOR..... | 74 |
| FIGURA 40 DIAGRAMA DE COLABORACIÓN REGISTRO DE USUARIO | 74 |
| FIGURA 41 DIAGRAMA DE COLABORACIÓN USUARIO..... | 75 |
| FIGURA 42 DIAGRAMA DE COLABORACIÓN APLICACIÓN | 76 |
| FIGURA 43 ARQUITECTURA DEL SISTEMA # 2 | 85 |
| FIGURA 44 APLICACIÓN WEB ÍNDEX..... | 86 |

| | |
|---|-----|
| FIGURA 45 REGISTRO DE USUARIO | 87 |
| FIGURA 46 MODULO DE USUARIO..... | 88 |
| FIGURA 47 MODULO PERFIL USUARIO..... | 89 |
| FIGURA 48 MODULO TUTORIAL | 90 |
| FIGURA 49 ICONO ACCESO A LA APLICACIÓN | 90 |
| FIGURA 50 ACERCA DE JAVA | 91 |
| FIGURA 51 CONFIGURACIÓN JAVA..... | 92 |
| FIGURA 52 ADVERTENCIA DE SEGURIDAD JAVA | 93 |
| FIGURA 53 DETALLE CERTIFICADO | 94 |
| FIGURA 54 APLICACIÓN APPLETT | 95 |
| FIGURA 55 PRUEBA APLICACIÓN APPLETT | 96 |
| FIGURA 56 APLICACIÓN APPLETT CARGAR ARCHIVO | 97 |
| FIGURA 57 PRUEBA APLICACIÓN CARGAR ARCHIVO | 98 |
| FIGURA 58 MODULO ADMINISTRADOR | 99 |
| FIGURA 59 MODULO PERFIL DE ADMINISTRADOR | 100 |
| FIGURA 60 MODULO REPORTE DE USUARIO | 101 |
| FIGURA 61 MODULO CREAR GRUPO | 102 |
| FIGURA 62 AVISO CERRAR SESIÓN..... | 103 |
| FIGURA 63 HOSTING..... | 105 |
| FIGURA 64 DOMINIO | 106 |
| FIGURA 65 REGISTRO DE DOMINIO | 106 |
| FIGURA 66 DNS..... | 107 |
| FIGURA 67 ARCHIVOS Y PAGINAS | 107 |
| FIGURA 68 BASE DE DATOS..... | 108 |

LISTA DE TABLAS

| | |
|---|----|
| TABLA 1 LIBERTADES DEL SOFTWARE LIBRE..... | 19 |
| TABLA 2 SENTENCIA SELECT. | 22 |
| TABLA 3 SENTENCIA INSERT | 22 |
| TABLA 4 SENTENCIA UPDATE | 23 |
| TABLA 5 SENTENCIA DELETE | 23 |
| TABLA 6 CAPA DE SOFTWARE DE LA JMV. | 37 |
| TABLA 7 DOCUMENTACIÓN DE DIAGRAMA DE CASOS DE USO #1 | 80 |
| TABLA 8 DOCUMENTACIÓN DE DIAGRAMA DE CASOS DE USO #2 | 81 |
| TABLA 9 DOCUMENTACIÓN DE DIAGRAMA DE CASOS DE USO #3 | 82 |
| TABLA 10 DOCUMENTACIÓN DE DIAGRAMA DE CASOS DE USO #4 | 83 |
| TABLA 11 DOCUMENTACIÓN DE DIAGRAMA DE CASOS DE USO #5 | 84 |

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Desarrollar una aplicación web orientada al análisis sintáctico de sentencias SQL-DML en base de datos MySQL.

1.2 OBJETIVOS ESPECÍFICOS

1. Identificar los requerimientos para el manejo del lenguaje estándar SQL-DML.
2. Diseñar una aplicación web dedicada al análisis sintáctico del lenguaje SQL.
3. Implementar pruebas de consulta por el usuario con sentencias SQL para verificar el correcto funcionamiento de la aplicación.

2. INTRODUCCION

2.1 AMBIENTACIÓN

Este proyecto pone en práctica todos los conocimientos y técnicas adquiridos, así como el análisis, planificación y desarrollo del mismo. Consistirá en el desarrollo una aplicación web orientada al análisis sintáctico¹ de sentencias SQL- DML² en base de datos MySQL, del que se pretenderá identificar los requerimientos necesarios para aplicación web como también el objeto de identificar el manejo de las sentencias del lenguaje SQL³; con el fin de implementarlo en pruebas de consultas.

El objetivo inmediato es analizar el manejo de las sentencias de los estudiantes durante los primeros pasos en las bases de datos o desarrollo web en la Universidad de Córdoba esta consistirá primero que todo en una encuesta, posteriormente el uso de la aplicación donde se dará a entender el buen uso de la misma.

Seguidamente se realiza pruebas de análisis de sentencia mediante dos formas, una es directa en que se podrá interactuar con una sola sentencia y analizar , la otra mediante un archivo introducido por el usuario con varias sentencias el cual deben contener la vista errónea o no, de la que será la que queremos analizar. Se implementara un sistema estadístico en el que se presentara el conteo de sentencias validas e inválidas generadas por un reporte de usuario y por ultimo todo esto integrado en una aplicación web para su correspondiente manipulación.

¹ Análisis sintáctico: recibe el texto de entrada y verifica que su orden sea el correcto.

² DML (Lenguaje de manipulación de datos)

³ SQL (Structured query language o sentencia de lenguaje estructurado)

2.2 PROBLEMÁTICA

En la actualidad las sentencias de lenguaje estructurado son esenciales para la búsqueda de información en las bases de datos, gracias a ellas se permite detallar diversos tipos de operaciones. Este lenguaje es estándar en la inmensa mayoría de los SGBD⁴ comerciales, debido a las buenas características que ofrece así como lo fácil de entender y de aprender. Las SGBD para analizar este tipo de sentencias usan interpretadores que juegan un papel importante pero a veces, sobre el código generado es necesario agregar o hacer modificaciones, por lo que es esencial ser capaz de depurar tales entradas, además de la falta de herramientas de filtrado y prueba, como también de una asistencia en su desarrollo y evidenciar la corrección de las consulta hacen de esta una carencia en el tratado y manejo de las sentencias ; esto fue planteado como inconvenientes en la investigación de Generación de casos de prueba para SQL investigado en Madrid, España, Facultad de Informática, Junio 2012.

El problema que se presenta para los que están empezando a dar sus primeros pasos en las bases de datos es que no tienen una idea clara de cómo es el manejo de la estructura de las sentencias SQL y al momento de realizar las consultas lo demuestran y aun sin ningún tipo de herramientas que faciliten la escritura, dificultan al usuario la ejecución de la misma.

⁴ SGBD: Sistema de Gestión de Bases de Datos.

2.3 ANTECEDENTES

Entre las investigaciones consultadas con respecto al diseño de aplicaciones web orientada al análisis sintáctico, a continuación se presentan varios proyectos integradores efectuados en los últimos años, cada uno con aportes valiosos para la experiencia en el proyecto.

Álvaro Mayorga, Estocolmo año: 2000 Corus Technologies

Es una empresa que ha diseñado y desarrollado Corus QL (CorusQuickLink), una potente y flexible herramienta de integración de aplicaciones.

El objetivo de esta tesis es investigar la aplicación de un depurador PL/SQL con un extremo inicial ejecutado en un navegador web y un paquete de PL/SQL basado en DBMS_DEBUG depuración de Oracle API, para determinar si es realizable o no. Un estudio de mercado de PL/SQL depuradores y un estudio de DBMS_DEBUG se llevaron a cabo y se estudiaron sus resultados determinado su viabilidad para su implementación.

Antonio Tenorio Fornes, Madrid España año: 2012 Generación de casos de prueba para SQL

Este proyecto lleva a cabo el desarrollo de una herramienta que, dada la definición de una base de datos SQL y una vista objetivo, encuentra una instancia de la base de datos para la cual la consulta de la vista no devuelve una respuesta vacía, lo que se considera un caso de prueba positivo. Aplicando recientes resultados de investigación, transforman la búsqueda de esta instancia a un problema de resolución de restricciones. Ampliando también el ámbito de aplicación de esta técnica presentando un nuevo enfoque para tratar condiciones existenciales anidadas.

**Diego Alconada Gonzales, universidad oberta Catalunya, año: 2013 AlcoNQL
herramienta de consulta SQL por medio de lenguaje natural.**

Este proyecto implementa una herramienta capaz de interpretar sentencias en lenguaje natural y convertirlas en consultas SQL, basado en la interpretación del lenguaje tomándolo como una tarea no muy sencilla. La interpretación de las palabras, con sus distintas variaciones (prefijos, sufijos, plurales, diminutivos, etc.). Unido a los distintos significados que pueden adquirir las mismas.

Una vez identificada la problemática de las palabras profundizan en la estructura de las frases, o la sintaxis. La gramática es muy variada por lo cual identifican qué papel juega cada palabra en una frase, identifican cual es el verbo que da la acción, o el sujeto que la desempeña.

Y por último la semántica. Interpretar el significado de una frase puede ser tarea ardua. Puede haber distintas frases con un mismo significado o Frases que cambien completamente su significado con cambios sutiles en su estructura.

2.4 JUSTIFICACION

Muchas de las herramientas o técnicas empleadas habitualmente en los lenguajes imperativos no resultan aplicables en los lenguajes declarativos, por lo que adoptar estas técnicas a menudo exige el desarrollo de ideas y conceptos nuevos, así como la falta de herramientas auxiliares tales como analizadores sintácticos más detallados con entornos gráficos propios, ha sido señalada como una posible limitación. Este mismo razonamiento es aplicable en general a todos los llamados lenguajes declarativos.

Esto es muy importante para los que manejan el desarrollo de las bases de datos sea más intuitivos, y permitirles la corrección de los errores.

Para los desarrolladores actualmente el creciente uso masivo de la tecnología y de Internet, ha facilitado al mundo el uso y acceso a la web, la utilización de los diferentes servicios como consulta de páginas estáticas, consulta de páginas dinámicas, asesorías online, entre otros; esto hace que la aplicación sea más accesible a los medios y a su manejo en diferentes partes.

La investigación desarrollada en Corus Tecnologías de Estocolmo, Suecia, entre junio y noviembre de 2000 y la investigación en la Generación de casos de prueba para SQL investigado en Madrid, España, Facultad de Informática, Junio 2012 hacen que sea necesario implementar una herramienta que permita verificar sentencias SQL y comprobar errores.

3. MARCO TEORICO Y MARCO CONCEPTUAL

A continuación se aborda el marco teórico y marco conceptual en los cuales se hace una revisión a los conceptos básicos para el entendimiento tales como depuración declarativa, Software (libre), SQL, sentencias sql-dml, analizador sintáctico entre otros. Se darán a conocer teorías, un poco del origen de los conceptos utilizados en este proyecto de grado.

3.1 SOFTWARE LIBRE

El software libre (en inglés "free software", aunque esta denominación a veces se confunde con "gratis" por la ambigüedad del término "free" en el idioma inglés, por lo que también se usa "libre software") es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software

sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público. (concepto s. l.)

De acuerdo con tal definición, un software es "libre" cuando garantiza las siguientes libertades:

Tabla 1 Libertades del software libre.

| Libertad | Descripción |
|--|--|
| 0 | La libertad de usar el programa, con cualquier propósito. |
| 1 | La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades. |
| 2 | La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo. |
| 3 | la libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie |
| Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable. | |

Fuente: Elaboración propia

3.2 LENGUAJE SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones. (SQL)

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros y no a registros individuales, permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

- **Lenguaje de definición de datos (LDD):** El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- **Lenguaje interactivo de manipulación de datos (LMD):** El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- **Integridad:** El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- **Definición de vistas:** El LDD incluye comandos para definir las vistas.
- **Control de transacciones:** SQL tiene comandos para especificar el comienzo y el final de una transacción.
- **SQL incorporado y dinámico:** Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.
- **Autorización:** El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

3.2.1 LENGUAJE DE DEFINICIÓN DE DATOS (DDL).

El lenguaje de definición de datos (DDL) es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Incluye órdenes para modificar, borrar o definir las tablas en las que se almacenan los datos de la base de datos. Existen cuatro operaciones básicas: **CREATE, ALTER, DROP y TRUNCATE.**

3.2.1.1 CREATE | CREAR

Este comando permite crear objetos de datos, como nuevas bases de datos, tablas de vistas y procedimientos almacenados.

3.2.1.2 ALTER | MODIFICAR

Este comando permite modificar la estructura de un objeto. Se pueden agregar/quitar campos a una tabla, modificar el tipo de un campo, agregar/quitar índices a una tabla, modificar un trigger, etc.

3.2.1.3 DROP | ELIMINAR

Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.

3.2.1.4 TRUNCATE | BORRAR TABLA

Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DROP, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande. La desventaja es que TRUNCATE sólo sirve cuando se quiere eliminar absolutamente todos los registros, ya que no se permite la cláusula WHERE. Si bien, en un principio, esta sentencia parecería ser DML (Lenguaje de Manipulación de Datos), es en realidad una DDL, ya que internamente, el comando TRUNCATE borra la tabla y la vuelve a crear y no ejecuta ninguna transacción.

3.2.2 LENGUAJE DE MANIPULACIÓN DE DATOS (DML).

Un lenguaje de manipulación de datos (DML) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relaciona.

3.2.2.1 SELECT

Tabla 2 Sentencia select.

| SINTAXIS SELECT | |
|---|---|
| Concepto | Forma básica |
| <p>La sentencia SELECT nos permite consultar los datos almacenados en una tabla de la base de datos.</p> | <pre> SELECT [ALL DISTINCT] <nombre_campo> [{,<nombre_campo>}] FROM <nombre_tabla> <nombre_vista> [{,<nombre_tabla> <nombre_vista>}] [WHERE <condicion> [{ AND OR <condicion>}]] [GROUP BY <nombre_campo> [{,<nombre_campo >}]] [HAVING <condicion>[{ AND OR <condicion>}]] [ORDER BY <nombre_campo> <indice_campo> [ASC DESC] [{,<nombre_campo> <indice_campo> [ASC DESC }]] </pre> |

Fuente: Elaboración propia

3.2.2.2 INSERT

Tabla 3 Sentencia insert

| SINTAXIS INSERT | |
|--|--|
| Concepto | Forma básica |
| <p>Una sentencia INSERT de SQL agrega uno o más registros a una (y sólo una) tabla en una base de datos relacional.</p> | <pre> INSERT INTO "tabla" ("columna1", ["columna2,..."]) VALUES ("valor1", ["valor2,..."]) </pre> <p>Las cantidades de columnas y valores deben ser las mismas. Si una columna no se especifica, le será asignado el valor por omisión.</p> |

Fuente: Elaboración propia

3.2.2.3 UPDATE

Tabla 4 Sentencia update

| SINTAXIS UPDATE | |
|---|---|
| Concepto | Forma básica |
| Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla. | UPDATE "tabla" SET "columna1" = "valor1" [,"columna2" = valor2",...] WHERE "columnaN" = "valorN" |

Fuente: Elaboración propia

3.2.2.4 DELETE

Tabla 5 Sentencia delete

| SINTAXIS DELETE | |
|---|--|
| Concepto | Forma básica |
| Una sentencia DELETE de SQL borra uno o más registros existentes en una tabla. | DELETE [FROM] "tabla" WHERE "columna1" = "valor1" |

Fuente: Elaboración propia

3.3 ANALIZADOR SINTACTICO

Los analizadores sintácticos fueron extensivamente estudiados durante los años 70 del siglo XX, detectándose numerosos patrones de funcionamiento en ellos, cosa que permitió la creación de programas generadores de analizadores sintácticos a partir de una especificación de la sintaxis del lenguaje en forma Backus-Naur por ejemplo, tales como yacc, GNU bison y javaCC.

Un analizador sintáctico (o parser) es una de las partes de un compilador que transforma su entrada en un árbol de derivación.

El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada. Un analizador léxico crea tokens de una secuencia de caracteres de entrada y son estos tokens los que son procesados por el analizador sintáctico para construir la estructura de datos, por ejemplo un árbol de análisis o árboles de sintaxis abstracta.

El análisis sintáctico también es un estado inicial del análisis de frases de lenguaje natural. Es usado para generar diagramas de lenguajes que usan flexión gramatical, como los idiomas romances o el latín. Los lenguajes habitualmente reconocidos por los analizadores sintácticos son los lenguajes libres de contexto. Cabe notar que existe una justificación formal que establece que los lenguajes libres de contexto son aquellos reconocibles por un autómata de pila, de modo que todo analizador sintáctico que reconozca un lenguaje libre de contexto es equivalente en capacidad computacional a un autómata de pila. (sintactico, s.f.)

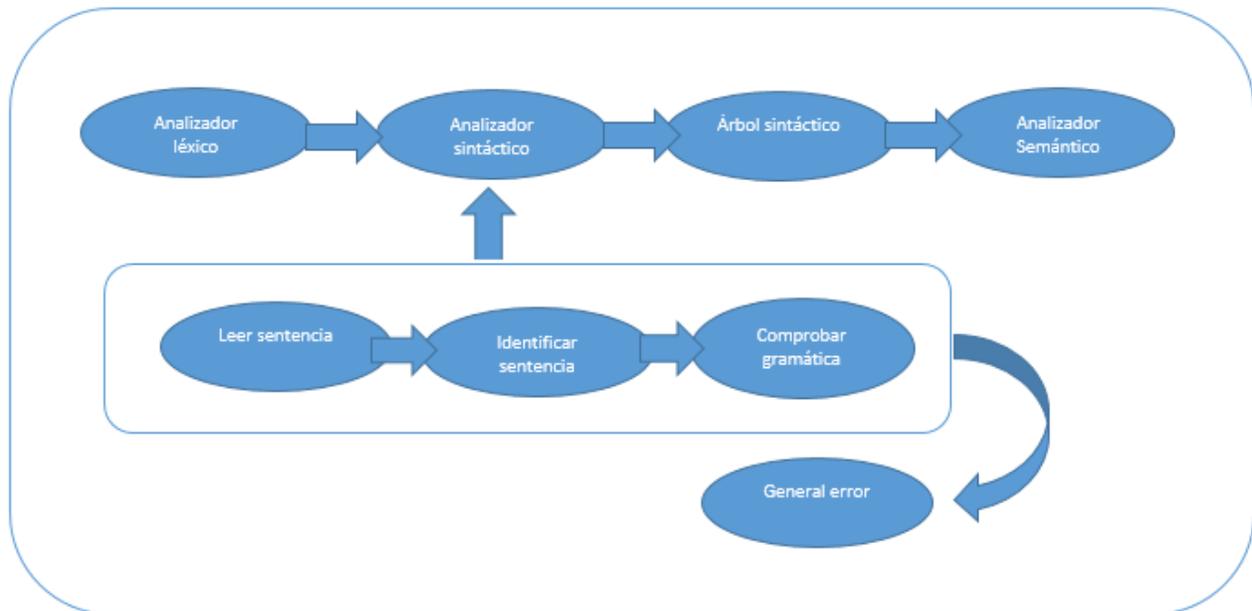


Figura 1 Estructura del analizador sintáctico. (Fuente propia)

3.4 DEPURACIÓN DECLARATIVA

Cuando se ejecuta un programa, con un objetivo dado, se esperan obtener las soluciones esperadas por el programador, quien tiene en mente el significado del programa, es decir, lo que éste debe calcular. La depuración es un proceso mediante el cual los programas detectan y señalan las causas de las discrepancias entre lo que el programa calcula realmente y lo que desea el programador. La búsqueda de estos errores se puede enfocar desde el punto de vista del conocimiento procedural, que tiene en cuenta la secuencia de operaciones aplicadas durante la ejecución, o desde el conocimiento declarativo del programa, el cual determina cuales de los resultados devueltos por los programas son correctos. (programacion_declarativa)

SQL es un lenguaje declarativo, es decir, que especifica qué es lo que se quiere y no cómo conseguirlo, por lo que una sentencia no establece explícitamente un orden de ejecución; de ahí la idea de nuestro proyecto de hacer un depurador para este lenguaje.

3.5 APPLETS JAVA.

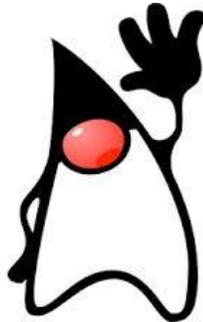


Figura 2 Applets java. Oracle, Duke, la mascota de Java, 1992. Recuperado de <http://www.oracle.com/us/sun/index.htm>

Los applets son aplicaciones Java a las que se accede en un servidor Internet, se transmiten a través de la red, se instalan automáticamente y se ejecutan como parte de una página web. Su gran ventaja es la portabilidad. Por tanto, los applets sólo pueden ser ejecutados por un navegador compatible con Java o por un programa visualizador de applets (como el programa *applet viewer*).

Para incluir un applet en una página HTML, en esta página hay que utilizar la etiqueta o tag **<APPLET>** y para el desarrollo del applets, Java maneja la clase **Applet**, que proporciona los métodos para su manejo. (Applet, s.f.)

3.5.1 ARQUITECTURA DE UN APPLET.

Los applets son programas basados en ventanas, y así su arquitectura es diferente a la de los programas que interactúan con una consola guiados por eventos. Así, un applet está esperando hasta que se produce un evento. Cuando este se produce, el applet es informado de ello, y efectúa la acción correspondiente a ese evento como son abrir un fichero, presentar una imagen, realizar un cálculo, etc.

Por tanto, es el usuario el que inicia la interacción con un applet y no al contrario. Estas interacciones se envían al programa como eventos que éste debe tratar.

