



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE EVALUACIÓN EN LÍNEA (SIEL) PARA EL ÁREA DE CIRCUITOS ELÉCTRICOS

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTAN:

GONZÁLEZ MEDINA ANA ROSA

HIDALGO AYALA LUIS

JAIMES VERGARA CARLOS ALBERTO

SEVILLA MARTÍNEZ PEDRO RENÉ

VELÁZQUEZ CASTILLO JUAN MANUEL

DIRECTOR DE TESIS:

ING. GLORIA MATA HERNÁNDEZ



CD. UNIVERSITARIA MÉXICO D.F.

2009

DEDICATORIA

Este trabajo de tesis ha sido realizado bajo la dirección de la INE. GLORIA MATA HERNÁNDEZ, Profesor Titular del Departamento de Ingeniería de Control, de la División de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, a quien le damos nuestros más sinceros agradecimientos por su paciencia, apoyo, calidez, sugerencias y confianza para la elaboración de este documento.

Gracias por todo lo que nos ha apoyado e impulsado a seguir adelante en este proyecto, hasta ver cristalizado nuestro sueño.

Sinceramente.

Ana Rosa

Carlos Alberto

Juan Manuel

Luis

Pedro René

AGRADECIMIENTOS

A mi mamá:

Verónica Blanca Molina Aquino

Amor infinito GRACIAS por TODO

Ana Rosa González Molina

Agradecimientos

Son tantas personas a las cuales debo este triunfo, para lograr alcanzar mi culminación académica, misma que es el anhelo de todos los que así lo deseamos.

A mis padres, por darme la vida, guiarme y educarme, por darme estabilidad emocional, económica y sentimental, para llegar hasta este logro, que definitivamente no hubiese podido ser realidad sin ustedes.

GRACIAS por permitirme ser parte de su FAMILIA. Mamá, serás por siempre mi inspiración para luchar por alcanzar mis metas, por enseñarme que todo se aprende y que todo esfuerzo es al final nuestra recompensa. Tu esfuerzo, se convirtió en tu triunfo y el mío, TE AMO.

A mis hermanos Virginia, María del Rocío, Alberto, Francisco Javier y María de Lourdes, porque gracias a esos años maravillosos que vivimos cuando niños, se forjaron en mi grandes sueños, uno de ellos hoy hecho realidad, quiero decirles que ustedes y mis padres son mi única familia de sangre. Gracias María de Lourdes por enseñarme que para vivir no es necesario ver, basta con tener ganas de ser alguien.

A mi esposa María del Carmen, por tu apoyo y comprensión, aunque fue difícil este tiempo por las múltiples ocasiones que tuve que salir a revisar el documento y se que llego a desesperarte, ojalá puedas entenderme.

A mis profesores de primaria, secundaria, preparatoria y de la facultad, porque con sus enseñanzas y esfuerzos, sembraron en mí el deseo de concluir este sueño, gracias encontrarlos en mi vida.

A la Universidad Nacional Autónoma de México, de la que me siento orgulloso de pertenecer y que hoy me brinda esta nueva oportunidad.

A todos mis amigos pasados y presentes, pasados por ayudarme a crecer y madurar como persona y presentes por estar siempre conmigo apoyándome en todo las circunstancias posibles, también son parte de esta alegría, LOS REQUERIDO.

A mi equipo de tesis: Ana Rosa, Pedro René, Carlos Alberto y Juan Manuel, que si bien no fue fácil reunirnos y ponernos de acuerdo, finalmente lo logramos y este documento es el resultado de ello, gracias por contribuir a alcanzar MI SUÑEÑO. SUÑEÑO, que ahora es una realidad.

Gracias a Dios por permitirme ser parte de este tiempo y por cuidar de mi familia y de mí.

Y a todos aquellos, que han quedado en los recintos más escondidos de mi memoria, pero que fueron partícipes en moldear a este su seguro servidor, GRACIAS.

Luis Hidalgo Ayala

Agradecimientos

Dedicatoria

A Emma Vergara Albarrán.

Le doy las gracias por su apoyo incondicional y por todo el tiempo y esfuerzo que ha invertido en cada uno de nosotros para ver alcanzar nuestros triunfos y compartirlos y disfrutarlos de igual manera, como si fueran los propios.

Usted es la mejor y más importantes de mis maestros, pues me ha enseñado a afrontar la vida en cada unos de sus aspectos, gracias a usted soy todo lo que hoy soy.

Gracias Mamá.

A José Carmen Jaimes Gallegos.

Para usted, muestra de que mucho de su trabajo ha fructificado. Gracias por que aún en momentos difíciles se mantuvo firme en apoyarnos y al igual que mi madre, es una de esas dos grandes columnas en donde estoy soportado.

Gracias Papá.

Carlos Alberto Jaimes Vergara



Primeramente agradezco a Dios por haberme dado la vida, haber nacido sano y completo para poder afrontar las diferentes dificultades que tiene la vida a lo largo de su senda.