3.5.2 ESTRUCTURA DE UN APPLET.

Hay cinco métodos que proporcionan el mecanismo que permite controlar la ejecución de un applet. Todos estos métodos tienen implementaciones por defecto, así que sólo han de sobrescribirse si se van a utilizar. Los métodos son:

- **Init():** es el primer método que se ejecuta, y sólo se ejecuta cuando se carga el applet. Este método se debe utilizar para realizar las acciones iniciales del applet: inicializar variables, distribuir los componentes gráficos del applet, etc.
- **Start():** se ejecuta después de `init()`. También se ejecuta cada vez que el applet tiene que volver a ejecutarse por ejemplo, cuando el usuario sale de la página web y después vuelve a la misma; o cuando se minimiza la ventana del navegador y se ejecuta otra tarea, y se vuelve a ejecutar el applet.
- **Paint():** este método se ejecuta cada vez que hay que pintar la pantalla del applet (por ejemplo, porque la ventana de la página web ha sido tapada por otra ventana, o porque se ha minimizado o restaurado su tamaño). Por eso en este método se deben introducir las acciones para volver a presentar los datos gráficos en el área de visualización del applet.
- **Stop():** este método se ejecuta cuando el navegador deja el documento HTML que contiene el applet (porque viaja a otra página). Se debe utilizar este método para suspender los hilos que no es necesario que se ejecuten mientras el applet no es visible. Estos hilos se pueden reiniciar cuando se ejecuta el método `start()` al volver el usuario a la página que contiene el applet.

- **Destroy():** se ejecuta cuando el applet va a ser borrado totalmente de memoria; entonces hay que liberar todos los recursos que el applet esté utilizando. El método stop () siempre se ejecuta antes que destroy ().

3.6 APLICACIONES WEB

Las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes como HTML. En ellas se utilizan lenguajes interpretados del lado del cliente como el JavaScript. Para así añadir elementos dinámicos a la interfaz del usuario. Ahora bien las interfaces web tienen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar y soltar no están soportadas por las tecnologías web estándar. (web A. , s.f.)

Los desarrolladores web comúnmente utilizan lenguajes interpretados del lado del cliente para añadir más funcionalidad, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez. Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías del lado del servidor.

Otra aproximación es utilizar los programas java applets o Macromedia flash para producir parte o toda la interfaz de usuario. Como casi todos los browsers incluyen soporte para estas tecnologías, usualmente por medio de plug-in, aplicaciones basadas en java o flash pueden ser implementados con aproximadamente la misma facilidad. En cuanto a las herramientas de software a utilizar, hoy en día están adquiriendo especial relevancia las herramientas de código libre. Las ventajas de utilizar servidores con código abierto como apache son indudables y junto a java se están convirtiendo en programas muy utilizados en estos entornos, por sus presentaciones y rendimientos equiparables o superiores a las de productos comerciales, y lo mejor de todo es que se reducen los costos de licenciamiento ya que no requieren pago por la utilización de estos.

3.6.1 ARQUITECTURA DE APLICACIONES WEB

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios.

Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. Requiere una interfaz de usuario que se instala y corre en una pc o estación de trabajo; y que envía solicitudes a un servidor para ejecutar operaciones complejas.

Al hablar del desarrollo de aplicaciones web resulta adecuado presentarla dentro de las aplicaciones multinivel. La arquitectura de las aplicaciones web suelen presentar un esquema de tres niveles así:

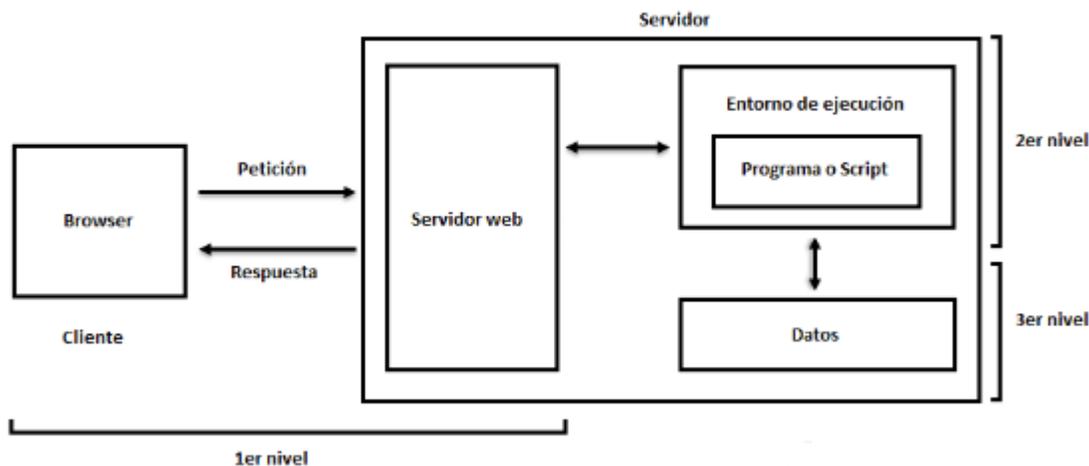


Figura 3. Esquema Niveles de aplicaciones web MVC (Fuente propia)

El primer nivel consiste en la capa de presentación que incluye no solo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente aun tipo de programa o script.

Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación web típica recogerá datos del usuario (primer nivel), los enviara al servidor, que ejecutara un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

3.7 SERVIDORES WEB

Es un programa que implementa el protocolo HTTP, este está diseñado para transferir lo que llamamos hipertextos, páginas web o paginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no se debe confundir los términos.HTML es un formato de archivo y HTTP es un protocolo.

Un servidor web se encarga de mantenerse a la espera de las peticiones HTTP llevaba a cabo por un cliente HTTP que es llamado navegador. Al El navegador realiza la petición al servidor y este responde con el contenido que él solicita. (web S. , s.f.)

3.7.1 SERVIDOR WEB APACHE

Apache es un servidor web código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool, es una tentativa de mejorar el servidor existente en la NCSA. La primera versión apareció en enero de 1996, el apache 1.0 hacia el 2000, el servidor web apache era el más extendido en el mundo. El nombre “apache” es un acrónimo de “a patchy server “un servidor de remiendos, es decir, un servidor construido con código preexistente, piezas y parches de código. Es la auténtica “kill App” del software libre de mayor éxito, por delante incluso del kernel de Linux. Desde hace años más del 60% de los servidores web de internet emplean apache. (apache, s.f.)

3.8.1.1 CARACTERÍSTICAS DEL SERVIDOR APACHE

- Nuevos módulos apache API: se utiliza un nuevo conjunto de interfaces de aplicación (APIs).
- Filtrado: los módulos pueden actuar como filtros de contenidos.
- Soporte para IPv6: se tiene soporte para la nueva generación de las direcciones IP.
- Directivas simplificadas: se han eliminado una serie de directivas complicadas y otras se han simplificado.
- Respuestas a errores en diversos idiomas: cuando usas documentos Server Side include (SSI), las páginas de errores personalizables se pueden entregar en diversos idiomas.
- Soporte a múltiples protocolos.

3.8 WEB SERVICE

Un servicio web (en inglés, *Web Service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

La principal razón para usar servicios Web es que se basan en HTTP sobre TCP (Transmission Control Protocol) en el puerto 80. Dado que las organizaciones protegen sus redes mediante *firewalls* que filtran y bloquean gran parte del tráfico de Internet, cierran casi todos los puertos TCP salvo el 80, que es, precisamente, el que usan los navegadores. Los servicios Web utilizan este puerto, por la simple razón de que no resultan bloqueados. (Service, s.f.)

3.8.1 SERVIDORES DE APLICACIONES PARA SERVICIOS WEB:

- JBoss servidor de aplicaciones J2EE Open Source de Red Hat inc.
- Oracle Fusion Middleware
- IBM Lotus Domino a partir de la versión 7.0
- Axis y el servidor Jakarta Tomcat (de Apache)
- ColdFusion MX de Macromedia
- Java Web Services Development Pack (JWSDP) de Sun Microsystems (basado en Jakarta Tomcat)
- JOnAS (parte de *ObjectWeb* una iniciativa de código abierto)
- Microsoft .NET
- Novell exteNd (basado en la plataforma J2EE)
- WebLogic
- webSphere
- JAX-WS con GlassFish
- Zope es un servidor de aplicación web orientado a objetos desarrollado en el lenguaje de programación Python
- VERASTREAM de AttachmateWRQ para modernizar o integrar aplicación host IBM y VT

3.9 AUTÓMATAS FINITOS

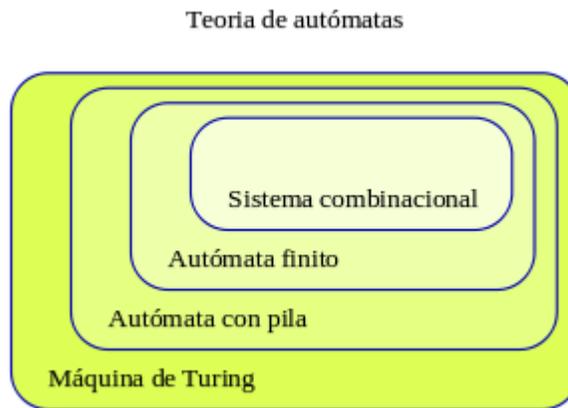


Figura 4 Autómata finito (Fuente propia)

Un autómata finito (AF) o máquina de estado finito es un modelo computacional que realiza cálculos en forma automática sobre una entrada para producir una salida.

Este modelo está conformado por un alfabeto, un conjunto de estados y un conjunto de transiciones entre dichos estados. Su funcionamiento se basa en una función de transición, que recibe a partir de un estado inicial una cadena de caracteres pertenecientes al alfabeto (la entrada), y que va leyendo dicha cadena a medida que el autómata se desplaza de un estado a otro, para finalmente detenerse en un estado final o de aceptación, que representa la salida. (Automata_finito, s.f.)

3.9.1 FUNCIONAMIENTO

En el comienzo del proceso de reconocimiento de una cadena de entrada, el autómata finito se encuentra en el estado inicial y a medida que procesa cada símbolo de la cadena va cambiando de estado de acuerdo a lo determinado por la función de transición. Cuando se ha procesado el último de los símbolos de la cadena de entrada, el autómata se detiene en el estado final del proceso. Si el estado final en el que se detuvo es un estado de aceptación, entonces la cadena pertenece al lenguaje reconocido por el autómata; en caso contrario, la cadena no pertenece a dicho lenguaje.

3.10 AUTÓMATA DE PILA

Para visualizar un autómata de pila podemos imaginar los estados y la cinta de entrada como en los autómatas finitos, pero ahora está la pila que podemos imaginar como una cinta interna (que siempre representamos como una columna) donde se van insertando o extrayendo los símbolos de pila según lo vayan mandando las transiciones.

La pila hace el papel de una memoria rudimentaria: sobre ella se escriben palabras y se van extrayendo símbolo a símbolo. Debe quedar claro el modo en que entendemos que se insertan las palabras en la pila: Si $\omega = a_1 \dots a_k$ es una palabra de longitud k y queremos insertarla en la pila de un AP, entendemos que el símbolo que queda en la cima de la pila es a_1 . (Automata de pila, s.f.)

3.11 BBDD: MySQL.



Figura 5: Icono MySQL. Oracle, Sakila mascota de MySQL .Recuperado de <http://www.mysql.com>

MySQL es principalmente que está desarrollado bajo la filosofía de código abierto y que es multiplataforma, aunque posee grandes características. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Es el sistema de gestión más rápido, los desarrolladores sostienen que MySQL es posiblemente la base de datos más rápida que pueden encontrar. En cuanto a facilidad de uso es un sistema de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes, además comprende SQL (structuredquery lenguaje, lenguaje de consulta estructurado), que es el lenguaje utilizado para todos los sistemas de bases de datos modernos.

Con respecto a capacidad pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias bases de datos simultáneamente. Pueden acceder de forma interactiva a MySQL empleando diferentes interfaces que le permitan introducir consultas y visualizar los resultados: cliente de línea de comando, navegadores Web o clientes del sistema por Windows. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java, PHP Y PITÓN. Por tanto, tiene la posibilidad de elegir entre usaron software cliente-pre empaquetado o escribir sus propias aplicaciones a medida. Por otro lado, MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar en Internet, por lo que puede compartir sus datos con

cualquiera, en cualquier parte, pero MySQL dispone de control de acceso, un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el Host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. (MYSQL)

3.12 LENGUAJE DE PROGRAMACIÓN: JAVA.



Figura 6: Icono java. Oracle, Java tm, 2014. Recuperado de <https://www.java.com>

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que JavaScript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

Java le permite jugar, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como, por ejemplo, cursos en línea, servicios bancarios en línea y mapas interactivos. Si no dispone de Java, muchas aplicaciones y sitios web no funcionarán.

Por defecto, Java le notificará inmediatamente que hay nuevas actualizaciones listas para instalarse. Si desea estar al día y mantener la seguridad de su computadora, es importante que acepte e instale las actualizaciones. Si recibe una notificación de

actualización de Java en su computadora Windows y no recuerda haberla descargado o instalado, lo más probable es que Java estuviera ya instalado en la nueva computadora.

La tecnología java, una tecnología madura extremadamente eficaz y sorprendentemente versátil se ha convertido en un recurso inestimable ya que permite a los desarrolladores.

- Desarrollar software en una plataforma y ejecutarlos en prácticamente cualquier otra plataforma.
- Crear programas para que funcionen en un navegador web y servicios web.
- Desarrollar aplicaciones para servidores como foros en líneas, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios basados en tecnología java para crear servicios o aplicaciones totalmente personalizados.
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo coste y prácticamente cualquier dispositivo digital.

(java)

Capa de software que implementa la máquina virtual de java

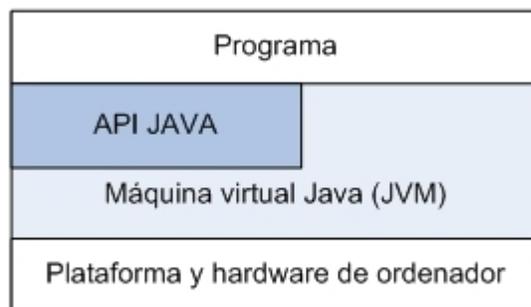


Tabla 6 Capa de software de la JVM. (Fuente propia)

3.13 PHP



Figura 7: Icono PHP. Rasmus Lerdorf, Hypertext Preprocessor, 1995. Recuperado de <http://php.net/>

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor, pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas GTK+.

PHP significa HypertextPre-processor. Inicialmente Fue creado originalmente por RasmusLerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre. (PHP, s.f.)

Características de PHP

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

3.14 HTML

HTML5.0

*Figura 8: Icono HTML. Berners-Lee, lenguaje de marcas de hipertexto, 2014.
Recuperado de <http://www.w3.org/>*

Las siglas de HyperTextMarkupLanguage (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo Java script), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. (HTML, s.f.)

VENTAJAS

- HTML puede ser escrito en cualquier editor de texto, aún en el más básico.
- Soporte Java Script
- Sintaxis Resaltada
- Soluciones Rápidas (Quick Fixes) y Verificación de Sintaxis Refactorización.
- Mejoras en el Desempeño
- Inicio hasta 40% más rápido
- Menor consumo.

3.15 DREAMWEAVER



Figura 9 Icono Dreamweaver. Adobe Systems, programa de estudio, 2014. Recuperado de <http://www.adobe.com/la/creativecloud.html>

Creador y editor de páginas web tipo WYSIWYG (What-You-See-Is-What-You-Get) creado por Macromedia (actualmente de adobe System).

Es la aplicación de este tipo más usada en el sector de diseño y programación web. Posee, como toda la línea Macromedia/Adobe, excelentes funcionalidades e integración Con otras herramientas.

La aplicación permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado. Permite la conexión a un servidor, a base de datos, soporte para programación en ASP, PHP, Java Script, cliente FTP integrado, etc. (Dreamweaver, s.f.)

3.16 NETBEANS



Figura 10 Icono Netbeans. Oracle, entorno de desarrollo, 2000. Recuperado de <https://netbeans.org>

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software. (Dreamweaver, s.f.)

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

4. METODOLOGIA

4.1 FASES DEL PROYECTO

La ejecución de la aplicación web orientada al análisis sintáctico de sentencias SQL-DML en base de datos mysql. (SQLparser) se desarrolló en distintas fases objetivos:

FASE 1

Identificar los requerimientos para el manejo del lenguaje estándar SQL-DML.

Se realizó historial de la investigación a través de los siguientes pasos como fueron:

- Búsquedas en internet de artículos asociados con el tema, libros, tesis, investigaciones.
- Experiencias, observaciones de clases de bases de datos en el tema de las sentencias SQL.
- Una amplia investigación en videos tutoriales, documentos en general sobre analizadores sintácticos, sentencias SQL y las bases de datos.

Pero partiendo de que es un lenguaje estándar ya implementado y de que estoy trabajando para base de datos MySQL he de utilizar el manual de referencia original de MySQL <http://dev.mysql.com/doc/refman/5.0/es/sql-syntax.html> en donde me enfocare en las sentencias de manipulación de datos DML (select, delete, update, insert), ya que es el tema central de mi proyecto.

FASE 2

Desarrollar una aplicación web dedicada al análisis sintáctico del lenguaje SQL.

En esta fase del proyecto se dividió en 3 partes:

- ✓ Programación de un algoritmo capaz de analizar las sentencias a partir de su estructura estándar a través de un modelo computacional.
- ✓ Adecuación de la aplicación para la web.
- ✓ Desarrollar la aplicación web.
- ✓ Desarrollo de un web service.

1) Programación de un algoritmo capaz de analizar las sentencias a partir de su estructura estándar a través de un modelo computacional:

Para la creación del algoritmo se partió del modelo de autómatas finito⁵ este modelo se resume en que a partir de unos estados se toma unas decisiones a medida que va recorriendo el árbol o la sentencia; estas decisiones son determinadas a través de la estructura estándar de MySQL. Para su programación se trabajó en el entorno de desarrollo Netbeans⁶ con una programación orientada a objetos.

Hay que destacar que el objetivo es un analizador sintáctico lo cual me resume en transformar su entrada en un árbol de derivación y analizarla en búsqueda de algún error.

2) Adecuación de la aplicación para la web:

Para adecuar la aplicación en la web se utilizó Applet⁷; esta me permite moldear la interfaz de usuario además de integrar a ella las clases java que se creó para el analizador de las sentencias SQL.

Debido a su estructura y funcionamiento el Applet se adecua a la página web

⁵ Autómatas finitos: ver en marco teórico.

⁶ Netbeans: ver en marco teórico.

⁷ Applet: ver en marco teórico.

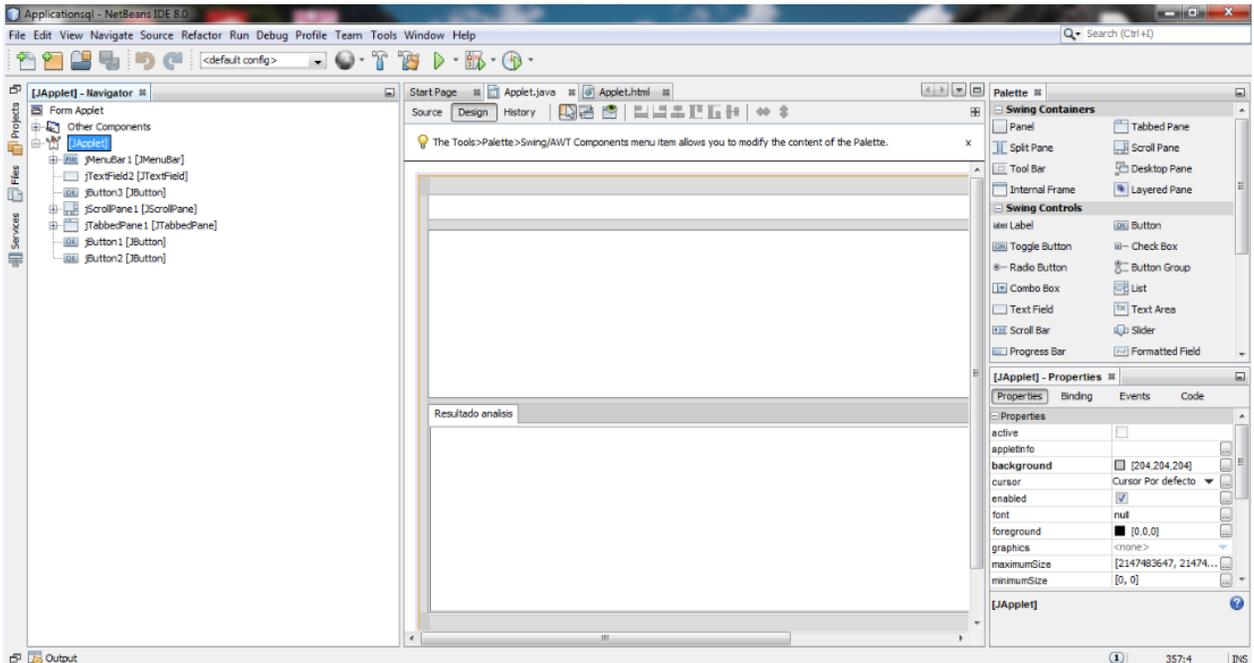


Figura 11 Netbeans Applet (Fuente propia)

2) Desarrollar la aplicación web.

Para el desarrollo de la aplicación web se partió del hecho de que la población objetivo era los estudiantes como razón social.

Como es dirigido a ellos se planteó un diseño web portal que constituye en la creación de cuentas de usuarios y administrador, módulo de perfil y módulo de la aplicación java donde vendría el Applet. Para esto se empleó el editor de páginas web Dreamweaver⁸, archivos PHP, HTML JavaScript, entre otras.

Cada módulo especifica una función en particular (ver manual de usuario) y para su implementación web (ver manual de instalación)

3) Desarrollo de un web service

Para el desarrollo del web service⁹ se partió del uso de las clases java ya

⁸ Dreamweaver: ver en marco teórico.