Agradezco a la Universidad Autónoma de México (UNAM), por la educación que imparte, por haber sido parte de sus estudiantes, de lo cual me enorgullezco, así también a todos los profesores que con sus consejos y sapiencia contribuyeron a ser de mi un mejor alumno y persona.

De una manera muy particular agradezco a mis compañeros de tesis: Ana Rosa, Carlos Alberto, Juan Manuel y Luis, el cual juntos hicimos que este proyecto llegara a su fin de una manera satisfactoria.

Por último agradezco el apoyo incondicional de mis padres a lo largo de mi vida, los cuales también fueron una motivación para la terminación de esta etapa en mi carrera profesional. Así también a todos aquellos familiares que en algún momento de mi vida tuvieron alguna palabra o gesto de apoyo.

Dedico este trabajo a mi madre María Irene y a mi padre Pedro (Q. E. P. D.)

Pedro René Serilla Martínez

Agradecimientos



Juan Manuel Velázquez Castillo

Primero agradecer a Dios que me ha guiado a ser quien soy, y sobre todo a darme la oportunidad de terminar mis estudios con las limitaciones que un estudiante puede tener.....

A mis Padres Goss y Vico dedicarles esta Carrera ya que fueron mi principal Inspiración, ya que ellos no tuvieron la misma oportunidad y decirles que me enorgullece ser su Hijo.....

A mi Papá decirle que el ha sido mi principal ejemplo en la vida y decirle que lo Quiero Mucho.....

A mi Mamá agradecerle por todo su apoyo en las buenas y en las malas, y a pesar de todo decirle que la Quiero Mucho.....

Y sobre todo me gustaría servirles de ejemplo a dos personitas que Quiero y son muy especiales para Mí, a mi Sobrino Joshua y a mi niño que viene en camino.....

GRACIAS.

1.	Introducción.....	3
1.1	Sistemas e-Learning.....	5
1.2	Características del sistema de evaluación en línea (SIEL) en el área de circuitos eléctricos.....	6
2.	Herramientas para el desarrollo del SIEL.....	9
2.1	Apache.....	10
2.2	MySQL.....	12
2.3	PHP.....	13
3.	Análisis estructurado.....	15
3.1	Modelo de proceso unificado.....	16
3.2	Estructura del SIEL.....	22
3.2.1	Modelo de requisitos del SIEL.....	26
3.2.2	Modelo de análisis del SIEL.....	27
3.2.3	Modelo de diseño del SIEL.....	51
4.	Diseño e implementación.....	73
4.1	Modelo Entidad-Relación.....	73
4.2	Base de datos del SIEL.....	76
4.3	Implementación del SIEL.....	90
5.	Procedimientos de metodología de pruebas.....	131
5.1	Pruebas de integración.....	134
6.	Conclusiones.....	145
7.	Anexo A.....	151
7.1	Instalación de herramientas.....	151
8.	Anexo B.....	159
8.1	Código fuente.....	159
	Tabla 4.9.....	159
	Tabla 4.13.....	162
	Tabla 4.17.....	164
	Tabla 4.18.....	165
	Tabla 4.19.....	166
	Tabla 4.23.....	167
	Tabla 4.25.....	168
	Tabla 4.26.....	168
	Tabla 4.27.....	169
	Tabla 4.28.....	169

Tabla 4.30.....	170
Tabla 4.31.....	170
Tabla 4.32.....	170
Tabla 4.35.....	175
Tabla 4.38.....	176
Tabla 4.44.....	178
Tabla 4.45.....	178
Tabla 4.46.....	178
Tabla 4.47.....	179
9. Anexo C.....	181
9.1 Método de Jacobi.....	181
10. Anexo D.....	185
10.1 Hojas de estilo en cascada.....	185
11. Anexo E.....	191
11.1 Instalación del servidor.....	191
12. Mantenimiento y actualización.....	203
13. Glosario de términos.....	207
14. Bibliografía.....	213

1. INTRODUCCIÓN



En un mundo globalizado, donde las tecnologías de la información se encuentran presentes en prácticamente todas las actividades, es de gran importancia involucrarlas al máximo en el ambiente educativo con el fin de apoyar las metodologías de aprendizaje de una manera más sólida.

Si bien existen algunas herramientas de aprendizaje y evaluación a distancia en el mercado, éstas son muy amplias y generales, por lo que es de importancia y gran ayuda para los académicos, contar con un sistema de evaluación en línea y de gestión automatizado a través de un sitio WEB¹, el cual permita a los alumnos, por un lado, conocer su avance en una determinada asignatura a través de reactivos dinámicos; y por otro lado, a los docentes poder supervisar el desempeño académico de sus estudiantes de manera automática, sistemática y oportuna de

¹ Red Informática.

acuerdo con el avance de los temas del curso.

Es por ello que se ha concebido la idea de realizar este trabajo con el objetivo de diseñar e implementar un Sistema de Evaluación en Línea (SIEL), adecuado y eficaz, especialmente para el área de Circuitos Eléctricos.

El desarrollo del SIEL inicia con la investigación de los sistemas actuales de enseñanza y evaluación en línea (*e-learning*), identificando sus características, estándares y especificaciones. Una vez obtenido un panorama amplio de ellos, se detallan las características del Sistema de Evaluación en Línea (SIEL), así como de las herramientas principales necesarias para su desarrollo e implementación como lo son Apache, MySQL y PHP, las cuales se caracterizan por ser software libre, no teniendo restricciones de licencias para su uso.

Posteriormente, se procede a realizar el análisis estructurado del SIEL, señalando los principales bloques o módulos que lo componen, se analiza y describe la Base de Datos (BD), adicionalmente se detallan los procesos necesarios que integran el Sistema explicándolos de una forma más específica, de acuerdo con el análisis estructurado previamente definido.

Se realizan las pruebas y validaciones del SIEL, a través de entradas, salidas y la propia Base de Datos, lo que permite verificar el funcionamiento y desempeño del SIEL.

Finalmente se plasman las conclusiones derivadas de los resultados obtenidos del SIEL en el área de Circuitos Eléctricos al igual que manuales de operación del sistema desarrollado y tutoriales incrustados en el proyecto.

1.1 SISTEMAS E-LEARNING

Hoy en día una de las formas más innovadoras de la educación a distancia son los sistemas *e-learning* que hacen uso de los servicios y facilidades de internet para llevar a cabo el proceso de enseñanza-aprendizaje.

Los sistemas *e-learning* tuvieron su origen en la industria, como resultado de la capacitación basada en computadoras CBT (*Computed Based Training*), actualmente estos sistemas son usados más allá de la capacitación, cubren los modelos de enseñanza-aprendizaje, y se han puesto al alcance de la población en general a través de distintas instituciones educativas tanto públicas como privadas a nivel mundial. Dichos sistemas permiten el acceso de los usuarios a contenidos digitales de formas y formatos diversos, cuya finalidad puede ser tanto la enseñanza como evaluación de determinado contenido temático.

El *software*² que se ha diseñado para llevar a cabo estas tareas de enseñanza-aprendizaje son los LMS (*Learning Management System*), mejor conocidos como Sistemas de Administración de Aprendizaje. Los LMS se instalan en servidores web y consisten en módulos para controlar procesos administrativos que permiten; entre otras tareas, configurar cursos (decidir las materias a las que puede tener acceso el alumno), dar de alta a alumnos, dar de alta a profesores, asignar cursos o evaluaciones a un alumno, llevar reportes del proceso y calificaciones.

El alumno hace uso del sistema a través de un sitio web donde puede tener acceso a las lecciones del curso, realizar actividades programadas, obtener asesoría del profesor e interactuar con otros alumnos, así como dar seguimiento a su propio proceso con datos estadísticos y calificaciones. (Ver figura 1.1).

² ES el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

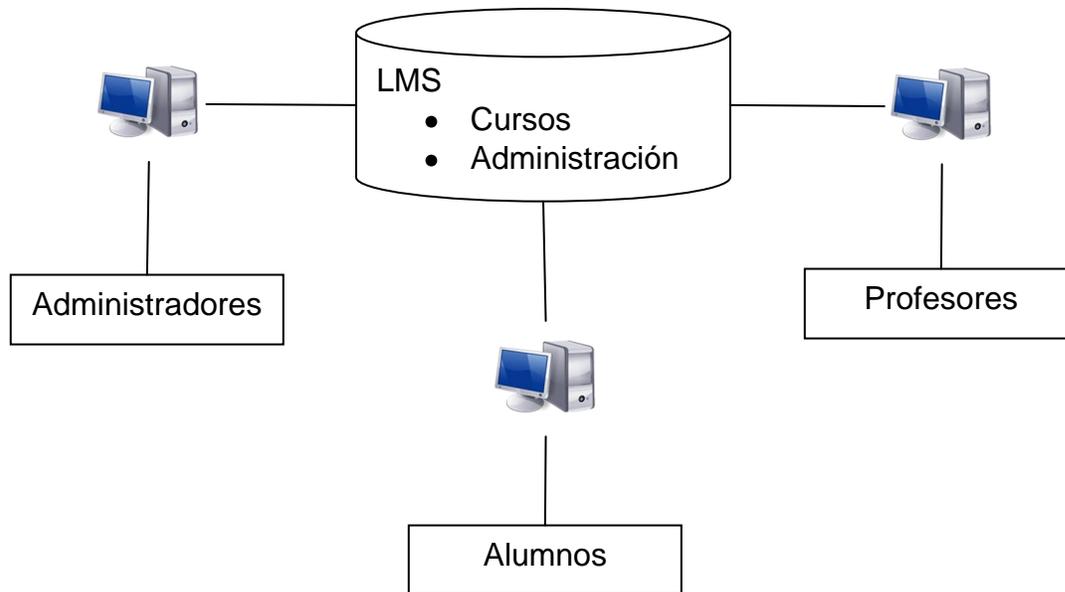


Figura 1.1 Modelo General de un LMS (Sistema de Administración de Aprendizaje).