⁹ Web service: ver en marco teórico.

implementadas para el analizador sintáctico.

Al igual que el Applet esta se programó en java con el IDE Netbeans y se partiendo de la creación de un web service /servicio con su forma abierta para peticiones del cliente. Esta al mandar una petición del que debe constar de la sentencia SQL, devolverá la respuesta del servicio que contendría la cadena resultante del método de la clase.

Method parameter(s)

| Type | Value |
|------------------|----------------------|
| java.lang.String | select * from tablas |

Method returned

java.lang.String : "Sentencia valida "

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:parser xmlns:ns2="http://servicioweb/">
      <sql>select * from tablas</sql>
    </ns2:parser>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:parserResponse xmlns:ns2="http://servicioweb/">
      <return>Sentencia valida
    </return>
  </ns2:parserResponse>
</S:Body>
</S:Envelope>
```

Figura 12 Web service (Fuente propia)

FASE 3

Implementar pruebas de consulta por el usuario con sentencias SQL para

verificación de la aplicación.

Para la implementación de pruebas se planteó una serie de procedimientos para la misma:

PROCEDIMIENTO DE LA PRUEBA

El procedimiento consta de las siguientes actividades: en un primer momento se dará a conocer de unas encuestas a los estudiantes donde se dará a conocer su nivel de conocimiento del tema SQL y otra al final donde me permitirá medir el nivel de satisfacción de la aplicación. En un segundo momento se aplicara otro tipo de prueba tipo ejercicio donde se pondrá a prueba el conocimiento del estudiante. Como tercer momento se procederá al manejo de la aplicación. (Ver evidencias de las encuestas, pruebas, manipulación de la aplicación en anexo)

Todo esto con el objetivo de recolectar información para el desarrollo, análisis y resultado de la prueba.

DESARROLLO DE LA PRUEBA

Durante la realización de la encuesta se tuvo en cuenta a 20 encuestados de los cuales todos fueron estudiantes de desarrollo web/ base de datos dado que la población en cuestión son los ideales para el manejo de la aplicación.

La encuesta se compone de 8 preguntas que arrojaron los siguientes resultados.

1) ¿Conoce usted que es SQL?



Figura 13 Encuesta pregunta # 1 (Fuente propia)

2) ¿Conoce usted que es SQL- DML?

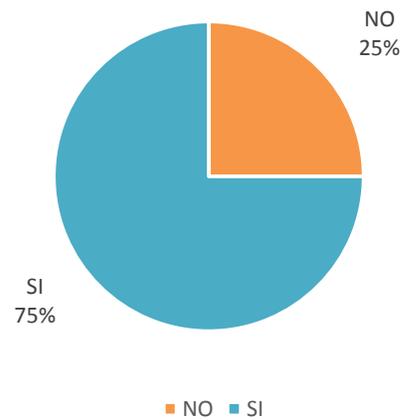


Figura 14 Encuesta pregunta # 2 (Fuente propia)

3) ¿Tiene algún conocimiento acerca de bases de datos?

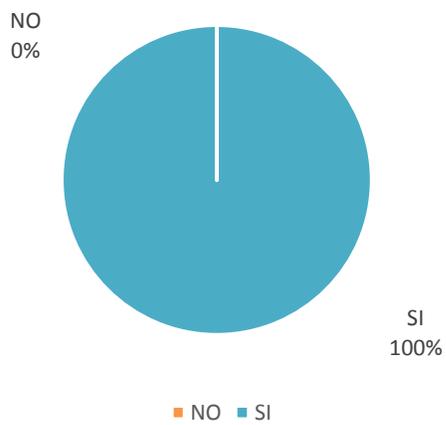


Figura 15 Encuesta pregunta # 3 (Fuente propia)

4) ¿Conoce algún gestor de bases de datos?

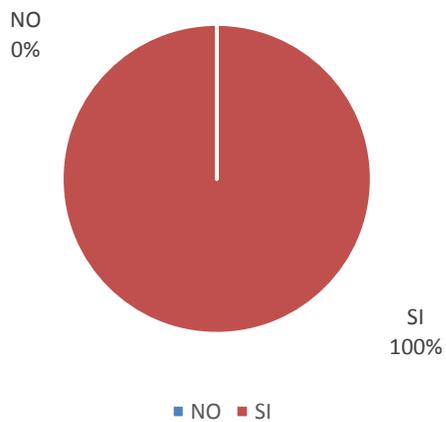


Figura 16 Encuesta pregunta # 4 (Fuente propia)

5) ¿Ha realizado alguna consulta en el gestor de bases de datos?

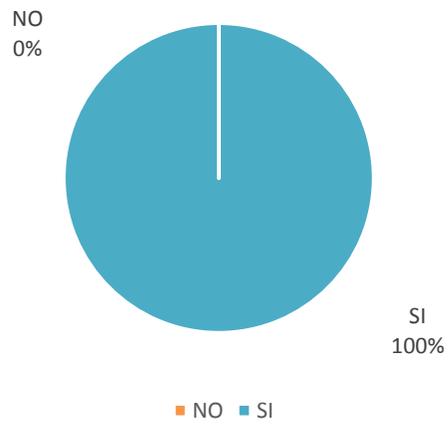


Figura 17 Encuesta pregunta # 5 (Fuente propia)

6) ¿Alguna vez ha obtenido un error sintáctico realizando la consulta?

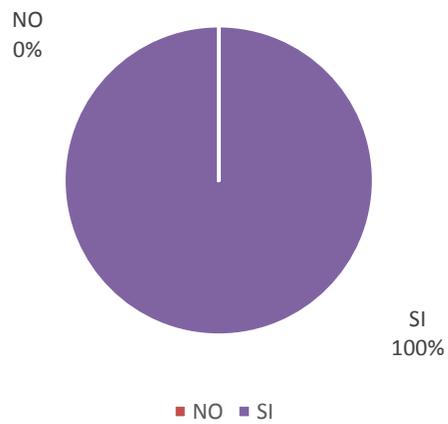


Figura 18 Encuesta pregunta # 6 (Fuente propia)

7) ¿El gestor de bases de datos le da una sugerencia para corregir el error?

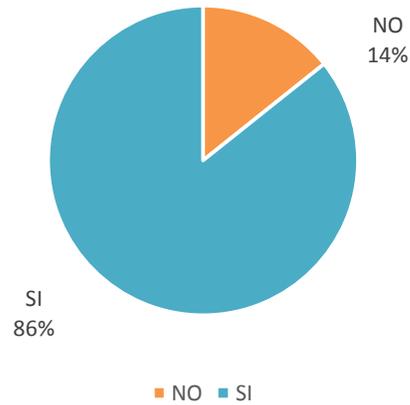


Figura 19 Encuesta pregunta # 7 (Fuente propia)

8) Si no tuviera un conocimiento completo de cómo escribir una sentencia SQL, le gustaría tener una aplicación que le sugiriera como debería escribirla

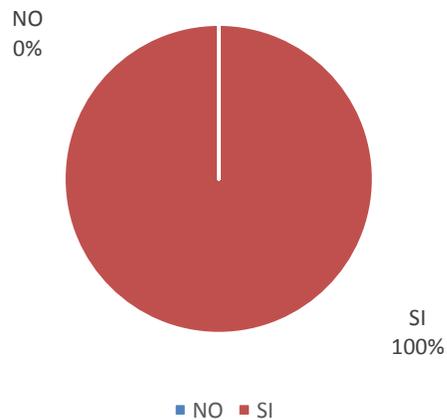


Figura 20 Encuesta pregunta # 8 (Fuente propia)

En la Prueba para aplicación se tomó una muestra de la población de 10 estudiantes partiendo de que alguno no tenían conocimientos fijos o no se acordaban. En la prueba se tuvo en cuenta lo siguiente:

- ✓ Introducción
- ✓ Ejercicios

Introducción:

SQL (Structured Query Language, Lenguaje Estructurado de Consultas): Lenguaje que permite expresar operaciones diversas (aritméticas, combinatorias, lógicas, selección y ordenación) con datos almacenados en bases de datos relacionales.

Las peticiones para actuar sobre los datos se expresan mediante sentencias. Éstas deben escribirse de acuerdo con las reglas sintácticas y semánticas del lenguaje.

Por motivos de la tesis el tipo de operación SQL a expresar serán DML: Lenguaje de Manipulación de Datos (Data Manipulation Language) sentencias que permiten realizar consultas y actualizaciones de datos (Inserción, borrado y modificación de filas): SELECT, INSERT, UPDATE, DELETE.

Ejercicios:

A continuación se mostraran 5 ejercicios básicos los cuales son peticiones que se deben expresar en sentencias:

- 1) Obtener los nombres de todos los trabadores de la empresa x
- 2) Obtener los nombres de los empleados que trabajan en el departamento 121
- 3) Añadir un nuevo producto de acuerdo con los siguientes datos: altavoces cuestan \$20.000, su código es 01 y de fabricante 2. (Código-nombre-precio-fabricante)

- 4) Eliminar a todos los empleados que trabajen para departamentos cuyos presupuestos sean superiores a los \$ 60.000
- 5) Reasignar a los empleados del departamento de investigación (código 77) al departamento de informática (código14)

A continuación se arrojaron los siguientes resultados de los ejercicios por estudiantes:

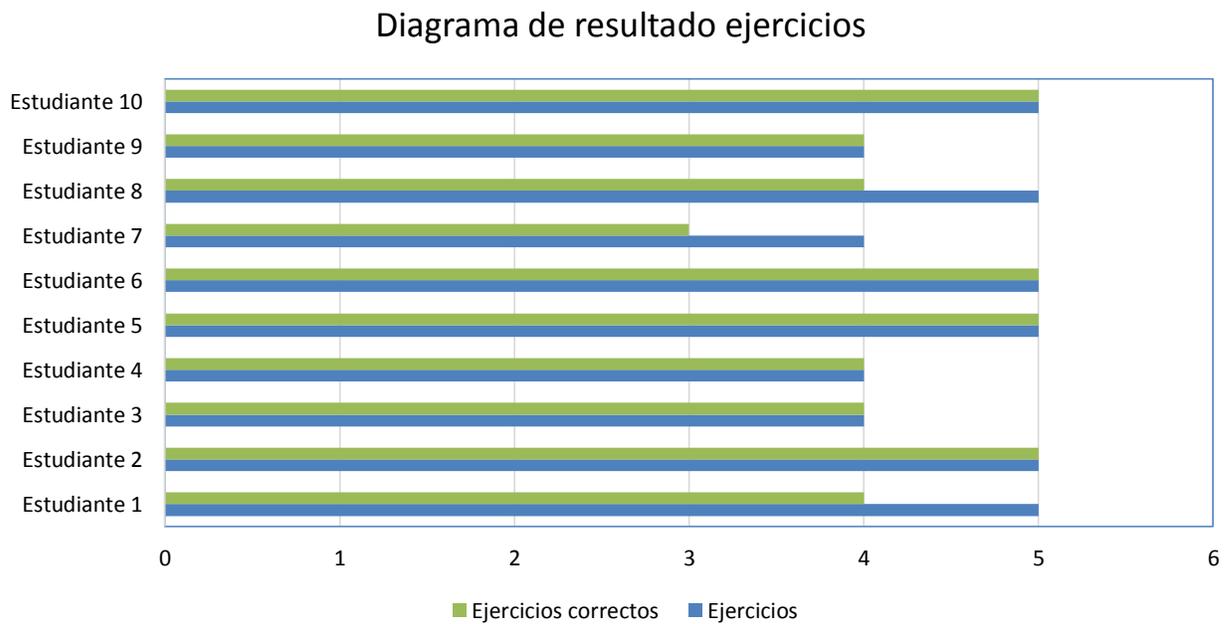


Figura 21 Diagrama de resultados ejercicios (Fuente propia)

Posteriormente para finalizar la prueba se compararon los resultados y se fue guiando a los estudiantes en grupo para probar la aplicación web.

En este caso los estudiantes debieron manejar la aplicación y registrarse



Reporte usuarios

Aquí está el listado de usuarios que están registrados en el sistema tenga en cuenta que los usuarios que no hallan usado la aplicación no tendrán detalles además si elimina un usuario se eliminará de manera permanente el registro.

Grupos: Todos ▾

| Grupo | ID | Nombre | Apellido | Detalle | Eliminar |
|-------|----|----------|---------------|---------|----------|
| 1 | 1 | juan | esteban | | |
| 1 | 14 | maria | angelica | | |