1.2 CARACTERÍSTICAS DEL SISTEMA DE EVALUACIÓN EN LÍNEA (SIEL) EN EL ÁREA DE CIRCUITOS ELÉCTRICOS

Tomando en cuenta que las herramientas de evaluación que existen actualmente en el mercado son muy amplias, se considera de gran utilidad desarrollar un sistema a través de un sitio web que cumpla con los requerimientos específicos del personal docente; en este caso, el área de Circuitos Eléctricos de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México.

Con base a esto, se ha desarrollado un Sistema de Evaluación en Línea (SIEL), como una herramienta de apoyo para académicos y estudiantes en el área de Circuitos Eléctricos, cuyas características principales son las siguientes:

- Ha sido desarrollado con herramientas y tecnología de libre uso, como lo son PHP, MySQL y Apache, entre otros.

- Se puede consultar en cualquier lugar y a cualquier hora contando con una computadora con conexión a Internet.
- Goza de un método de utilización fácil, pero al mismo tiempo cuenta con acceso restringido y sesiones con límite de tiempo preestablecido.
- Tiene la ventaja de incorporar reactivos dinámicos programados en el sistema que permitan modificar; mediante algoritmos, los parámetros de los problemas propuestos en los reactivos, lo que provee de diferentes ejercicios durante la evaluación.
- El académico cuenta con un sistema de gestión de recursos para conocer el avance de sus estudiantes, a través de cuantificaciones como, índices de desempeño, reportes de evaluaciones, teniendo la libertad de incorporar nuevos reactivos de acuerdo a las necesidades.
- El estudiante puede evaluar su avance en el área de circuitos eléctricos a lo largo del curso.

2.1 APACHE

Apache es un servidor de páginas web de código abierto, multiplataforma y modular, fue creado en el año 1996 y se desarrolló con todo éxito dentro del proyecto HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferencia de HiperTexto) *Server de la Apache Software Foundation*.

La *Apache Software Foundation* tiene su origen en febrero de 1995, fecha en que el software más popular de servidores de HTTP para la web era el desarrollado por Rob McCool, miembro del Centro Nacional para Aplicaciones de Supercomputación (NCSA), de la Universidad de Illinois, el servidor HTTPd³.

El desarrollo del software del servidor de NCSA (software de dominio público y código abierto) tuvo algunos problemas a mediados de 1994, esta circunstancia obligó a que muchos programadores de páginas web que utilizaban esa aplicación tuvieran que desarrollar sus propias extensiones y corregir de forma individual los fallos en el funcionamiento de la aplicación original.

Un pequeño grupo de aquellos desarrolladores, entró en contacto vía correo electrónico, con el objeto de coordinar y juntar sus trabajos de corrección para mejorar la aplicación original de NCSA. Fueron Brian Behlendorf y Cliff Skolnick quienes coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores, en una computadora instalada en California. Este software consolidado Apache.

El nombre “Apache” fue elegido por el respeto que se tiene a los nativos indios americanos de la tribu Apache, bien conocidos por sus excelentes habilidades en la guerra y su estrategia de resistencia. También se hace referencia a un juego de

³ (HTTP daemon). Programa que corre de fondo en un servidor Web y espera peticiones de entrada para responderles.

palabras sobre “*a patchy web server*” (un servidor a partir de una serie de parches), sin embargo este no es su origen.

Apache, al igual que todos los servidores web, es la interfaz entre el usuario web que pide una URL (*Uniform Request Locator*, Localizador Uniforme de Recursos) y el sistema donde se encuentra ese servicio.

El servidor y el usuario web se comunican gracias al protocolo HTTP (*Hypertext Transfer Protocol*). Al visualizar una página web hay una serie de eventos que ocurren, muchos de los cuales no son tangibles para nosotros, pero que debemos considerar, estos eventos empiezan a ocurrir en el lado del usuario, mismos que inician con una petición que es enviada al servidor y termina con la respuesta de visualizar la página en su navegador. Dichos eventos podemos relacionarlos conforme a la siguiente lista: (Ver figura 2.1).

- La computadora del usuario se conecta a Internet.
- El usuario abre un navegador.
- El usuario hace una petición de una página web. Al hacer esto, un mensaje es enviado a un servidor de nombres, y el servidor de nombres direcciona al usuario al servidor que hospeda la página web.
- El servidor que hospeda la página recibe la petición y regresa la página pedida.
- Si la página es dinámica, es decir, que contiene *scripts* (archivos de texto con alguna programación), como PHP, el servidor compila los *scripts* con un compilador que genera HTML.
- El servidor entonces envía la página HTML generada al navegador.
- El navegador recibe el HTML y despliega la interpretación de la página.

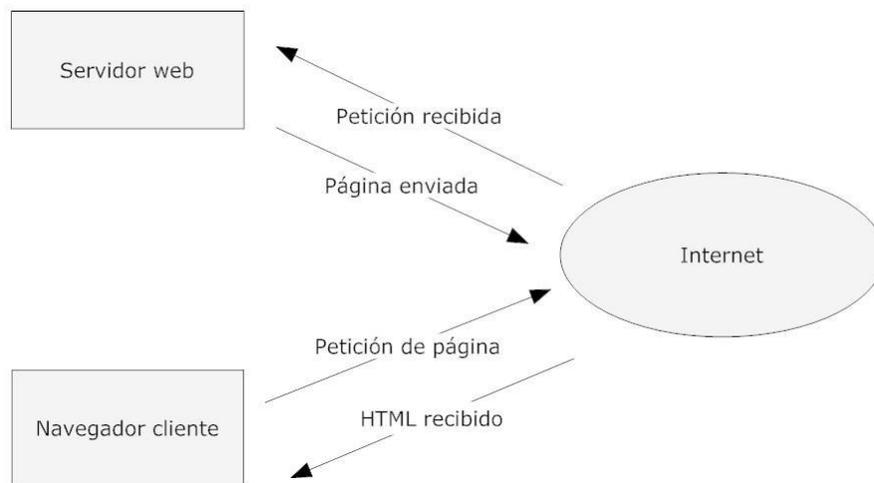


Figura 2.1. Proceso usuario/servidor

2.2 MySQL

MySQL⁴ es un sistema administrador de bases de datos, que archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo, esto permite que se tenga mayor velocidad y flexibilidad en el acceso a la información. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas conforme se requiera.

Dado que las computadoras nos permiten el manejo de grandes cantidades de información, los administradores de bases de datos como MySQL, juegan un papel central en la computación, como aplicaciones independientes o como parte de otras aplicaciones.

Structured Query Language, (SQL: Lenguaje de Consulta Estructurado) es comercializado por vez primera en el año de 1981 por IBM, y fue presentado al *American National Standards Institute* (ANSI: Instituto Nacional Americano de Estándares), por ser considerado un estándar de las bases de datos relacionales.

⁴ MySQL es una idea originaria de la empresa opensource [MySQL AB](#).

MySQL es multiplataforma, maneja diferentes opciones de almacenamiento según se requiera, velocidad en las operaciones o el mayor número de operaciones disponibles, tiene transacciones y claves foráneas, conectividad segura, replicación y búsqueda e indexación de campos de texto, entre sus principales características.

2.3 PHP

PHP (*Hypertext Pre-Processor*, Pre-procesador de Hipertexto) es un lenguaje de programación, concebido principalmente como herramienta para el desarrollo de aplicaciones web. PHP permite diseñar páginas dinámicas de servidor, es decir, generar páginas bajo petición; capaces de responder de manera inteligente a las demandas del cliente y a la automatización de diversas tareas.

PHP es un software de distribución libre y multiplataforma, comenzó siendo un conjunto de instrucciones que permitían a su creador, Rasmus Lerdorf, el control de los accesos a sus páginas personales. A este conjunto de scripts les denominó como PHP (*Personal Home Page tools*). Poco a poco Rasmus fue completando las funcionalidades básicas de su herramienta escribiendo programas en lenguaje C. En 1995 decidió liberar el código fuente para que cualquiera pudiera utilizarlo y colaborar en su mejora, de este modo nació PHP/FI. A finales de 1997 se libera PHP/FI 2.0, pasando de ser el proyecto de una sola persona al desarrollo de un equipo.

A diferencia de otros lenguajes de programación como Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado, normalmente una página HTML, enviada al navegador. (Ver Figura 2.2.)



Figura 2.2. Ejecución de un programa en PHP

3. ANÁLISIS ESTRUCTURADO



El Sistema de Evaluación en Línea (SIEL) es una herramienta para el área de Circuitos Eléctricos; que permite, por un lado presentar exámenes que el sistema evalúa para conocer el avance de los alumnos y fortalecer los temas que pudieran no estar ampliamente comprendidos; y por otro, sirve a los docentes como herramienta para conocer el desempeño de sus estudiantes al obtener, accesos, estadísticas, reportes y calificaciones. (Ver figura 3.1).