| 1 | 15 | filenis | parra | | |
| 2 | 13 | victor | Escobar safar | | |
| 2 | 16 | pablo | esteban | | |
| 2 | 17 | jaime | ballesta | | |
| 2 | 18 | JONATHAN | PATERNINA | | |
| 2 | 19 | mayerly | rios | | |
| 2 | 20 | Jesus | Valderrama | | |
| 2 | 21 | mercedes | monterrosa | | |
| 2 | 22 | CARLOS | HENAZO | | |
| 2 | 23 | adela | barrios | | |
| 2 | 24 | manuel | fernando | | |
| 2 | 25 | leydis | vega | | |
| 2 | 26 | Cristian | Camilo | | |

Figura 22 Modulo reporte de usuario (Fuente propia)

Como se puede apreciar entrando como administrador se ve en detalle los estudiantes que probaron la aplicación

Ahora bien los ejercicios constaron de sentencias definidas de la siguiente manera

- 2 Select
- 1 Insert
- 1 Update
- 1 Delete

Para un total de 5 que corresponden al número de ejercicios.

La guía correspondió a ir probando la aplicación paralela a los ejercicios, así velar por el correcto funcionamiento de la aplicación y debido al tiempo que se me suministro ir trabajando con los estudiantes.

Como vemos al ir ingresando una estructura de sentencia la aplicación arroja sugerencias

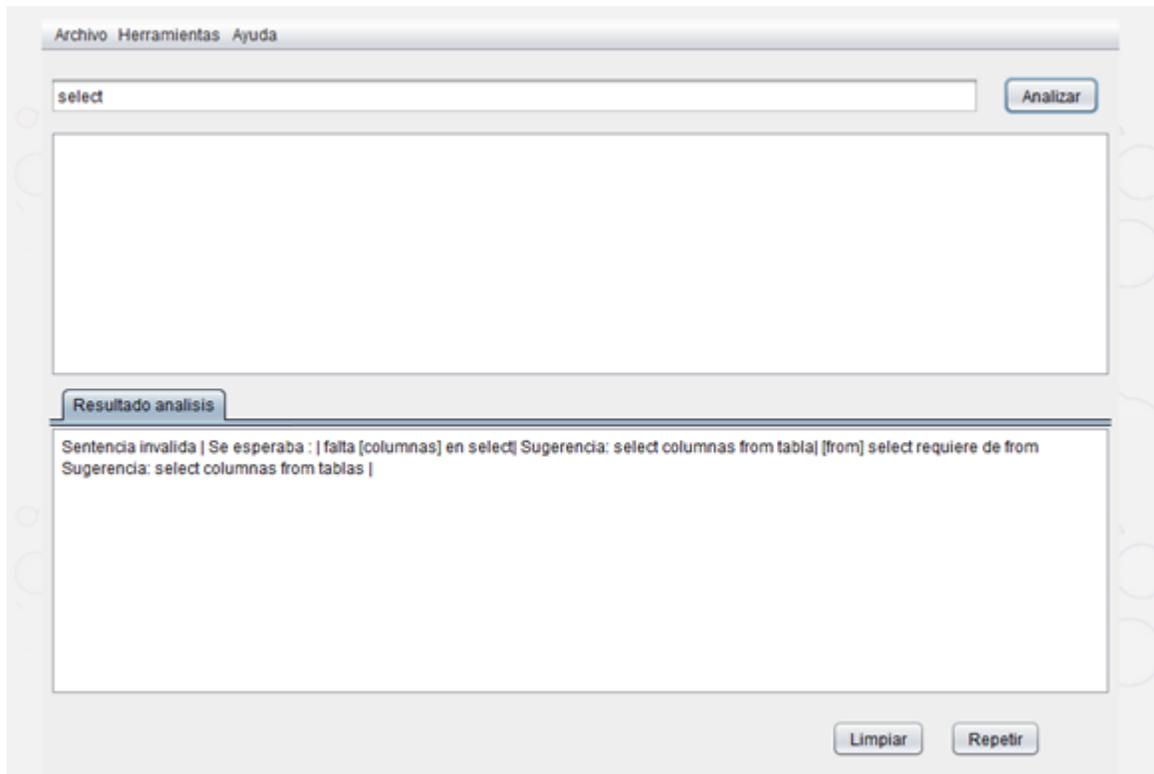


Figura 23 Aplicación Applet analizando (Fuente propia)

4.2 METODOLOGIA DE DESARROLLO DEL PRODUCTO

MODELO DE DESARROLLO AGIL

Para mi proyecto en particular elegí la metodología ágil XP (Programación Extrema). Ya que es la que más se acerca a la distribución de mi proyecto, por la necesidad de cambio constante sin afectar la funcionalidad. Esta metodología es más extendidas y conocidas además están cobrando especial importancia en el desarrollo Web ya que las aplicaciones suelen ser fácilmente divisibles en pequeñas partes sencillas.

Este proceso está basada en la simplicidad, la comunicación, la retroalimentación y la refactorización de código, para así extraer los requisitos y la identificación de las partes funcionales en las que se divide la solución a medida que se va avanzando en el

proyecto. La arquitectura del proceso se modela con orientación a objetos se trabaja en cuatro etapas o fases:

Planificación

Se evaluaron los requisitos para el desarrollo de la aplicación teniendo en cuenta los objetivos propuestos y se planteó las herramientas, tecnologías y pruebas que se utilizarán en el proyecto; tales como Netbeans IDE como herramienta de programación, del que se usaron framework que me ayudarían al desarrollo ágil de la aplicación; tecnologías java ,Applet y demás.

Diseño

Se especificó diseños simples y específicos, se realizó parte del diseño de la aplicación web (se utilizó Dreamweaver con sus respectivos clases de estilo), la parte de la aplicación java fue ágil en el sentido del Applet ya que IDE me facilitaron su manejo con sus herramientas de interfaz de usuario, así en este punto se pueda conseguir un diseño fácilmente entendible e implementable.

Codificación

Durante esta fase la aplicación se mueve en torno a los requerimientos en este caso el cliente (asesor) me incluye nuevas funcionalidades específicas para el mejoramiento del sistema, la programación fue orientado a objetos del que se implementaron 2 clases java mutuamente instanciadas, que proponen diferentes métodos: 1 de ellos es la clase Applet que maneja métodos imperativos de él y métodos incluyentes míos, la otra se comprende de código ligado a al análisis de sentencias SQL.

La parte de la base de datos se realizó con respecto a los casos de uso, el manejo de las peticiones y consultas se hicieron con PHP, JSP y JavaScript y se utilizó base de

datos MySQL con servidor apache(es necesario mencionar que este es un caso aparte ya que el proyecto comprende de la página como tal y de la aplicación java a integrar)

Cabe destacar que las iteraciones en búsqueda de los errores o actualizaciones son constantes eh inmediatos, pero sin alterar el funcionamientos de la misma.

Prueba

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

Para esta fase lo que se implementa a las pruebas se le va haciendo al proyecto, cada prueba se realizada con respecto funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

5. DESARROLLO

5.1 ARQUITECTURA DEL SISTEMA

Este modelo fue basado en la arquitectura modelo vista controlador (MVC).

5.1.1 MODELO MVC

El modelo (MVC) o model-view-controller surgió como patrón arquitectónico para el desarrollo de interfaces graficas de usuarios en entornos smaltlak. Su concepto se basaba en separar el modelo de datos de la aplicación de su representación de cara al

usuario y de la interacción de este con la aplicación mediante la división de esta en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que este necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre el modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

En la siguiente figura se muestra la arquitectura del sistema cliente-servidor con sus respectivas 3 capas (aplicación, negocio, datos) del modelo MVC:



Figura 24 Arquitectura del sistema (Fuente propia)

5.2 DISEÑO DEL SISTEMA

5.2.1 MODELO ENTIDAD RELACIÓN

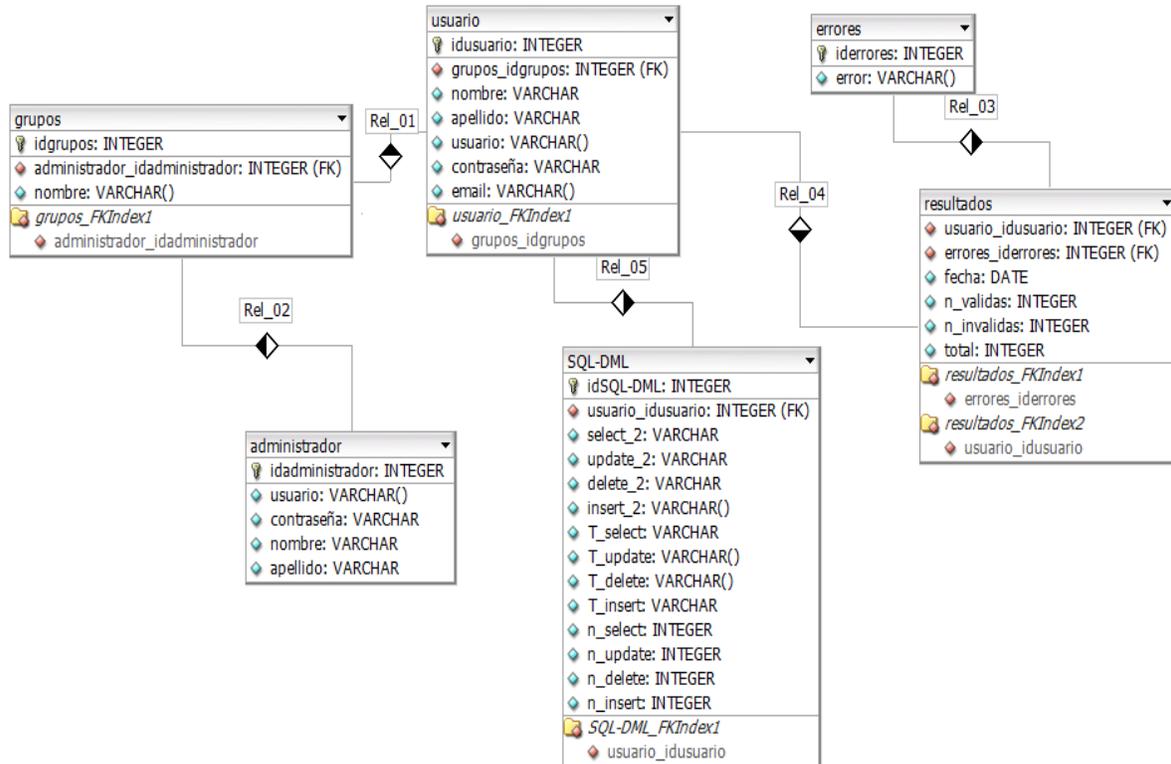


Figura 25 Modelo entidad relación (Fuente propia)

5.2.2 DIAGRAMA DE CLASE

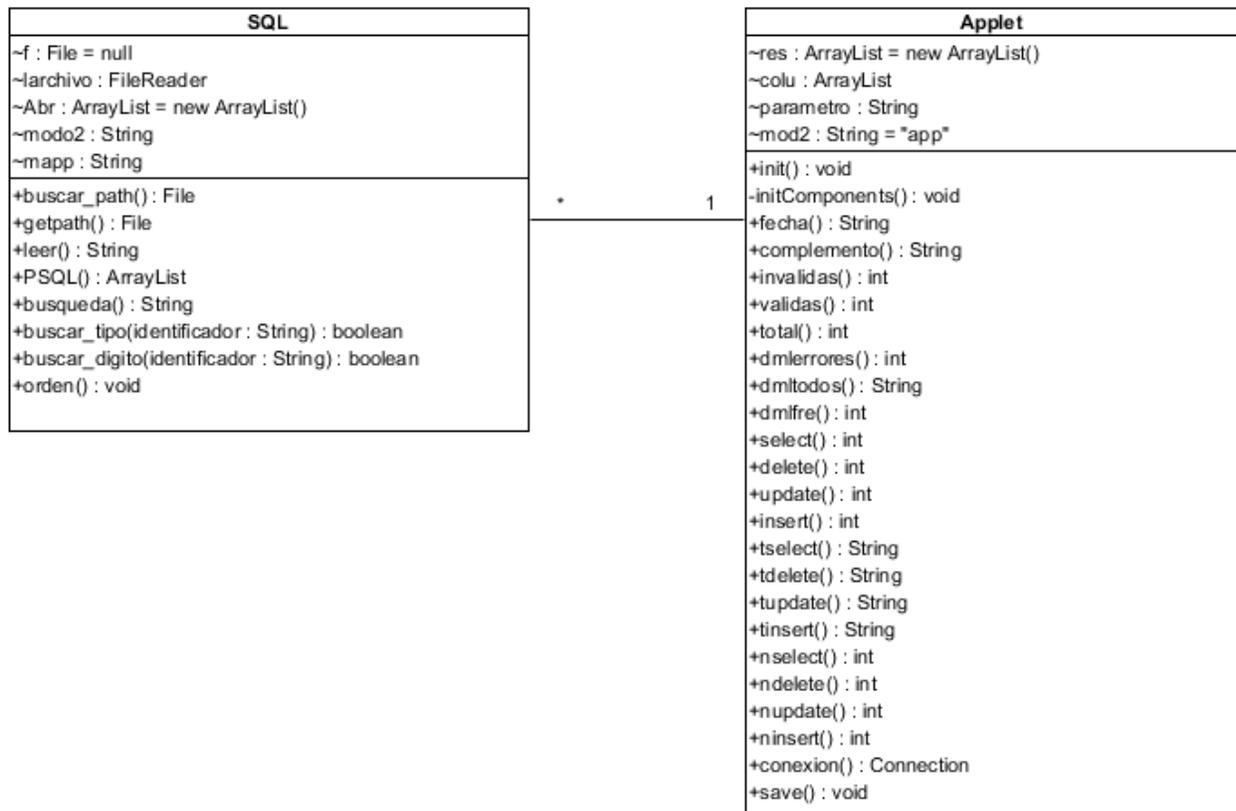


Figura 26 Diagrama de clase (Fuente propia)

5.2.3 DEFINICIÓN DE USUARIOS DE SISTEMA.

5.2.3.1 USUARIOS DEL SISTEMA

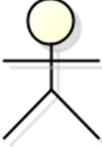
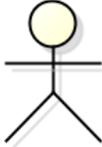
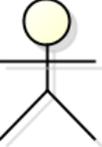
| USUARIO | DESCRIPCION |
|---|---|
|  Administrador | Este usuario representa al administrador es el encargado de crear grupo de usuarios, eliminar grupos ver grupos, modificar su perfil generar reporte de los usuarios registrados ver los detalles de las sentencias. |
|  Usuario | Este usuario registra sus datos, inicia sesión, puede modificar perfil, utilizar la aplicación cargar un archivo de las sentencias SQL. |
|  Sistema | Este usuario corresponde al sistema (aunque su rol en la mayoría de los casos no es necesario), es necesario pensar que dentro de un actor autónomo que interactúa mediante mensajes de aceptación es un usuario del sistema. |

Figura 27 Usuarios del sistema (Fuente propia)

5.2.4 DIAGRAMAS DE CASOS DE USOS

Los diagramas de casos de uso se utilizan para instruir los requerimientos del sistema al ver cómo reacciona a eventos que se producen en su ámbito o en él mismo. Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Se establecieron las siguientes estructuras de los diagramas de casos de uso del aplicativo web como se muestra a continuación:

5.2.4.1 DIAGRAMA DE CASO DE USO (ACCESO AL SISTEMA)

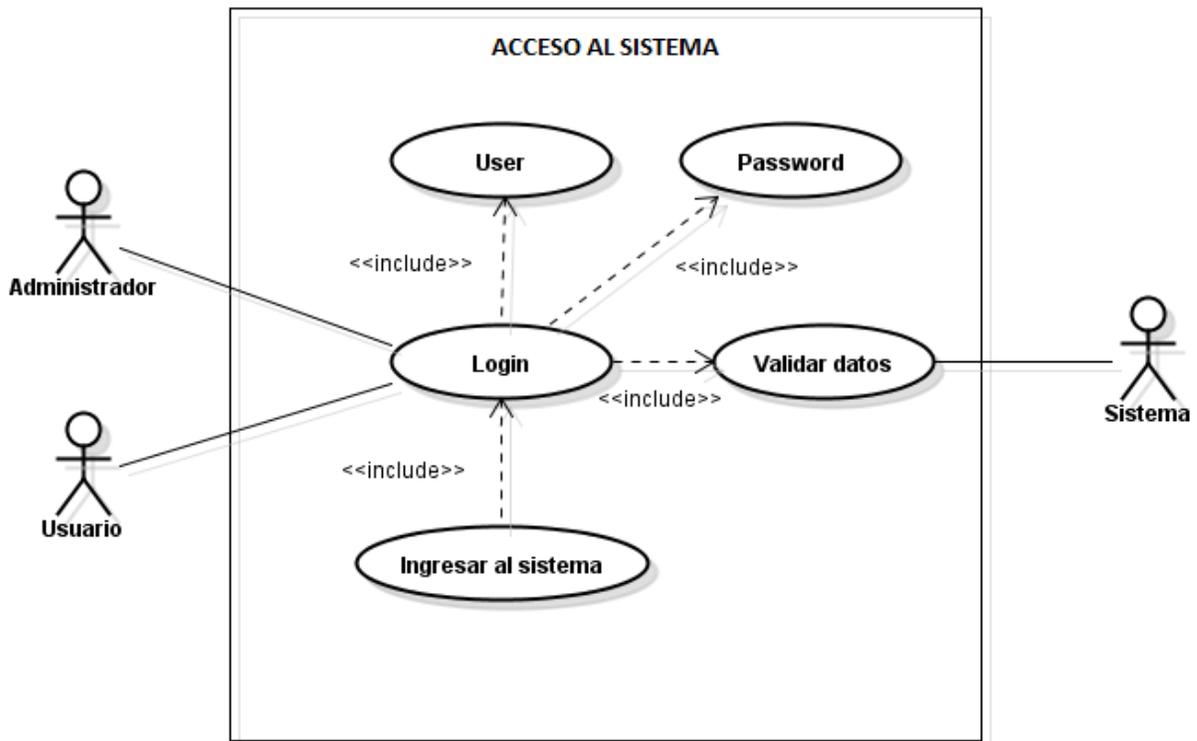


Figura 28 Diagrama de caso de uso Acceso al Sistema (Fuente propia)

5.2.4.2 DIAGRAMA DE CASO DE USO (ADMNISTRADOR)

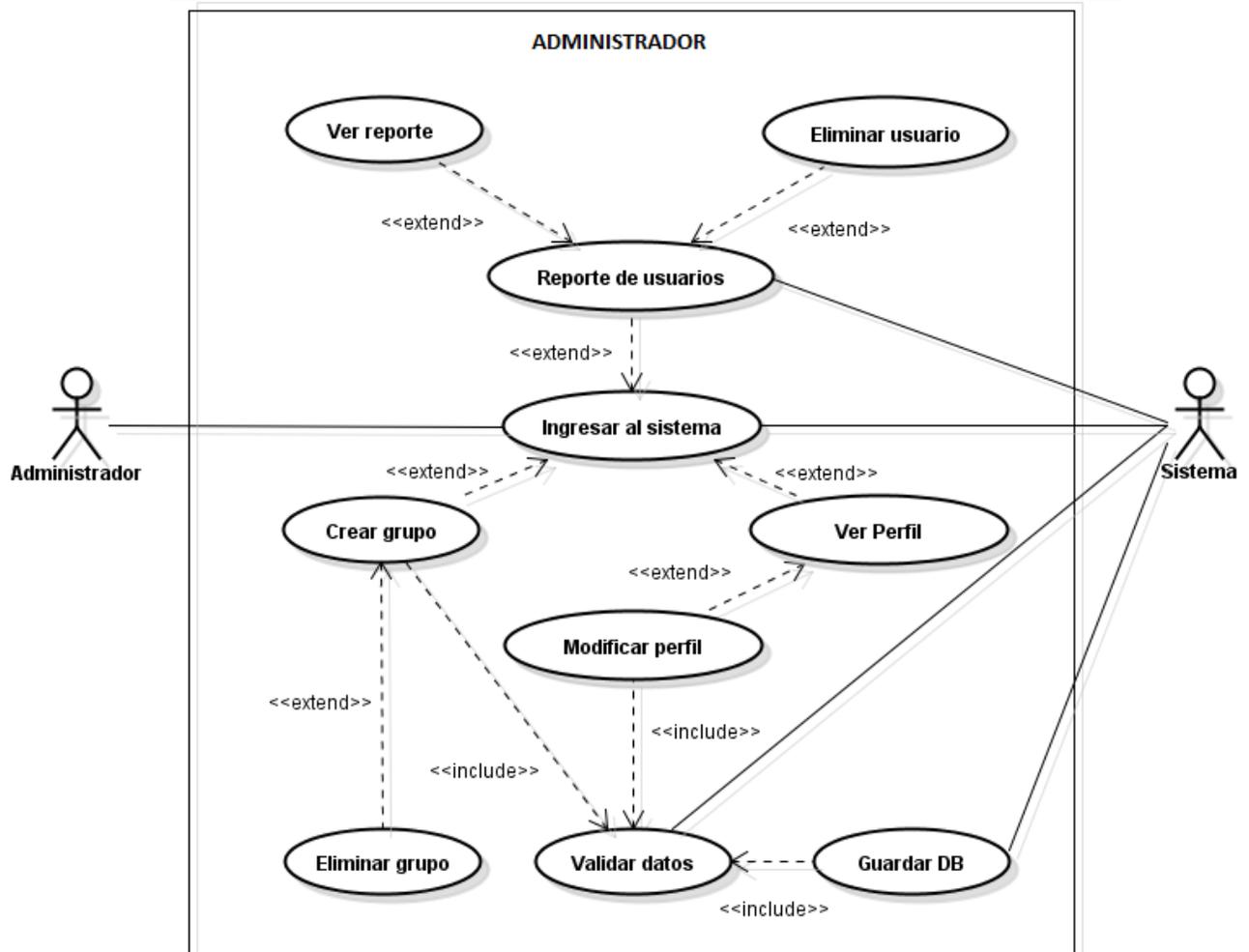


Figura 29 Diagrama de caso de uso Administrador (Fuente propia)

5.2.4.3 DIAGRAMA DE CASO DE USO (USUARIO)

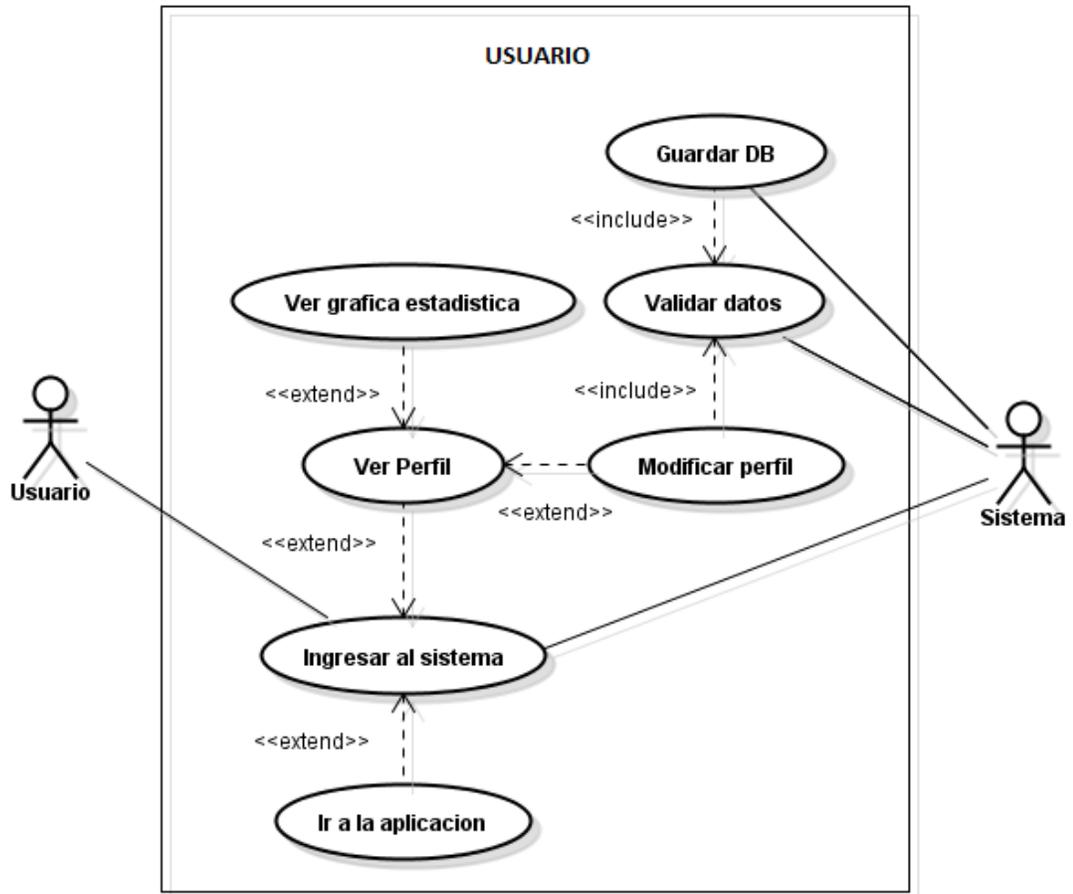


Figura 30 Diagrama de caso de uso Usuario (Fuente propia)

5.2.4.4 DIAGRAMA DE CASO DE USO (REGISTRO DE USUARIO)

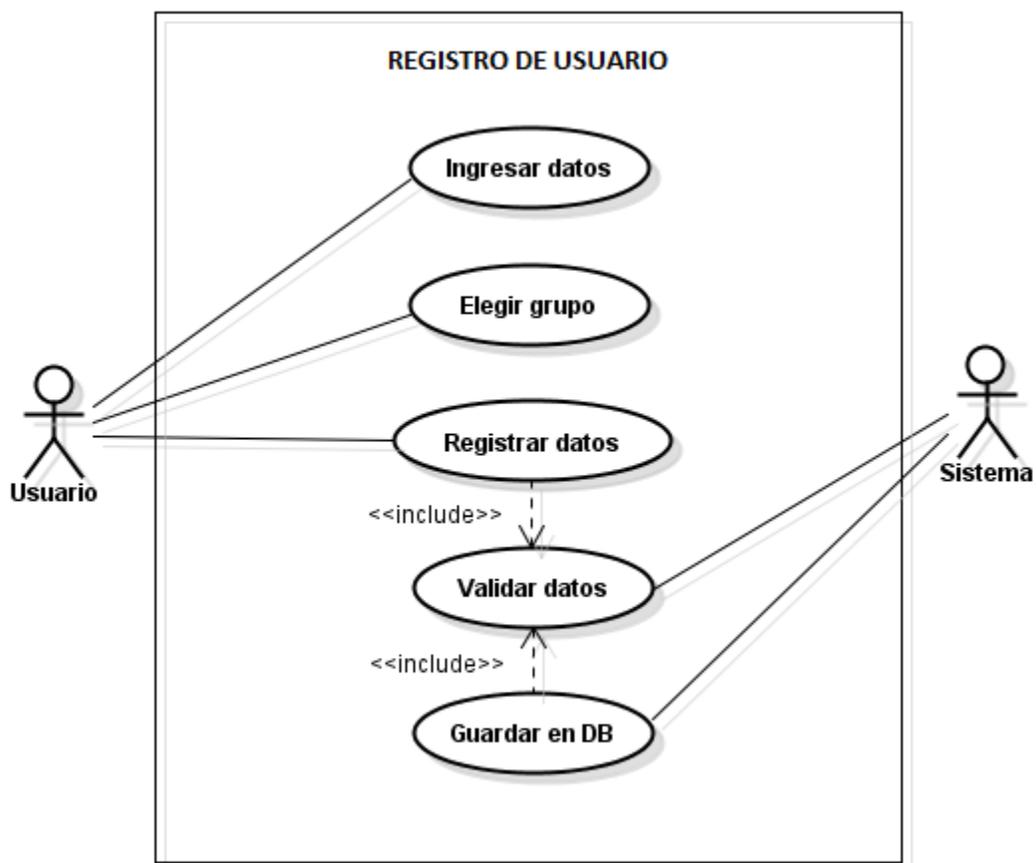


Figura 31 Diagrama de caso de uso Registro de usuario (Fuente propia)

5.2.4.5 DIAGRAMA DE CASO DE USO APLICACIÓN

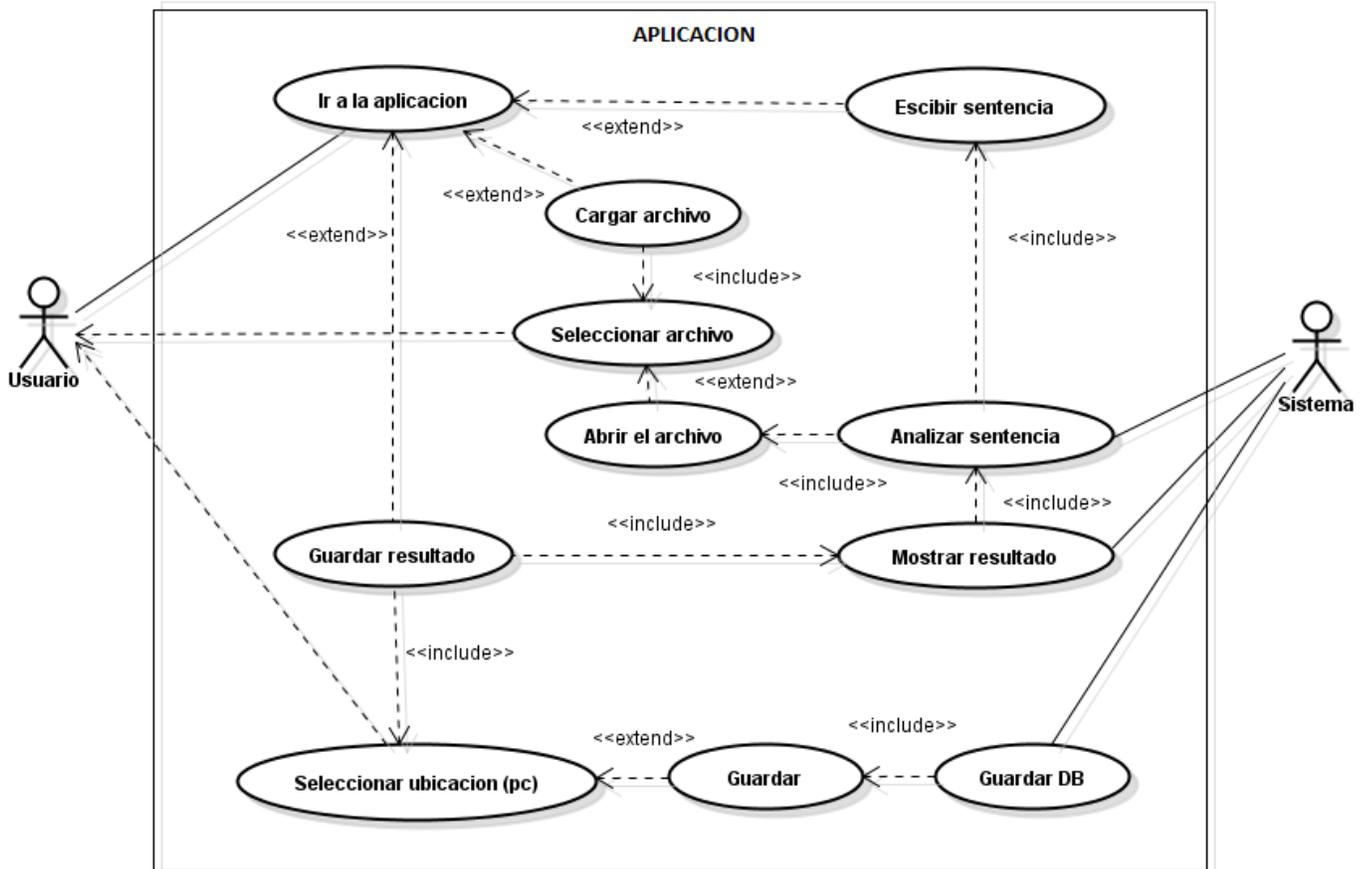


Figura 32 Diagrama de caso de uso Aplicación (Fuente propia)

5.2.5 DIAGRAMAS DE SECUENCIAS

Estos diagramas son los que muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. A pesar de que el diagrama de casos de uso permite el modelado de una vista *business* del escenario, el diagrama de secuencia muestra los detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

5.2.5.1 DIAGRAMA DE SECUENCIA ACCESO DEL SISTEMA.

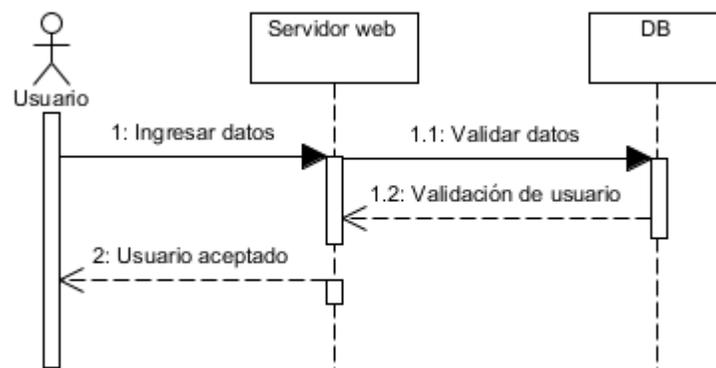


Figura 33 Diagrama de secuencia Acceso del sistema (Fuente propia)

5.2.5.2 DIAGRAMA DE SECUENCIA DEL ADMINISTRADOR

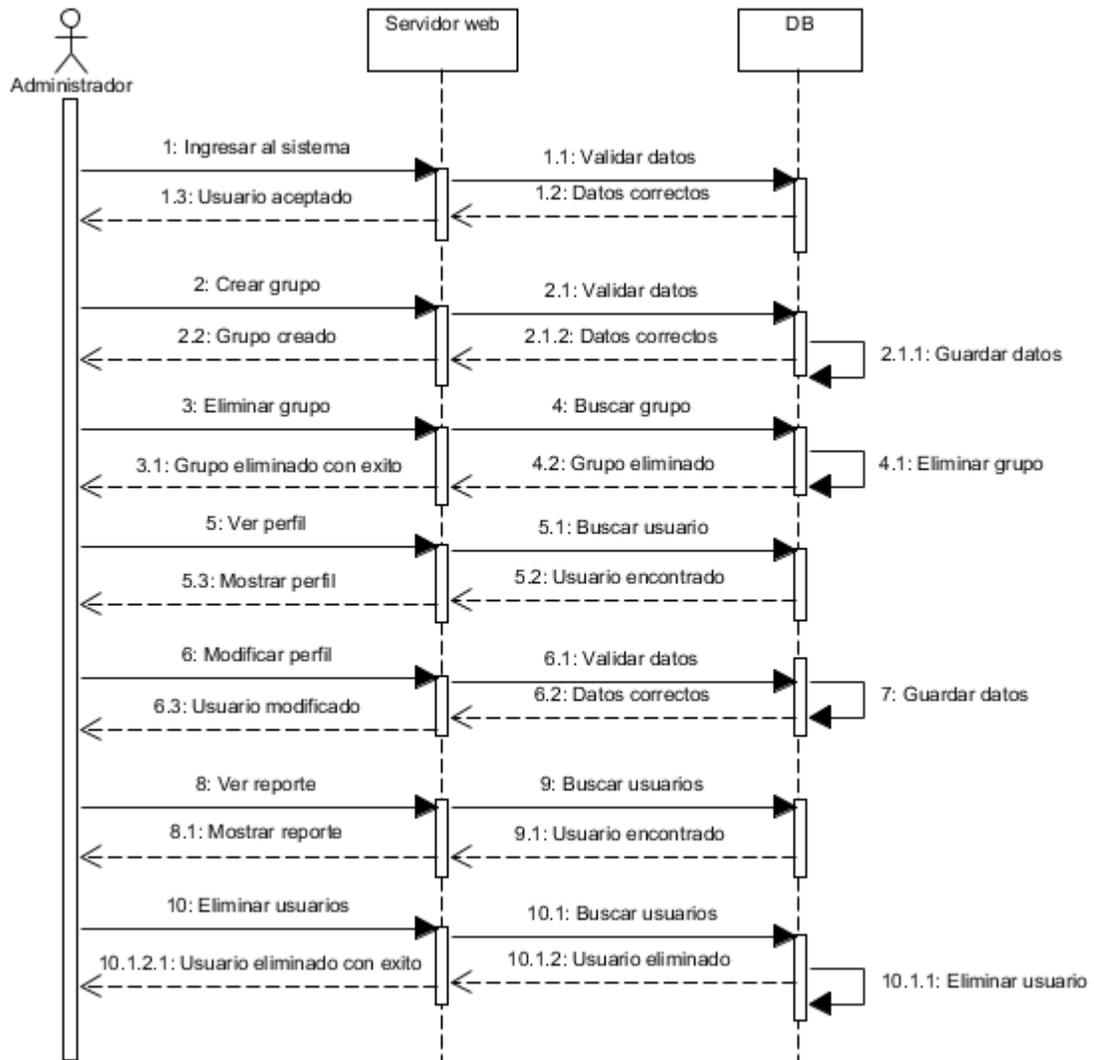


Figura 34 Diagrama de secuencia Administrador (Fuente propia)

5.2.5.3 DIAGRAMA DE SECUENCIA REGISTRAR USUARIO

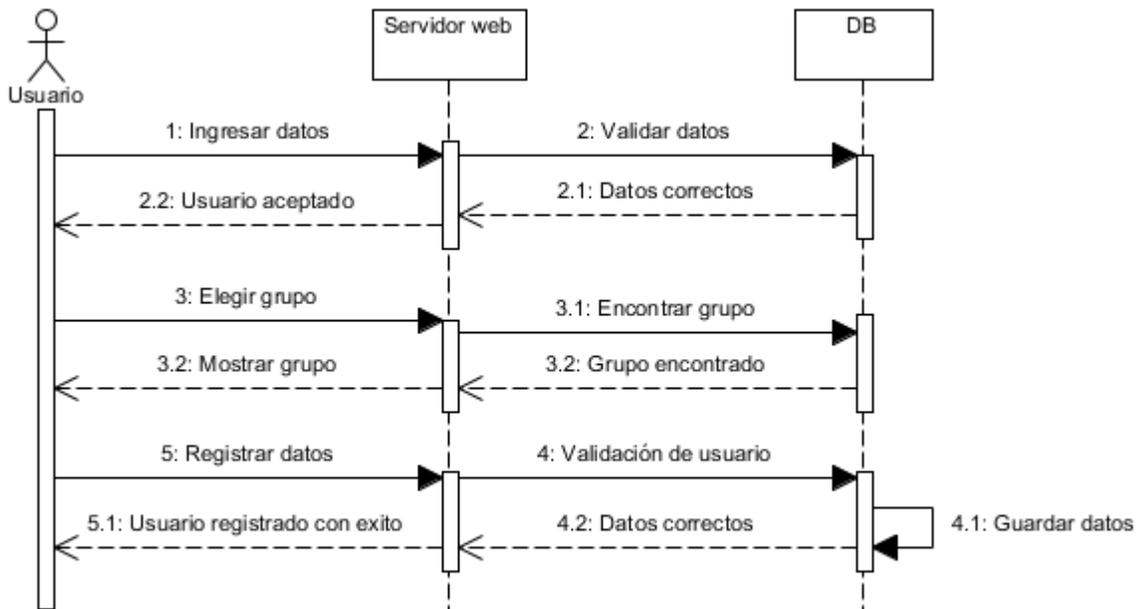


Figura 35 Diagrama de secuencia Registro de usuario (Fuente propia)

5.2.5.4 DIAGRAMA DE SECUENCIA USUARIO

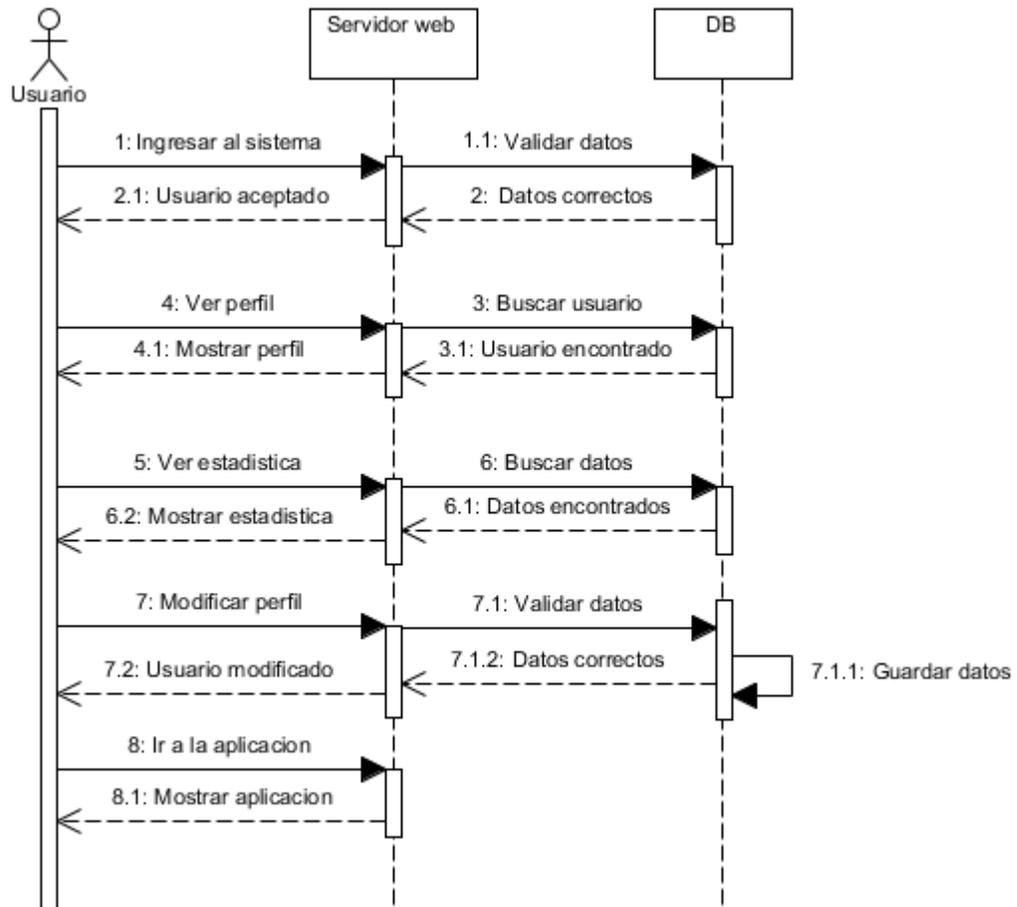


Figura 36 Diagrama de secuencia Usuario (Fuente propia)

5.2.5.5 DIAGRAMA DE SECUENCIA APLICACION

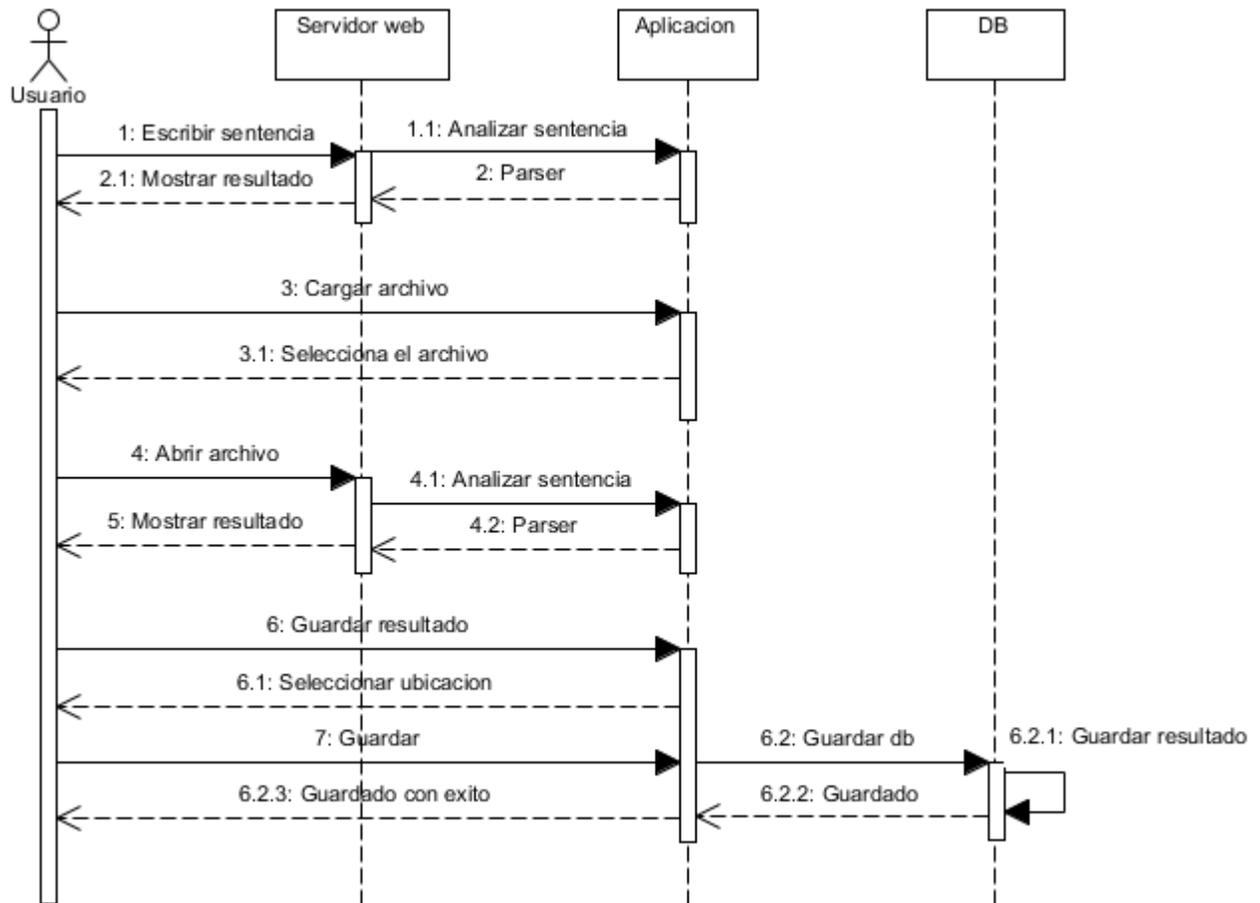


Figura 37 Diagrama de secuencia Aplicación (Fuente propia)

5.2.6 DIAGRAMAS DE COLABORACION

Un diagrama de comunicaciones, inicialmente llamado un diagrama de colaboración, es un diagrama de interacción que muestra información similar a los diagramas de secuencia pero su foco principal es en la relación de objetos que se muestran con conectores de asociación entre ellos. Los mensajes se agregan a las asociaciones y se muestran como flechas cortas apuntando en la dirección del flujo del mensaje. La secuencia de los mensajes se muestra a través de un esquema enumerado.

A continuación se muestran los diagramas de colaboración del sistema:

5.2.6.1 DIAGRAMA DE COLABORACION ACCESO AL SISTEMA

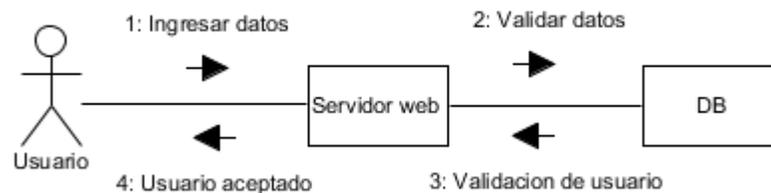


Figura 38 Diagrama de colaboración acceso al sistema (Fuente propia)

5.2.6.2 DIAGRAMA DE COLABORACION ADMINISTRADOR

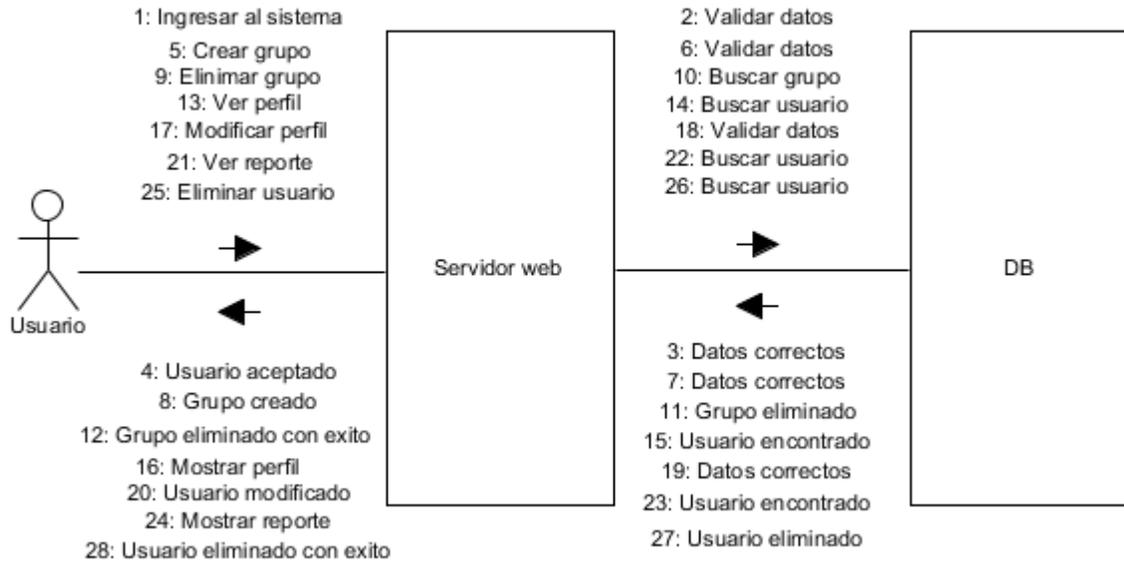


Figura 39 Diagrama de colaboración Administrador (Fuente propia)

5.2.6.3 DIAGRAMA DE COLABORACION REGISTRO DE USUARIO

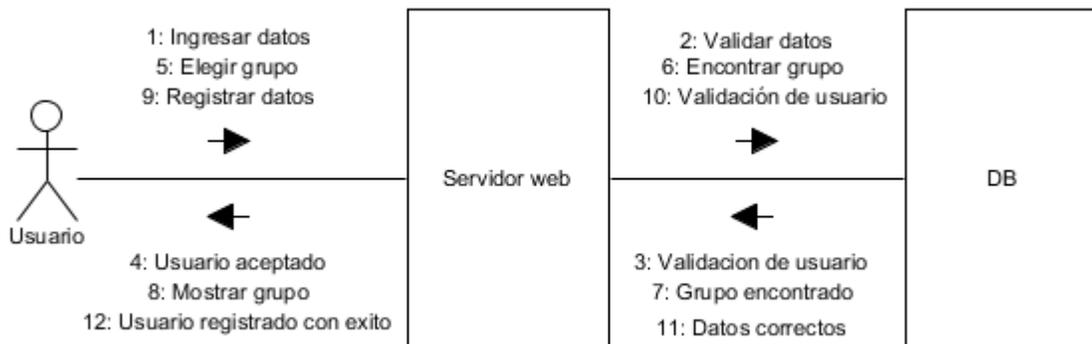


Figura 40 Diagrama de colaboración Registro de usuario (Fuente propia)

5.2.6.4 DIAGRAMA DE COLABORACION USUARIO

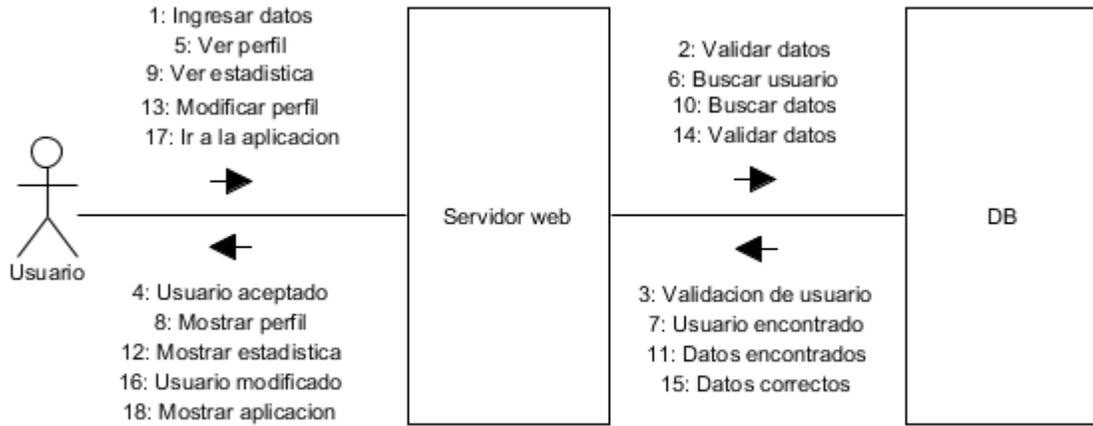


Figura 41 Diagrama de colaboración Usuario (Fuente propia)

5.2.6.5 DIAGRAMA DE COLABORACION APLICACION

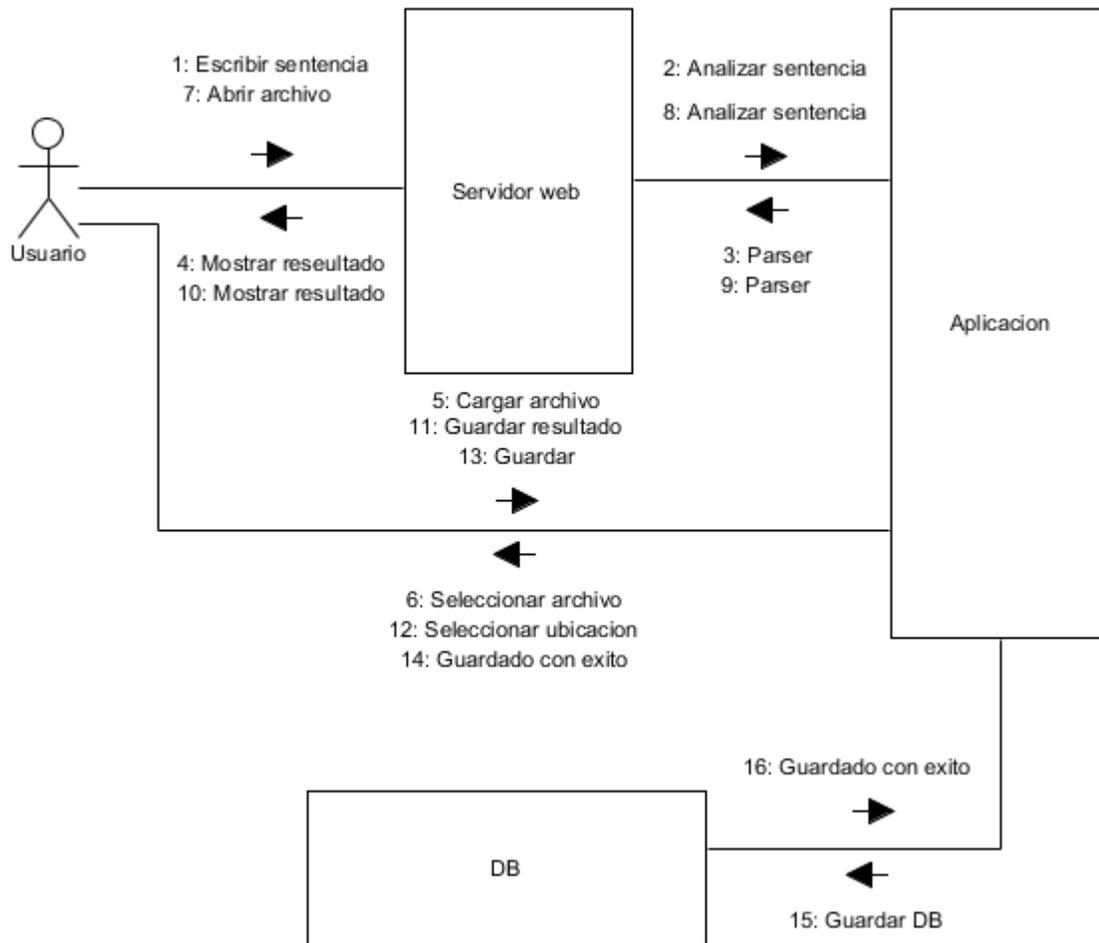


Figura 42 Diagrama de colaboración Aplicación (Fuente propia)

6. CONCLUSIONES

Del anterior proyecto investigativo se puede concluir que:

Una de las metas del proyecto era identificar los requerimientos para el manejo del lenguaje SQL-DML, se tuvo en cuenta que estas sentencias es un lenguaje estándar y basado en MySQL de la que se implicó el manejo de la documentación y los manuales de referencia de la misma; tales estándares me permitieron diseñara un modelo para el desarrollo del analizador sintáctico.

Dicho analizador se logró mediante programación orientada a objetos definida con los criterios investigados eh implementada en un componente de una aplicación que se ejecuta en el contexto de otro programa es decir un Applet.

Otras de las metas a alcanzar era de desarrollar una aplicación web dedicada al análisis sintáctico del lenguaje; el resultado de esto fue la implantación del Applet que fue programado como se mencionó antes dentro de la página web lo por su diseño y arquitectura me permitieron proporcionar un sistema de control y flexibilidad para el usuario que lo maneje.

Por ultimo una vez implementado la aplicación web se logró implementar pruebas de consulta por el usuario con sentencias SQL para verificación de la aplicación, de lo que arrojó que gran parte de los estudiantes que fueron los objetos de estudio manejan las sentencias SQL, pero no tienen conocimiento de la estructura completa, gracias a la implementación del analizador sintáctico y mediante las pruebas realizadas se pudo concluir que un gestor más intuitivo que permita al usuario saber en qué se equivocó durante una consulta de sentencias SQL-DML y que además proporcione una sugerencia para aquellos usuarios inexpertos, mejora el manejo de las sentencias SQL-DML, dándole cavidad a mi aplicación web. En un futuro a corto plazo estaría planeando modificar la aplicación web para que sea compatible con dispositivos móviles.

7. BIBLIOGRAFÍA

- Alconada González, D. (s.f.). *AlcoNQL*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/handle/10609/18849>
- apache, S. w. (s.f.). Obtenido de <http://www.mastermagazine.info/termino/3866.php>