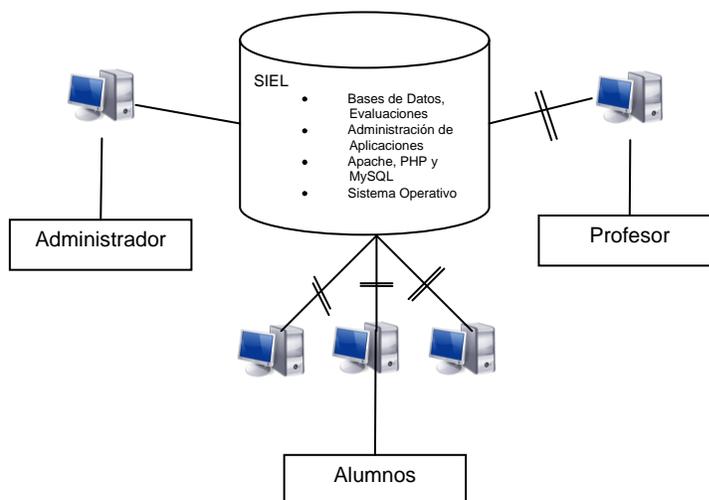


Figura 3.1. Concepción del SIEL como un sistema e- Learning.

3.1 MODELO DE PROCESO UNIFICADO

El proyecto se llevo a cabo bajo la metodología del Proceso Unificado para hacer el análisis del SIEL, por estar orientada a objetos, que cuenta con etapas comunes a la mayoría de las metodologías que existen: análisis, diseño, implementación y pruebas. La diferencia consiste en el enfoque que se da para abordar estas etapas; en algunas metodologías llamadas modelos. Los Casos de Uso, son la base para especificar los requerimientos del sistema en el que se centra el Proceso Unificado, presentan una forma accesible para iniciar la definición del sistema, esta es una de las causas por la cual se eligió esta metodología.

La metodología del Proceso Unificado, se compone de tres modelos: de requisitos, de análisis y de diseño. En el modelo de requisitos se enuncian los Casos de Uso y los actores, en el modelo de análisis se amplían dichos Casos de Uso y se describen los objetos para posteriormente; en el modelo de diseño, realizar los diagramas de secuencia y las tarjetas de objetos. (Ver figura 3.2).

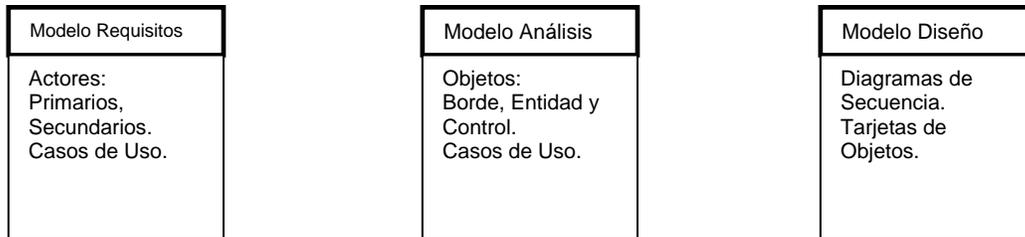


Figura 3.2. Modelo del Proceso Unificado

El modelo de requisitos, se basa en la descripción de requerimientos de un sistema mediante Casos de Uso; y consta de tres ejes principales, visualmente representado por un diagrama de tres dimensiones. (Ver figura 3.3).

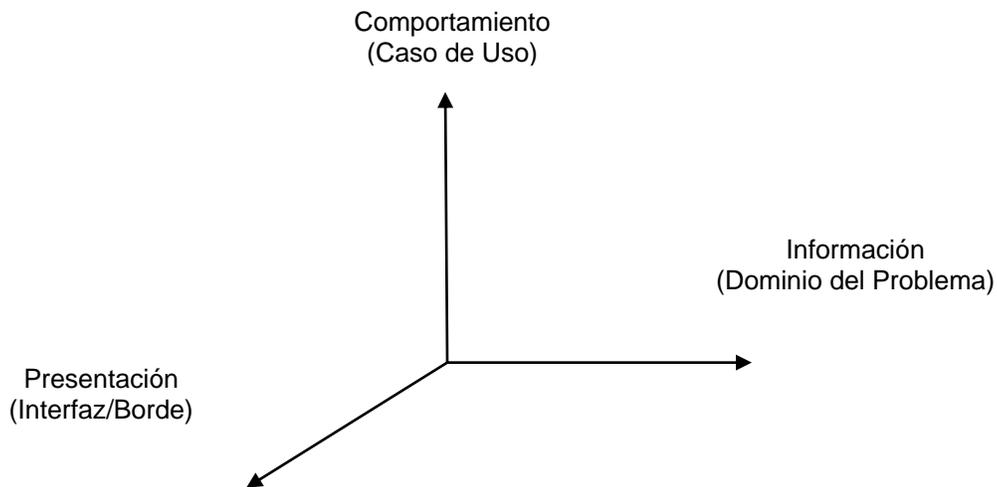


Figura 3.3. Los tres ejes del modelo de requisitos.

El eje de comportamiento basado directamente en el modelo de Casos de Uso, muestra la funcionalidad que tiene el sistema desde el punto de vista del usuario.