- Applet. (s.f.). *mabellanas*. Obtenido de <http://www.dma.fi.upm.es/mabellanas/voronoi/java/java.htm>
- Automata de pila*. (s.f.). Obtenido de <http://www2.dis.ulpgc.es/~mluengo/automatas/teoria/tema4.pdf>
- Automata_finito. (s.f.). *wikipedia*. Obtenido de http://es.wikipedia.org/wiki/Automata_finito
- concepto, s. l. (s.f.). *wikipedia*. Obtenido de http://es.wikipedia.org/wiki/software_libre
- concepto, s. (s.f.). *wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/software>
- Diagramas. (s.f.). http://www.sparxsystems.com.ar/resources/tutorial/uml2_communicationdiagram.html.
- Dreamweaver. (s.f.). *Adobe*. Obtenido de http://es.wikipedia.org/wiki/Adobe_Dreamweaver
- Fornes, A. T. (2012). *generacion de casos de pruebas para SQL*. madrid, españa .
- HTML. (s.f.). *Wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/HTML>
- J.froufe. (s.f.). tutorial java.
- java. (s.f.). *java.com*. Obtenido de http://www.java.com/es/download/whatis_java.jsp
- Jimenez, P. M. (1997). *el lenguaje de programacion java*. anaya multimedia.
- mayorga, a. (2000). *Desig and implementation of a web based PL/SQL debugger using Oracle's debug API,suecia,corus.tecnologias estocolmo*. estocolmo.
- MYSQL, d. p. (s.f.). *danielpecos.com*. Obtenido de http://danielpecos.com/docs/mysql_postgres/x57.html

Naish, L. (1997). A Declarative Debugging Scheme. *journal of funtional and logic programming*.

NetBeans. (s.f.). *Wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/NetBeans>

PHP. (s.f.). *developer*. Obtenido de <https://developer.mozilla.org/es/docs/User:Marti1125>

programacion_declarativa. (s.f.). *clubensayos.com*. Obtenido de <http://clubensayos.com/tecnologica/paradigmas-de-programacion/1015235.html>

Service, W. (s.f.). *Wikipedia*. Obtenido de http://es.wikipedia.org/wiki/Servicio_web

sintactico, a. (s.f.). *wikipedia*. Obtenido de http://es.wikipedia.org/wiki/Analizador_sintactico

SQL. (s.f.). *wikipedia*. Obtenido de <http://es.wikipedia.org/wiki/S>

vela, N. j. (2009). *implementacion de un depurador declarativo para el lenguaje de bases de datos SQL*. madrid, españa.

web, A. (s.f.). *wikipedia*. Obtenido de http://es.wikipedia.org/wiki/Aplicación_web

web, S. (s.f.). *blogspot*. Obtenido de http://servidorwebenlainstitucioneducativa.blogspot.com/2009_11_01_archive.html

8. ANEXOS

8.1 DOCUMENTACION DE DIAGRAMA DE CASOS DE USO

| Caso de uso | 1 | Nombre | Acceso al sistema |
|--------------------------|--|---------------|-------------------|
| Autores | Usuario, administrador | | |
| Objetivo | Acceder al sistema | | |
| Descripción | Este caso de uso permite a los usuarios validar su identidad ante el sistema, muestra en pantalla un cuadro de dialogo donde se digita el login de usuario y contraseña para poder ingresar a la aplicación. | | |
| Precondición | El usuario debe estar registrado al sistema. | | |
| Flujo normal | Una vez validado en el sistema el usuario podrá acceder a su cuenta de usuario. | | |
| Flujo alternativo | Si no posee una cuenta de usuario, necesitara crear una cuenta en el registro de usuario. | | |
| Excepciones | En el caso de que el usuario y la contraseña son incorrecto el sistema mandara un mensaje de error de validación. | | |

Tabla 7 Documentación de diagrama de casos de uso #1 (Fuente propia)

| | | | |
|--------------------------|---|---------------|---------------|
| Caso de uso | 2 | Nombre | Administrador |
| Autores | Administrador | | |
| Objetivo | Ingresar como administrador | | |
| Descripción | Este caso de uso permite al administrador acceder al sistema manejar la información correspondiente a los usuarios que usan la aplicación, además de crear grupos, eliminar grupo y editar su perfil de administrador. | | |
| Precondición | El usuario debe ser el administrador con su respectiva usuario y contraseña | | |
| Flujo normal | El administrador acceder al sistema y maneja la información correspondiente a su cuenta de usuario. | | |
| Flujo alternativo | Ninguno. | | |
| Excepciones | En el caso de no poder ingresar como administrador, el sistema mandara un mensaje de error de validación. Además tendrá que comunicarse con el administrador de la base de datos. Ingreso no valido al sistema, le saldrá un mensaje advirtiendo que no ha iniciado sesión y será redireccionado a login. | | |

Tabla 8 Documentación de diagrama de casos de uso #2 (Fuente propia)

| Caso de uso | 3 | Nombre | Usuario |
|--------------------------|---|---------------|---------|
| Autores | Usuario | | |
| Objetivo | Ingresar como usuario | | |
| Descripción | Este caso de uso permite al usuario acceder al sistema manejar que pueda interactuar con la aplicación y administrar su perfil. | | |
| Precondición | Haber ingresado como usuario del sistema. | | |
| Flujo normal | El usuario acceder al sistema y manejar la información correspondiente de su cuenta. | | |
| Flujo alternativo | Ninguno | | |
| Excepciones | Ingreso no valido al sistema, le saldrá un mensaje advirtiéndolo que no ha iniciado sesión y será redireccionado a login. | | |

Tabla 9 Documentación de diagrama de casos de uso #3 (Fuente propia)

| Caso de uso | 4 | Nombre | Registro de usuario |
|--------------------------|--|---------------|---------------------|
| Autores | Usuario | | |
| Objetivo | Registrarse al sistema | | |
| Descripción | Se muestra en pantalla un formulario donde se registran los datos para registrarse al sistema este consta del nombre, apellido, usuario, contraseña ,correo y de los grupos que estén disponible | | |
| Precondición | Usuario no debe existir en el sistema. | | |
| Flujo normal | El usuario ha realizado con éxito el registro y podrá acceder al sistema. | | |
| Flujo alternativo | Si el usuario ya tiene una cuenta; podrá ingresar al sistema. | | |
| Excepciones | Si el usuario ya existe o alguno de sus datos, saldrá una advertencia de duplicación de datos. | | |

Tabla 10 Documentación de diagrama de casos de uso #4 (Fuente propia)

| Caso de uso | 5 | Nombre | Aplicación |
|--------------------------|--|---------------|------------|
| Autores | Usuario | | |
| Objetivo | Manejar la aplicación | | |
| Descripción | Se muestra en pantalla la aplicación a manejar tipo Applet por donde el usuario podrá interactuar con ella y realizar operaciones como analizar sentencias , subir archivo con sentencias para analizar y guardar resultados | | |
| Precondición | Haber ingresado con usuario al sistema. Tener los niveles de seguridad Java compatibles con la aplicación. | | |
| Flujo normal | Se hará uso de la aplicación donde podrá hacer las operaciones de análisis de la sentencia. | | |
| Flujo alternativo | Ninguno. | | |
| Excepciones | Si no carga la aplicación asegúrese de tener java instalado. | | |

Tabla 11 Documentación de diagrama de casos de uso #5 (Fuente propia)

8.2 DIAGRAMAS DEL SISTEMA

8.2.1 ARQUITECTURA DEL SISTEMA

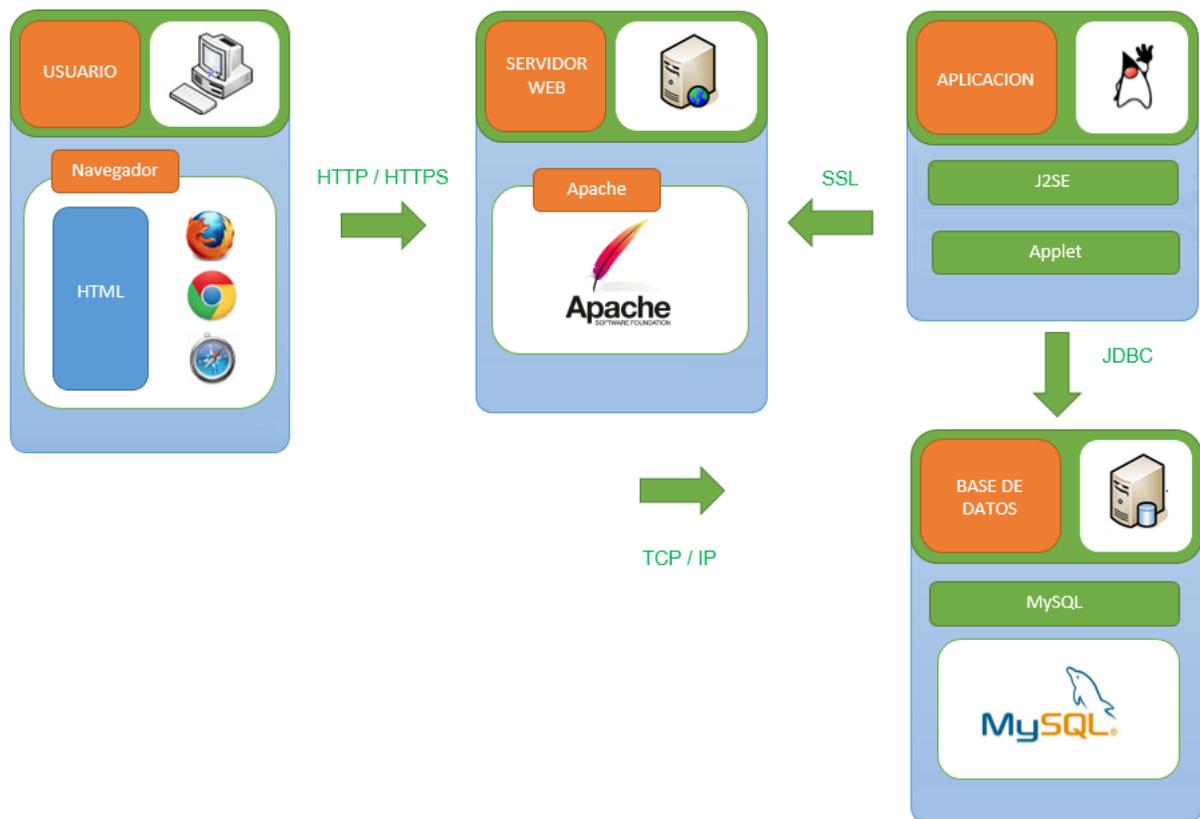


Figura 43 Arquitectura del sistema # 2 (Fuente propia)

8.3 MANUAL DE USUARIO

El objetivo de este apartado es el de explicar el funcionamiento de la tesis “diseño e implementación de la aplicación web orientada al análisis sintáctico de sentencias SQL-DML en base de datos MySQL”

El acceso a la aplicación se realizará mediante el empleo de un navegador WEB. Cabe destacar que esta aplicación es compatible con Google Chrome, Mozilla Firefox y safari

Para un completo funcionamiento de la aplicación, su respectivo control y manejo se dará a conocer las respectivas instrucciones.

Se ingresa a la página <http://proyectosqlparser.tk/> donde mostrara la siguiente página



Figura 44 Aplicación web Índice (Fuente propia)

En esta se solicita un usuario y contraseña para ingresar al sistema, en dicho caso que no lo tenga debe ir a registrarse

Sabias que!

Muchos compiladores y analizadores no ayudan mucho en dar una sugerencia en como se debe corregir el error encontrado.
Sqlparser no solo encuentra el error ,te brinda sugerencias y datos estadísticos definidos en que tanto te hallas equivocado.

Regístrate!

Nombre:

Apellido:

Usuario:

Contraseña:

R-Contraseña:

Email:

Grupo:

Figura 45 Registro de usuario (Fuente propia)

En esta el usuario debe ingresar nombre, apellido, un nombre de usuario, una contraseña, correo y por ultimo elegir un grupo el cual solo estará habilitado por el administrador.

Una vez registrado en sistema el usuario podrá acceder a ella mostrándole un módulo de interfaz de usuario

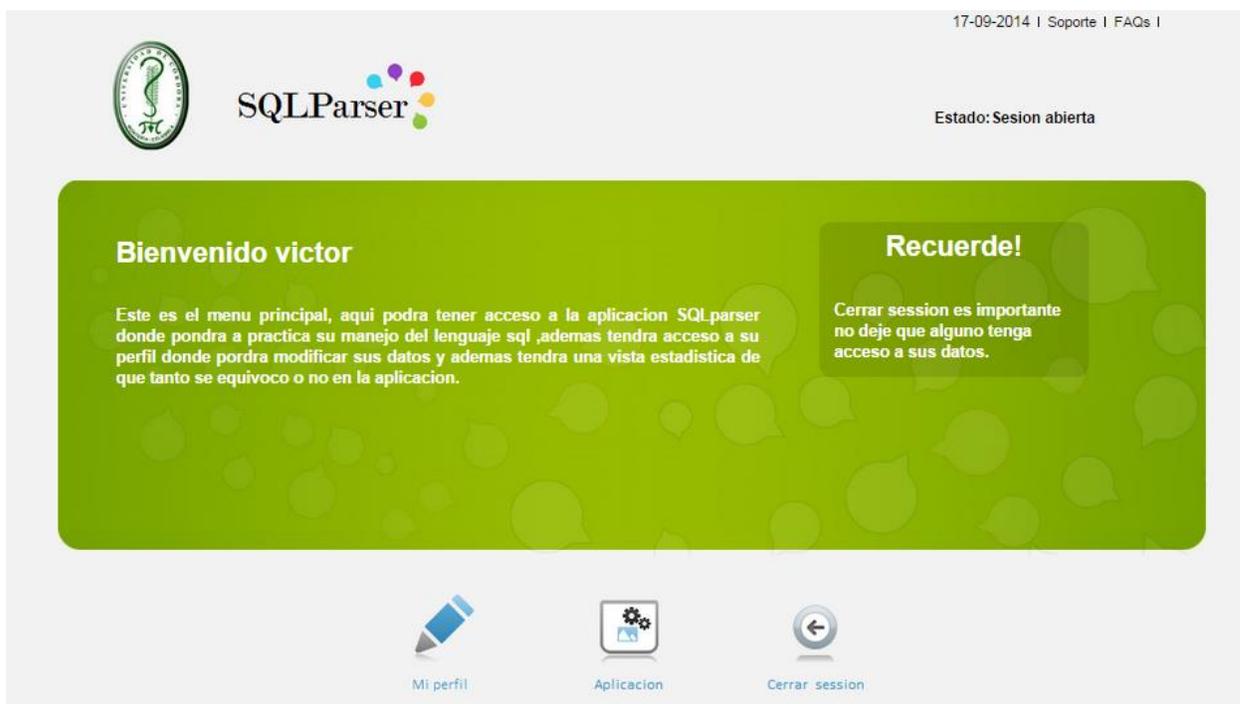


Figura 46 Modulo de usuario (Fuente propia)

A continuación la página nos presenta 3 opciones:

- ✓ Mi perfil
- ✓ Aplicación
- ✓ Cerrar sección

En mi perfil se muestra información del usuario, sus datos y la capacidad de modificarlas además muestra una gráfica estadística generada por un reporte de la aplicación al usarse.



Figura 47 Modulo perfil usuario (Fuente propia)

En aplicación al ingresar nos mostrara un pequeño tutorial de cómo funciona la aplicación

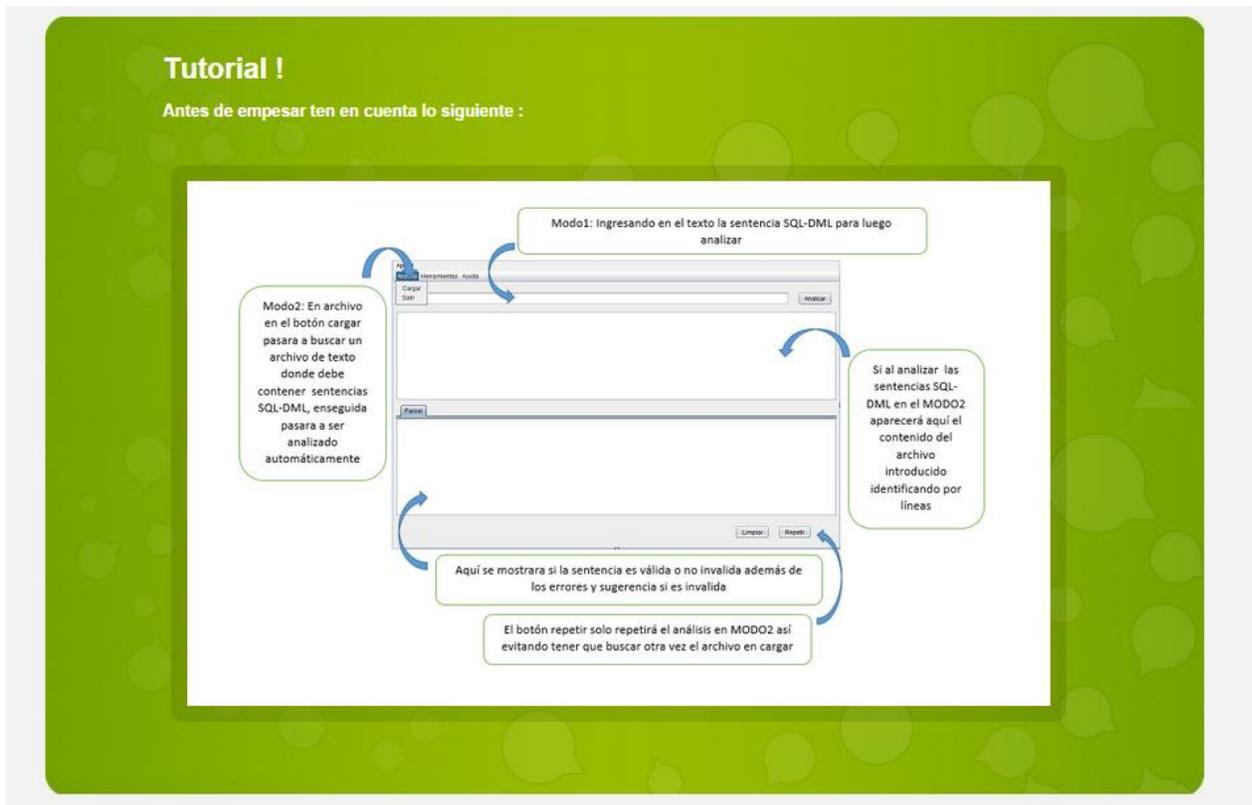


Figura 48 Modulo tutorial (Fuente propia)

Seguido de eso iremos y seleccionaremos el icono de la aplicación como tal para ingresar



Figura 49 Icono acceso a la aplicación (Fuente propia)

Una vez en la página de la aplicación hay que tener en cuenta ciertos requisitos

Debe tener instalado java



Figura 50 Acerca de java (Fuente propia)

Esta debe ser en versiones anteriores a 1.8 ya que los niveles de seguridad son muy riguroso para este tipo de aplicación que será un Applet.

Cabe mencionar que se configurara el nivel de seguridad de java para así no tener problemas con ella.

Aquí se muestra como debe estar configurado java



Figura 51 Configuración java (Fuente propia)

Una vez instalado java vamos a la aplicación hay nos aparecerá una advertencia de seguridad así como esta

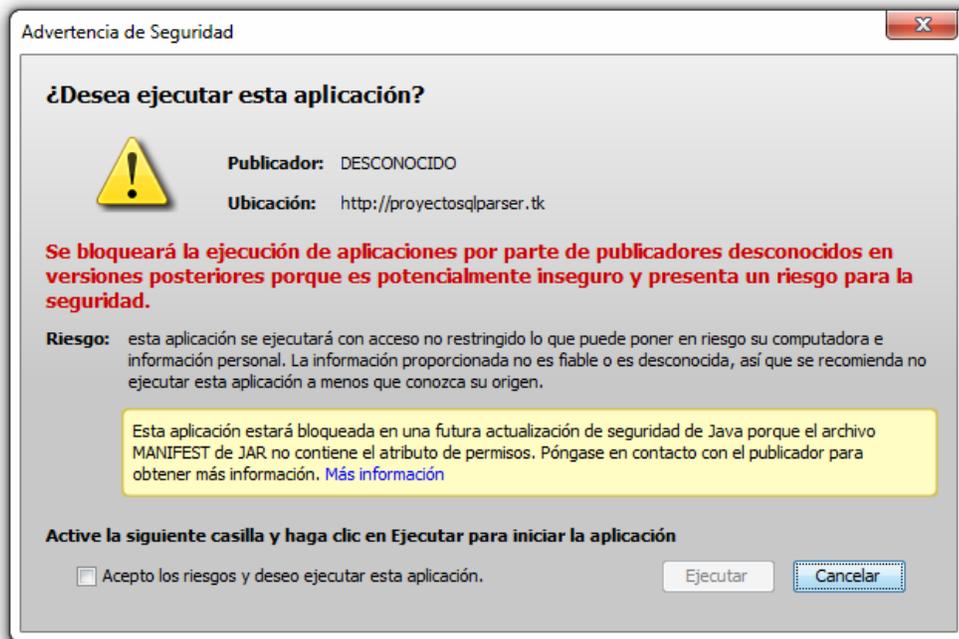


Figura 52 Advertencia de seguridad java (Fuente propia)

En donde nos da una breve descripción de los riesgos aun así la aplicación es segura. Vemos que en darle en más información nos dará una aclaración más concerniente a la aplicación.

Cabe mencionar así como dice en la configuración de java para que el applet funcione debe tener una firma digital

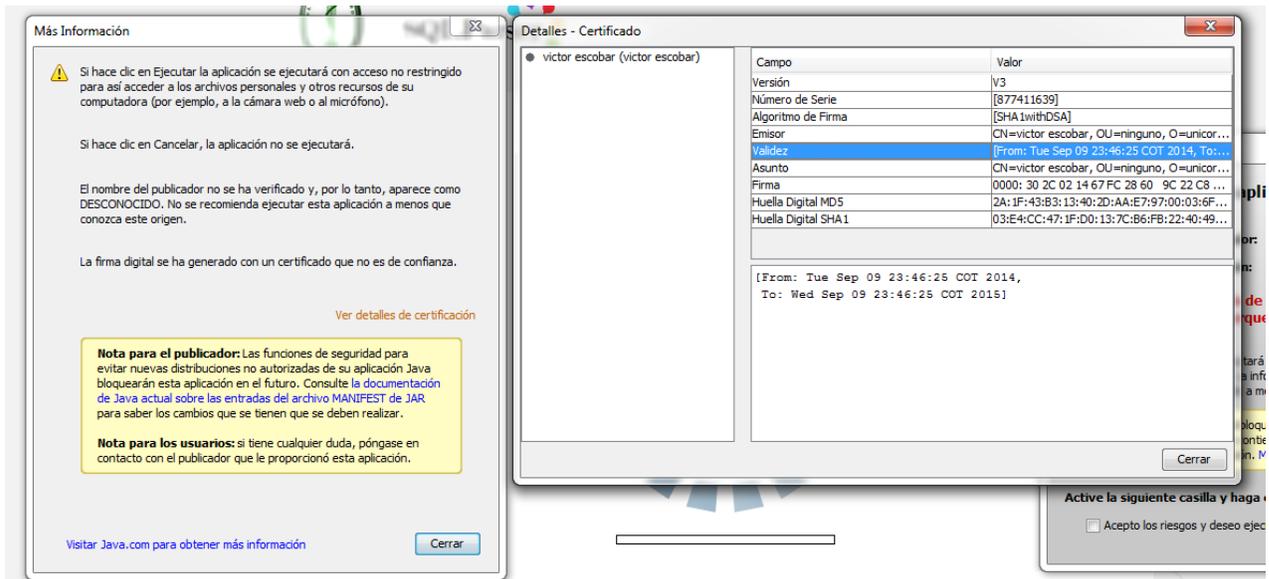


Figura 53 Detalle certificado (Fuente propia)

Aquí vemos que ver detalle de certificación nos muestra que está firmado digitalmente.

Una vez realizado todo esto ya podremos usar la aplicación Applet.

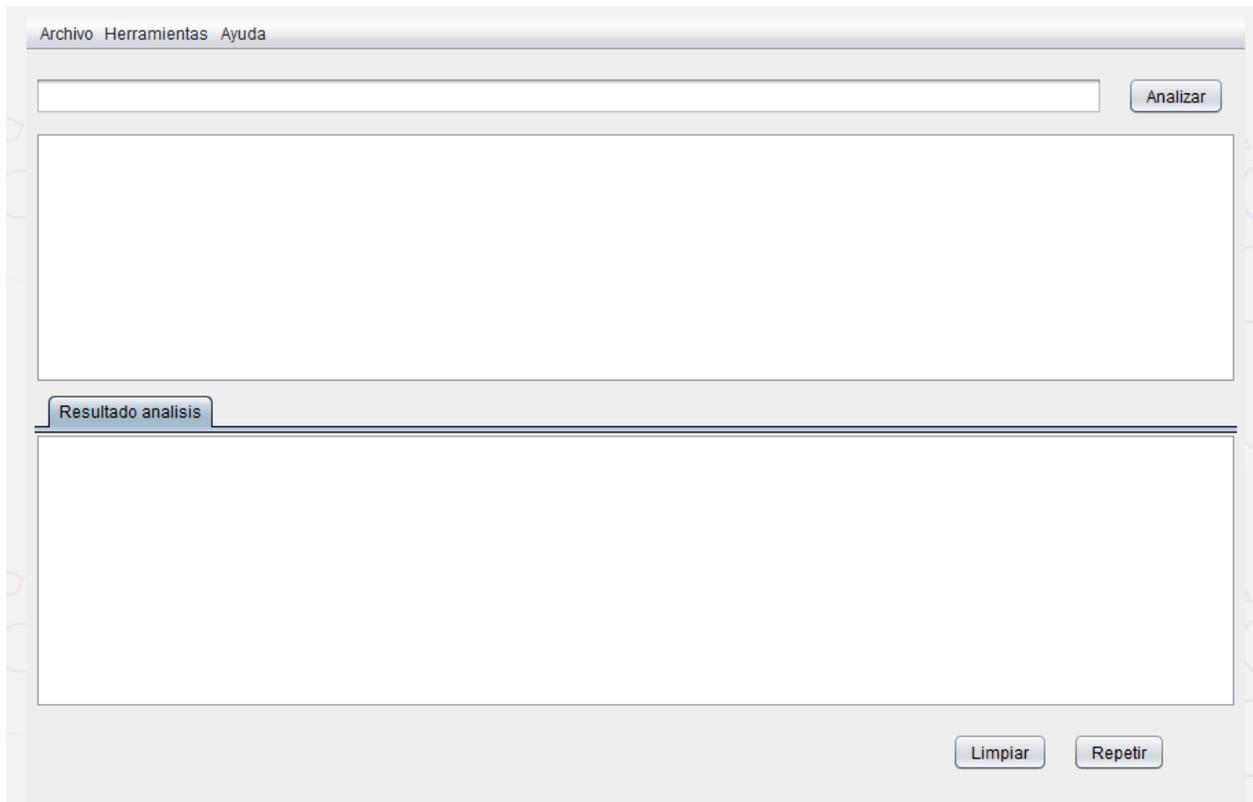


Figura 54 Aplicación Applet (Fuente propia)

Una vez en la aplicación en la interfaz veremos botones tales como

- Analizar
- Archivo
- Herramientas
- Repetir
- Limpiar

Como la función de la aplicación es analizar sentencia SQL tenemos dos formas de analizar una que es mediante interfaz y el otro mediante archivo introducido.

- ✓ En analizar se analizara la sentencia sql escrita en la interfaz

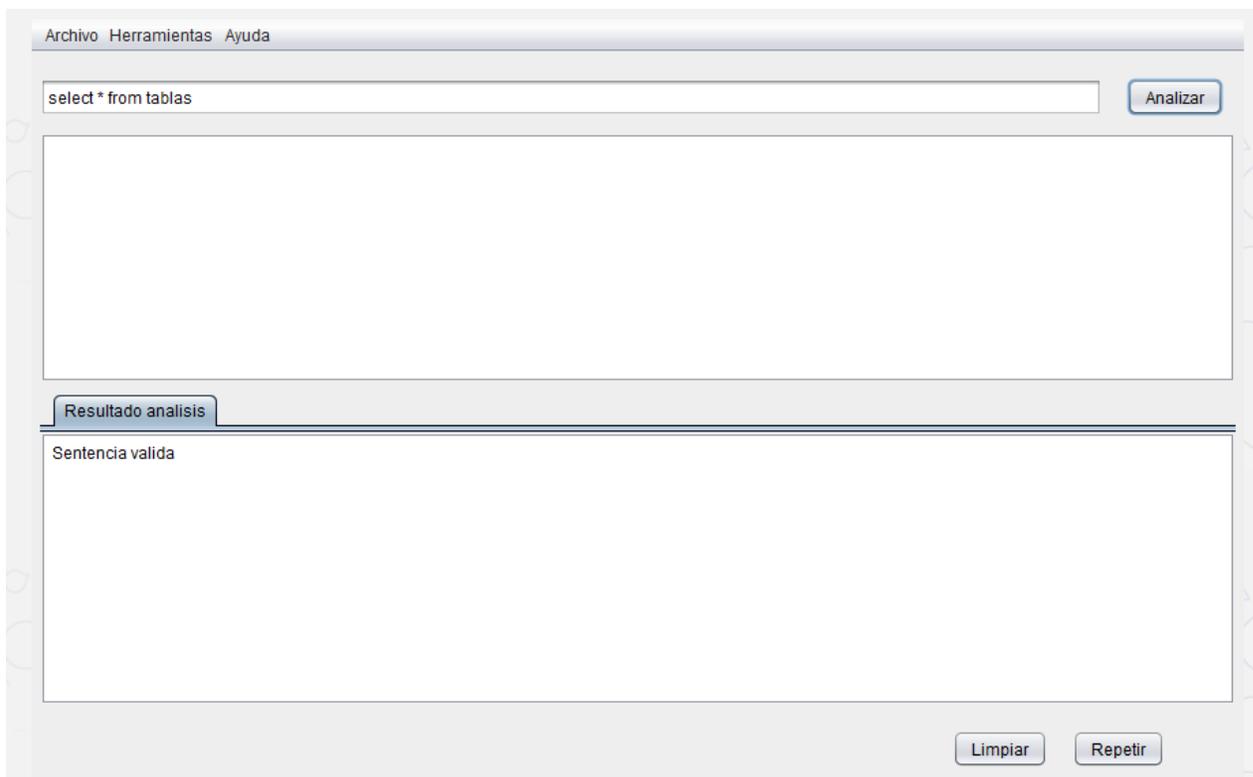


Figura 55 Prueba aplicación Applet (Fuente propia)

Notamos que nos mostrara el resultado del análisis

- ✓ En archivo nos permite carga un archivo para posteriormente ser analizado, este de ser suministrado como .txt

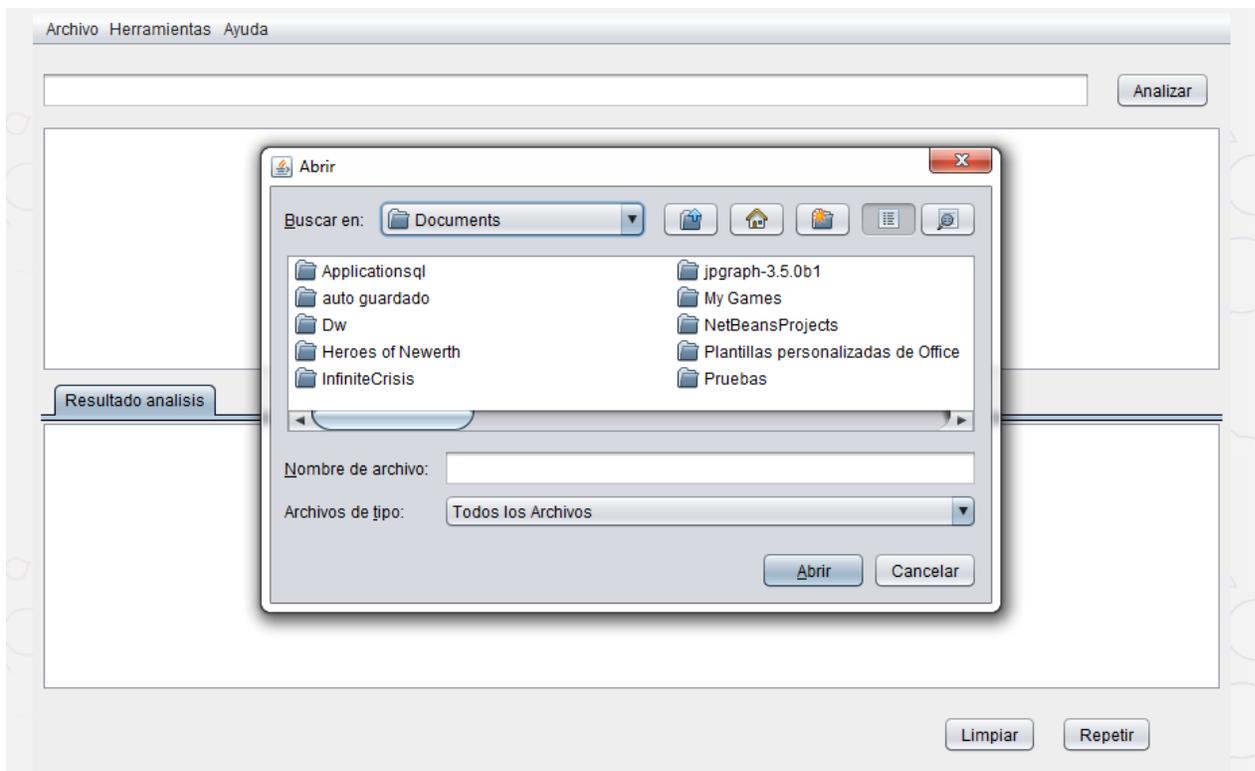


Figura 56 Aplicación Applet Cargar archivo (Fuente propia)

Dicho archivo debe contener varias sentencias SQL pero cada una debe estar terminada con punto y coma (;) así la aplicación diferencia una de la otra.

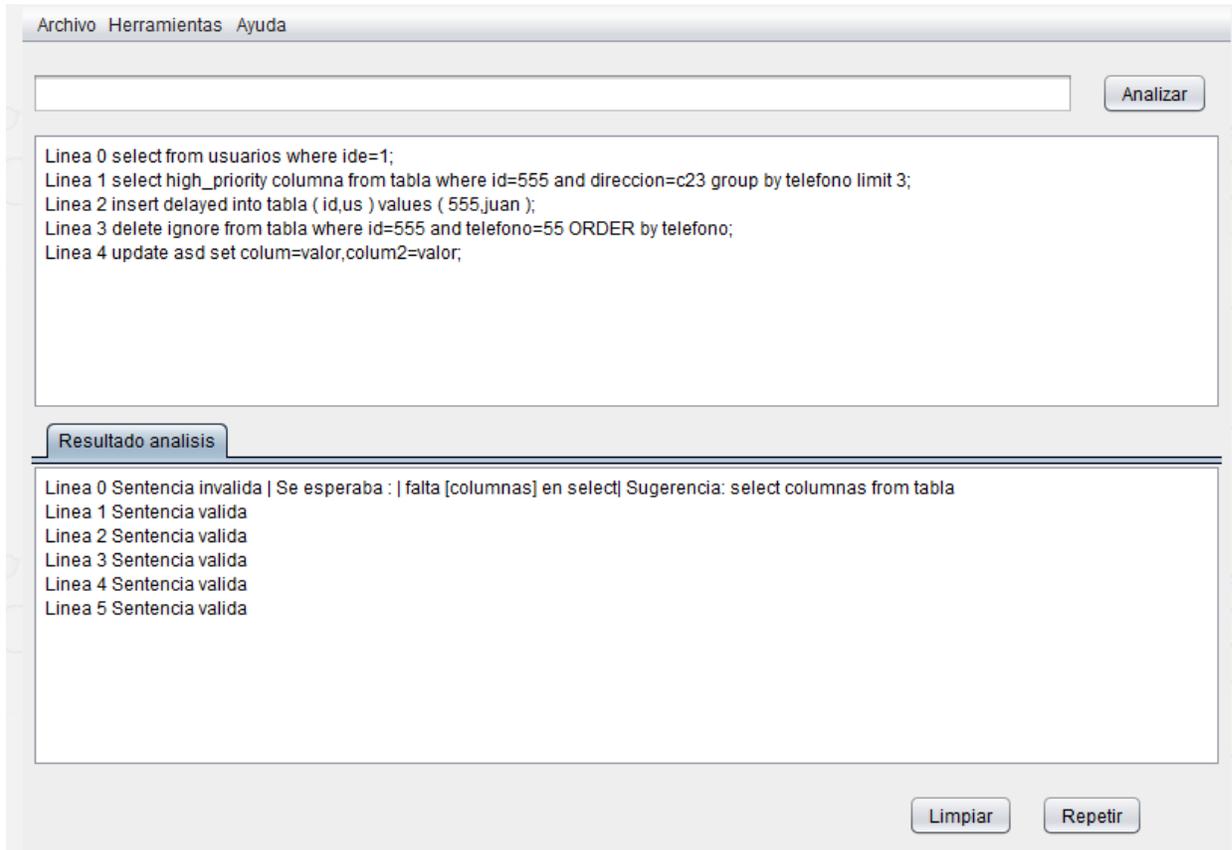


Figura 57 Prueba aplicación cargar archivo (Fuente propia)

Vemos que nos arroja el análisis de las sentencias y en la parte de arriba vemos todas las sentencias que estaban en el archivo plano.

- ✓ En el botón herramientas se podrá guardar los resultados tanto en la pc como en la base de datos.
- ✓ En respetar volverá a repetir el último análisis que se generó con el archivo plano así no tenerlo que buscar de nuevo.