El eje de presentación, interfaces o bordes; detalla como interactúa el sistema con actores externos al ejecutar los Casos de Uso.

El eje de información o de dominio del problema, explica los aspectos estructurales del sistema. Permite separar e identificar rápidamente los objetos que pueden ser guardados en una Base de Datos. También puede ser usado como un parámetro de llamadas entre funciones. El propósito general del dominio del problema, en el modelo de requisitos, es formar una base común de entendimiento de desarrollo. Se deben reconocer todos los conceptos, para formar una terminología común al razonar sobre los Casos de Uso y disminuir los malos entendidos entre el analista y el usuario.

Dentro de los Casos de Uso la entidad básica es el actor y cada Caso de Uso es una secuencia de eventos iniciada por el usuario.

Los usuarios son las personas que usan el sistema, mientras que los actores son entidades dentro del sistema, las cuales permiten que éste funcione llevando a cabo tareas específicas; dentro de este marco, una base de datos es considerada como un actor secundario.

En el SIEL se tienen usuarios y actores con el mismo nombre, sin embargo, son distintos conceptos, ya que los usuarios administradores, profesores y alumnos, son las personas que hacen uso del sistema y los actores administrador, profesor y alumno son entidades con diversas funciones, procesos y tareas específicas dentro del sistema.

Un Caso de Uso representa un conjunto de eventos, los cuales son realizados cuando el usuario interactúa con el sistema para realizar una actividad específica a través de las tareas, funciones y procesos realizados entre los diversos actores. El modelo de Casos de Uso permite la subdivisión de los mismos en subflujos.

En el modelo de análisis, es necesario declarar los tipos de objetos que intervienen en los Casos de Uso (borde, entidad y control), para llevar a cabo la implementación, los diferentes actores que se describen en el modelo de requisitos reciben el nombre de objetos en el modelo de análisis.

Borde: Son objetos que se encargan de describir la relación bidireccional que existe entre los actores y el sistema, por un lado interpreta los eventos realizados por el actor en eventos comprendidos, y por otro los eventos ejecutados por el sistema, son desplegados de manera que puedan ser interpretados por otro actor. Existe dos tipos principales de Bordes, los que se comunican con otros sistemas (interfaces) y los que se comunican con el usuario (pantallas). El propósito de este objeto es presentar gráficamente toda la funcionalidad del Caso de Uso al usuario.

Entidad: Son objetos que están ligados a la información que se guarda en el sistema a través de las bases de datos, y que es usada en las diferentes pantallas donde el usuario interactúa con el sistema.

Control: Son objetos asignados por cada Caso de Uso, y se encargan de manejar los eventos asociados a éste, así como su relación con las diferentes entidades.

Cuando se tienen identificados los objetos que integran el sistema (Borde, Entidad y Control), el siguiente paso es graficar el Diagrama de Secuencia también conocido como de interacción o eventos, el cual consiste en la secuencia detallada de cómo interactúan dichos objetos para llevar a cabo la funcionalidad del Caso de Uso.

Los diagramas de secuencia describen los flujos principales así como los subflujos de los Casos de Uso, debe de haber una fluidez de los eventos dentro de la arquitectura de análisis, correspondiente a cada Caso de Uso. Dicha secuencia de

eventos se da en el dominio del tiempo, en donde el tiempo avanza hacia abajo; estos diagramas de secuencia son utilizados en el modelo de diseño.

El diseño de objetos consiste en ampliar el análisis hecho con anterioridad añadiendo detalles para lograr una interfaz preliminar antes de comenzar la implementación final del sistema.

Para crear un objeto se seguirá el diseño por responsabilidades (RDD – *Responsability-Driven-Design*). Este diseño se basa en el modelo cliente-servidor, donde los objetos se ven como clientes cuando generan alguna petición hacia otro objeto, y como servidores cuando reciben peticiones de otro objeto. De tal manera, un mismo objeto en distintos momentos se ve como cliente o servidor.

La funcionalidad ofrecida por los objetos servidores se define en términos de sus responsabilidades, las cuales se satisfacen al lograr sus servicios con los demás objetos. Los servicios y responsabilidades corresponderán, finalmente a los métodos que se implementarán en ese objeto. A su vez, los objetos servidores pueden tener colaboraciones con otros objetos, para lograr la satisfacción de responsabilidades que por sí solas no se logran. Como consecuencia de esto se integran las responsabilidades y colaboraciones entre los objetos para definir contratos, los que definen la naturaleza y alcance de las interacciones cliente-servidor. Los contratos y colaboraciones representan los requisitos de servicios entre los objetos.

En el diseño de responsabilidades, se identifican los contratos entre los objetos cliente y servidor, diseñando su distribución entre los distintos objetos del sistema. (Ver figura 3.4).