- ✓ El limpiar, limpiara toda la interfaz.

Un actor especial del sistema es el de administrador este al igual que el de usuario tiene un modulo



Figura 58 Modulo administrador (Fuente propia)

Aquí vemos que tiene al igual que el usuario nos presenta unas opciones en este caso

- Mi perfil
- Reporte
- Crear grupos
- Cerrar sesión

- ✓ Mi perfil es igual al del usuario con la exención que no muestra diagramas

Recuerde !!

Estos son sus datos de admin solo existira 1 cuenta de admin asi que por favor recuerdelo, en caso que se le olvide la contraseña ingrese a la base de datos de hostinger.

Datos de usuario!

Nombre:

Apellido:

Usuario:

Contraseña:

R-Contraseña:

Figura 59 Modulo perfil de administrador (Fuente propia)

El administrado solo podrá modificar su perfil como único administrador del sistema.

- ✓ En reporte observamos que tenemos la lista de usuarios registrados

Reporte usuarios

Aquí esta el listado de usuarios que estan registrados en el sistema tenga en cuenta que los usuarios que no hallan usado la aplicacion no tendran detalles ademas si elimina un usuario se eliminará de manera permanente el registro.

Grupos: Todos

| Grupo | ID | Nombre | Apellido | Detalle | Eliminar |
|-------|----|--------|----------|---------|----------|
| 1 | 1 | | | | X |
| 1 | 14 | | | | X |
| 1 | 15 | | | | X |
| 2 | 13 | victor | Escobar | | X |
| 2 | 16 | | | | X |
| 2 | 17 | | | | X |
| 2 | 18 | | | | X |
| 2 | 19 | | | | X |
| 2 | 20 | | | | X |

Anterior 1 2 3 4 5 Siguiete

Figura 60 Modulo reporte de usuario (Fuente propia)

Además podemos ver en detalle su reporte, sino tiene el sistema mostrara que no lo ha generado.

Se podrá eliminar al usuario esto debido a que un usuario este dentro de un grupo que no pertenece.

Además se puede organizar por grupo así tener una vista generalizada.

✓ Seguirnos con crear grupo



Figura 61 Modulo crear grupo (Fuente propia)

Aquí vemos los grupos existentes al lado izquierdo, del lado derecho notamos que están las opciones de crear grupo y eliminar.

Por ultimo tanto el usuario como el administrador tienen la opción de cerrar sesión como todo aplicativo web.

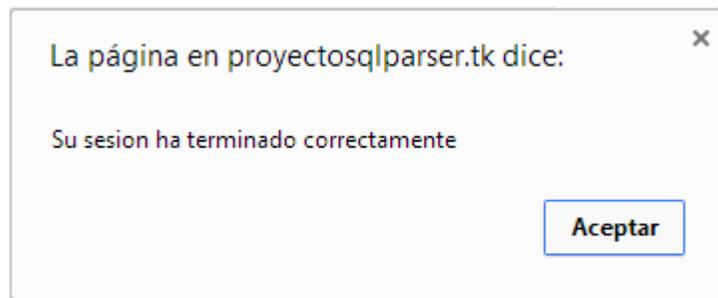


Figura 62 Aviso cerrar sesión (Fuente propia)

8.4 MANUAL DE INSTALACION

El objetivo de este manual es el de llevar a cabo la instalación de la aplicación web orientada al análisis sintáctico de sentencias SQL-DML en base de datos MySQL

Antes de la instalación del sistema WEB por favor asegúrese de que ha restaurado las bases de datos pertinentes.

Requerimientos necesarios

Para el correcto funcionamiento de la aplicación WEB es necesario tener los siguientes requerimientos:

Hostin que soporte PHP, MsqI y Apache; un dominio , las paginas a subir y la base de datos a exportar.

Hosting

En mi caso eh utilizado como hostin la página web www.hostinger.com el cual tiene las siguientes características:

Hosting en español con PHP y MySQL, sin publicidad, con cPanel y creador de sitios web.

Características del hosting:

- ✓ 2 GB de espacio
- ✓ 100 GB de transferencia mensual
- ✓ 2 bases de datos MsqI
- ✓ 2 dominios aparcado
- ✓ 2 usuarios ftp
- ✓ 2 subdominios
- ✓ 2 cuentas de correo
- ✓ Instalador automático(50 scripts)
- ✓ Acceso consola SSL
- ✓ 2 administradores de archivos
- ✓ Estadísticas del sitio
- ✓ Sin publicidad

Servicio técnico: Tiene foro de soporte y sistema de tickets.

Uptime (si se mantiene constante el servicio): las caídas del servidor no son constantes y si se cae no dura más de 2 min.



Figura 63 Hosting (Fuente propia)

Dominio

En hostinger le permite al usuario crear el dominio mas no de regístralo como tal.

Sabemos que los dominios terminados en .com, .ar , .mx son de tipo paga, así que en este caso se ha utilizado un registro de domino gratis.

Para crearlo gratis se debe ingresar a <http://www.dot.tk/es/index.html?lang=es>

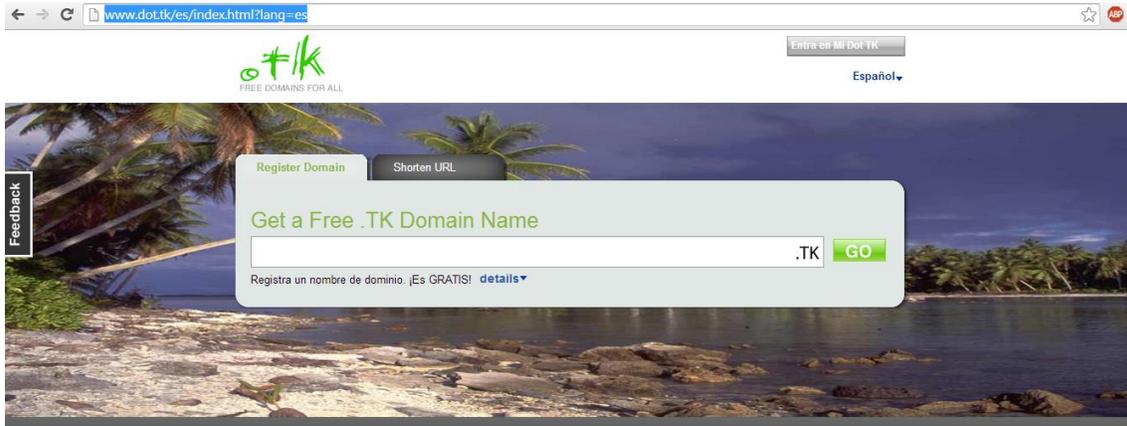


Figura 64 Dominio (Fuente propia)

Una vez ingresado se pasa a indicarle el dns del host en este caso para direccionarlo a los servidores de hostinger

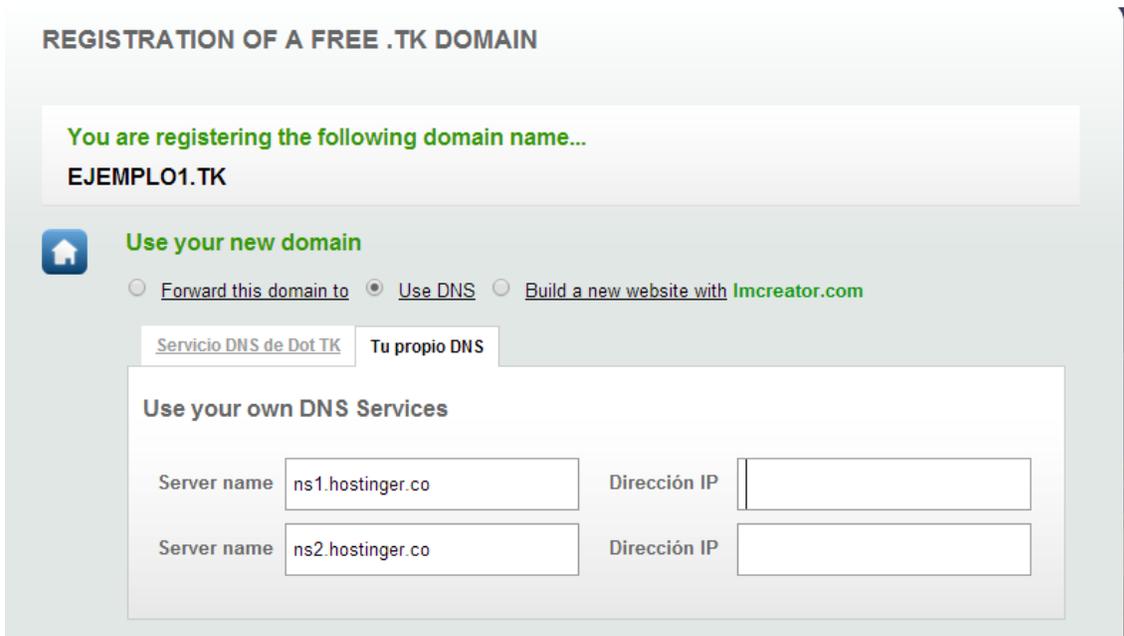


Figura 65 Registro de dominio (Fuente propia)

Los DNS se pueden obtener en detalles con sus respectivas direcciones ip, en cuentas de usuario

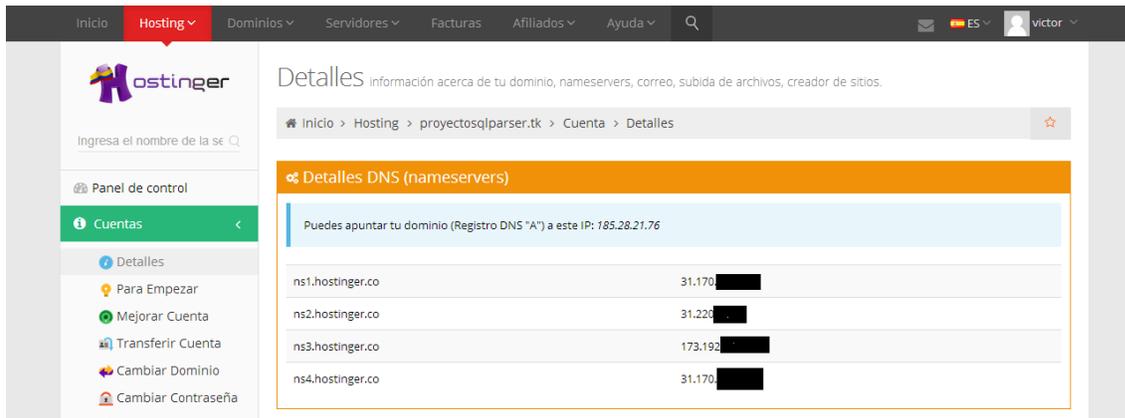


Figura 66 DNS (Fuente propia)

Nota: para que el NDS surja efecto hay que esperar unas horas.

Paginas

Las paginas a subir y archivos son los siguientes:

| Nombre | Fecha de modifica... | Tipo | Tamaño |
|-----------------------|----------------------|---------------------|--------|
| css | 06/05/2014 14:20 | Carpeta de archivos | |
| images | 05/05/2014 13:07 | Carpeta de archivos | |
| jQueryAssets | 11/04/2014 11:40 | Carpeta de archivos | |
| paginas | 11/05/2014 10:31 | Carpeta de archivos | |
| index | 01/05/2014 14:33 | Archivo HTML | 7 KB |
| script_acceso_usuario | 11/05/2014 10:59 | Archivo PHP | 2 KB |

Figura 67 Archivos y paginas (Fuente propia)

Hay que tener en cuenta que el nombre de la base de datos, el usuario y la contraseña y acceso al servicio cambia en cada proceso de creación del hosting.

Así que hay que acceder a los archivos:

Script-acceso_usuario, dentro de páginas en el archivo conex.php, en las grafica.php y grafica2.php.

En el caso del Apple hay que comunicarse con el creador ya que este requiere conectarse con la base de datos necesita modificarse además ser certificado.

La base de datos a exportar es de tipo SQL:

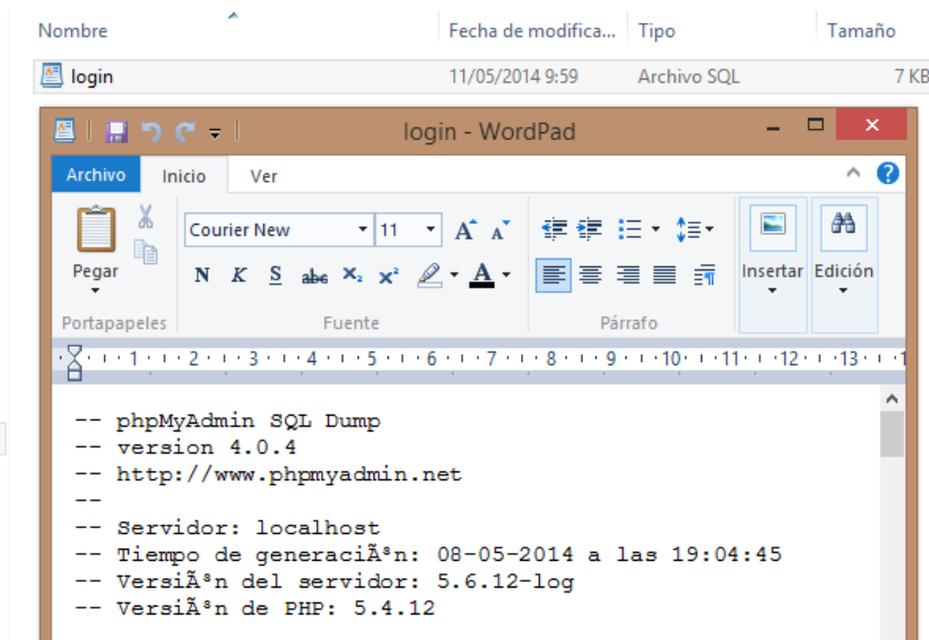


Figura 68 Base de datos (Fuente propia)

Una vez realizado el hosting y el dominio, subido la base de datos y las páginas ya es posible acceder a nuestra aplicación web!!

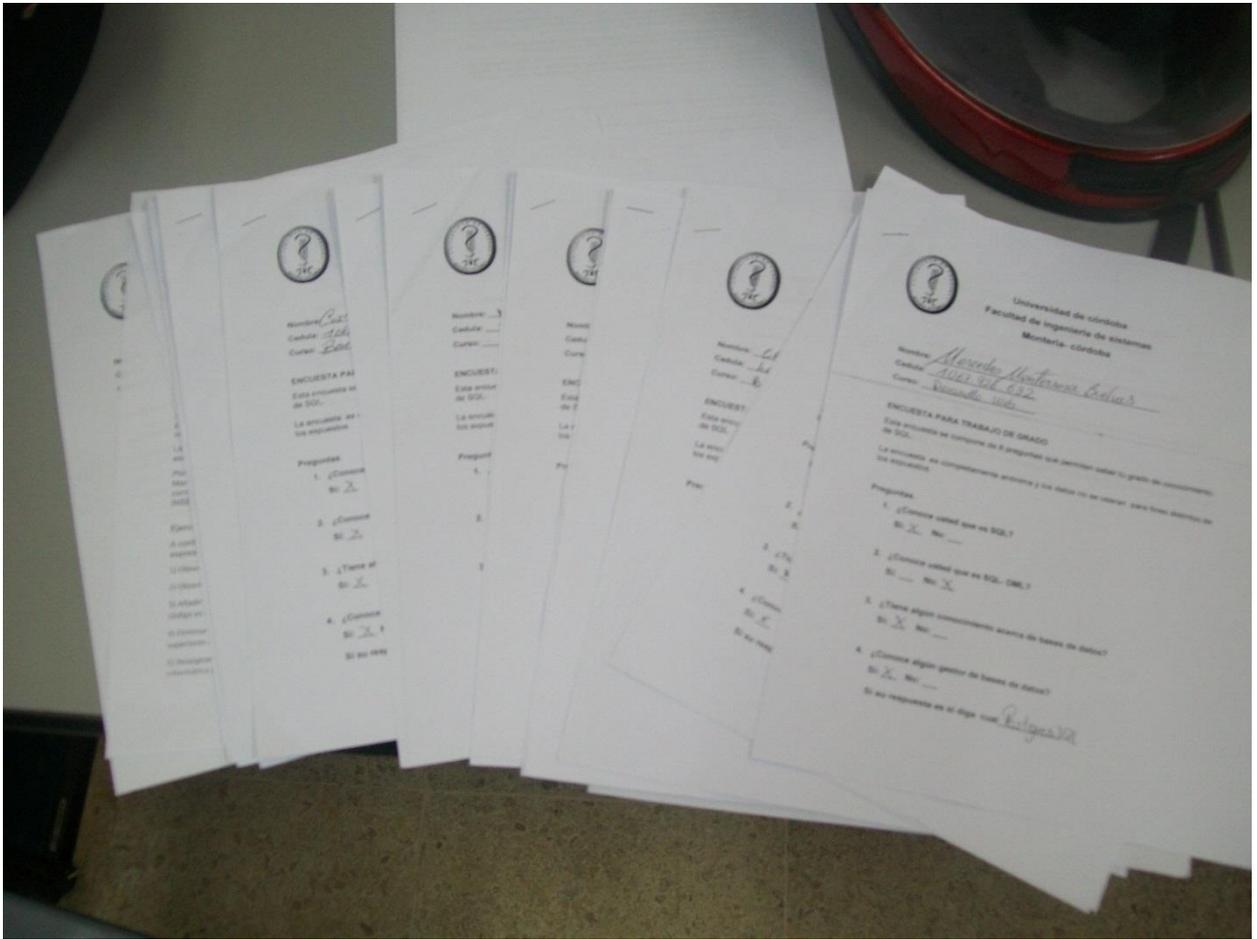
9. EVIDENCIAS

SALA DE INFORMATICA ALUMNOS





ENCUESTAS

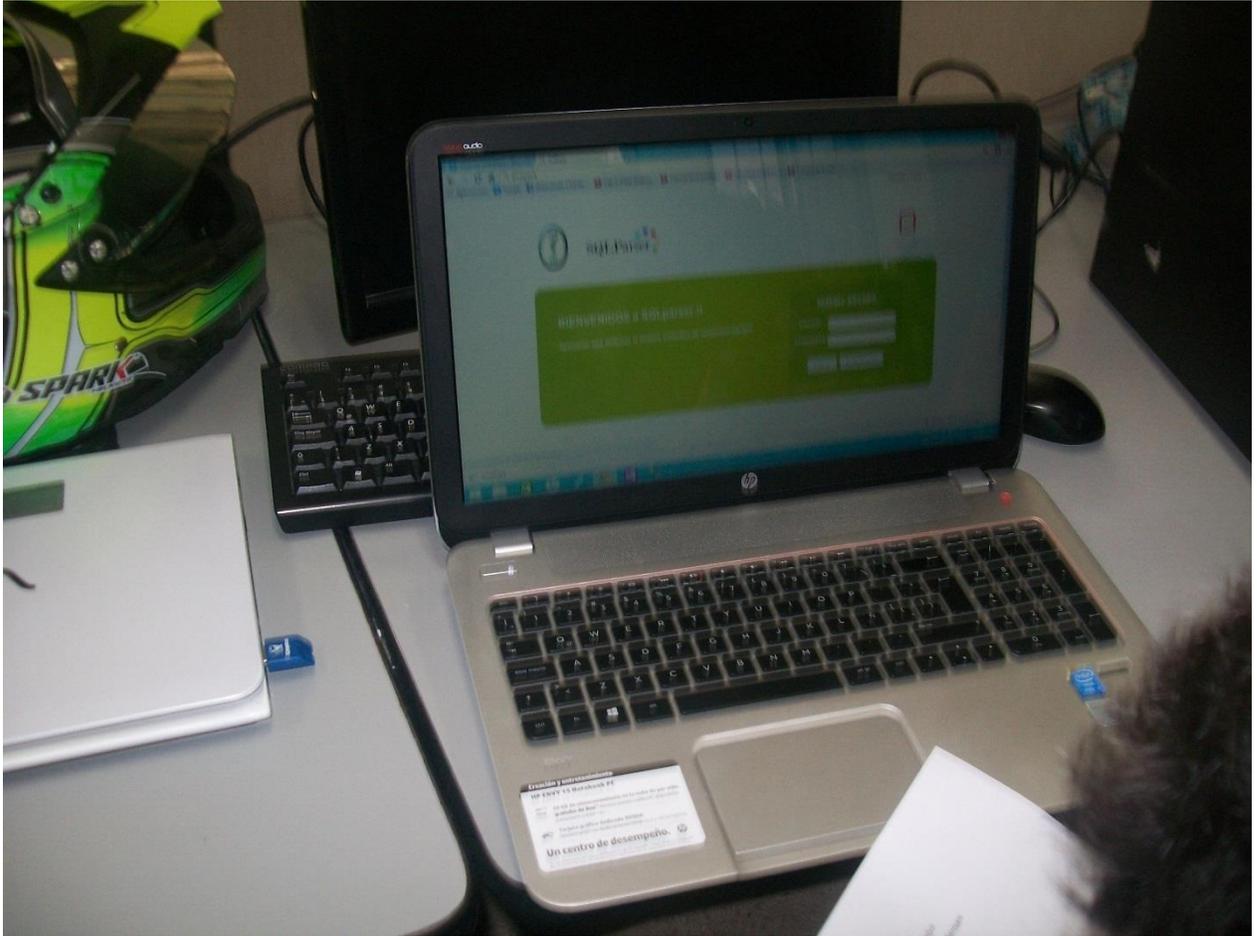


EJERCICIOS DE ACTIVIDAD



INTERACTUANDO CON LA APLICACIÓN







APLICACIÓN