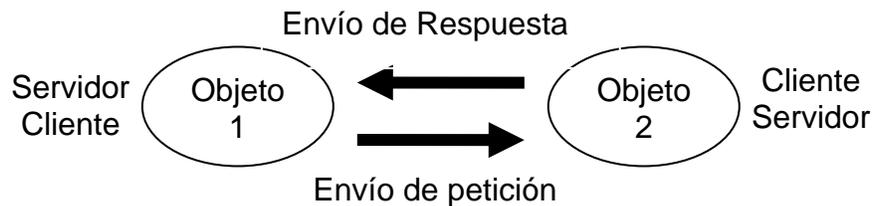


Figura 3.4. Modelo cliente servidor

En el diseño de responsabilidades se identifican varios términos:

- **Tarjetas de Clases:** También conocidas como tarjetas CRC (Clase-Responsabilidad-Colaboración), permiten al diseñador visualizar las diferentes clases de manera independiente y detallada. Para nuestro caso usaremos las tarjetas para describir las páginas usadas en los Casos de Uso, los cuales por convención serán referenciados como objetos.
- **Responsabilidades:** en general, el comportamiento de la clase no debe descomponerse directamente en operaciones, sino seguir un procedimiento más natural comenzando con una descripción verbal de las responsabilidades o función que cumple cada clase u objeto dentro del sistema. Las responsabilidades se identifican a partir de los Casos de Uso generados durante el modelo de análisis, en donde se determina cuáles representan las acciones que deba ejecutar algún objeto dentro del sistema.
- **Colaboraciones:** Representan solicitudes de un objeto cliente a un objeto servidor. Los objetos pueden desempeñar el papel de clientes o servidores, dependiendo de su actividad en ese momento. Durante el proceso de identificación de las colaboraciones se toma la tarjeta de clase, escribiendo en la columna derecha el nombre de la clase servidor, o sea, la clase que colabora para satisfacer la necesidad del cliente.
- **Contratos:** Entre dos clases u objetos representa una lista de servicios que un objeto solicita a otro objeto. Todos los servicios especificados en un contrato particular son la responsabilidad del servidor para ese contrato. En cada tarjeta de clase se divide la sección de responsabilidades en dos. En la parte superior

se especifica una sección para los contratos, en tanto que en la parte inferior, se especifica una sección para las responsabilidades privadas. Cada contrato debe incluir nombre y número de contrato. Se debe listar cada contrato para el cual este objeto es un servidor.

Algoritmos: Definen la lógica utilizada por cada operación para resolver la responsabilidad a la que corresponden. Los algoritmos se especifican de manera declarativa o de procedimiento, dependiendo de su complejidad. Un algoritmo declarado de manera declarativa, se hace mediante una especificación textual, en cuyo caso se usa algún conocimiento adicional para implementar el algoritmo. Por su parte en el algoritmo que se explica por procedimiento, se hace mediante diagramas de flujo.

3.2 ESTRUCTURA DEL SIEL

El SIEL, permite al alumno realizar una evaluación de los conocimientos adquiridos en el área de Circuitos Eléctricos, por medio de una serie de reactivos los cuales están divididos según su complejidad, en Básicos y Avanzados. Una vez realizado el examen los resultados son guardados en una Base de Datos, para su consulta posterior. El alumno tiene la posibilidad de realizar dos intentos para cada nivel de examen.

Por otro lado, se permite a los profesores dar de alta grupos y alumnos. Se cuenta también con un usuario tipo Administrador que puede dar de alta a profesores y a otros Administradores.

El sistema cuenta con una pantalla de inicio, donde el usuario (Administrador/Profesor/Alumno) ingresa el nombre de usuario y contraseña que le corresponden para que el sistema lo autentifique y éste pueda ingresar al sistema en el módulo correspondiente.

- Módulo Administrador
- Módulo Profesor
- Módulo Alumno

Cabe mencionar que el examen se concibe en un módulo aparte, dentro del módulo Alumno. Para cada uno de estos módulos se cuenta con las siguientes actividades o acciones.

Módulo Administrador

- Alta Administrador/Profesor
- Consulta de información Administrador/Profesor
- Baja Administrador/Profesor de la BD

Módulo Profesor

- Alta grupo/Alumno/Reactivo de examen
- Consulta de información del Alumno
- Consulta de resultados del examen

Módulo Alumno

- Selección y realización de Examen Básico/Avanzado
- Consulta de reportes de examen
- Cambio de contraseña

Módulo Examen.

- Este módulo se encargará de seleccionar los reactivos de acuerdo al valor en puntos, para posteriormente comparar los resultados del sistema con los del alumno y determinar si la respuesta del reactivo fue correcta y así poder determinar la calificación obtenida por el alumno. Esta información es guardada en la Base de Datos para su posterior consulta.

A continuación se muestra el Diagrama General de concepción del SIEL:

Sistema de Evaluación en Línea en el área de Circuitos Eléctricos

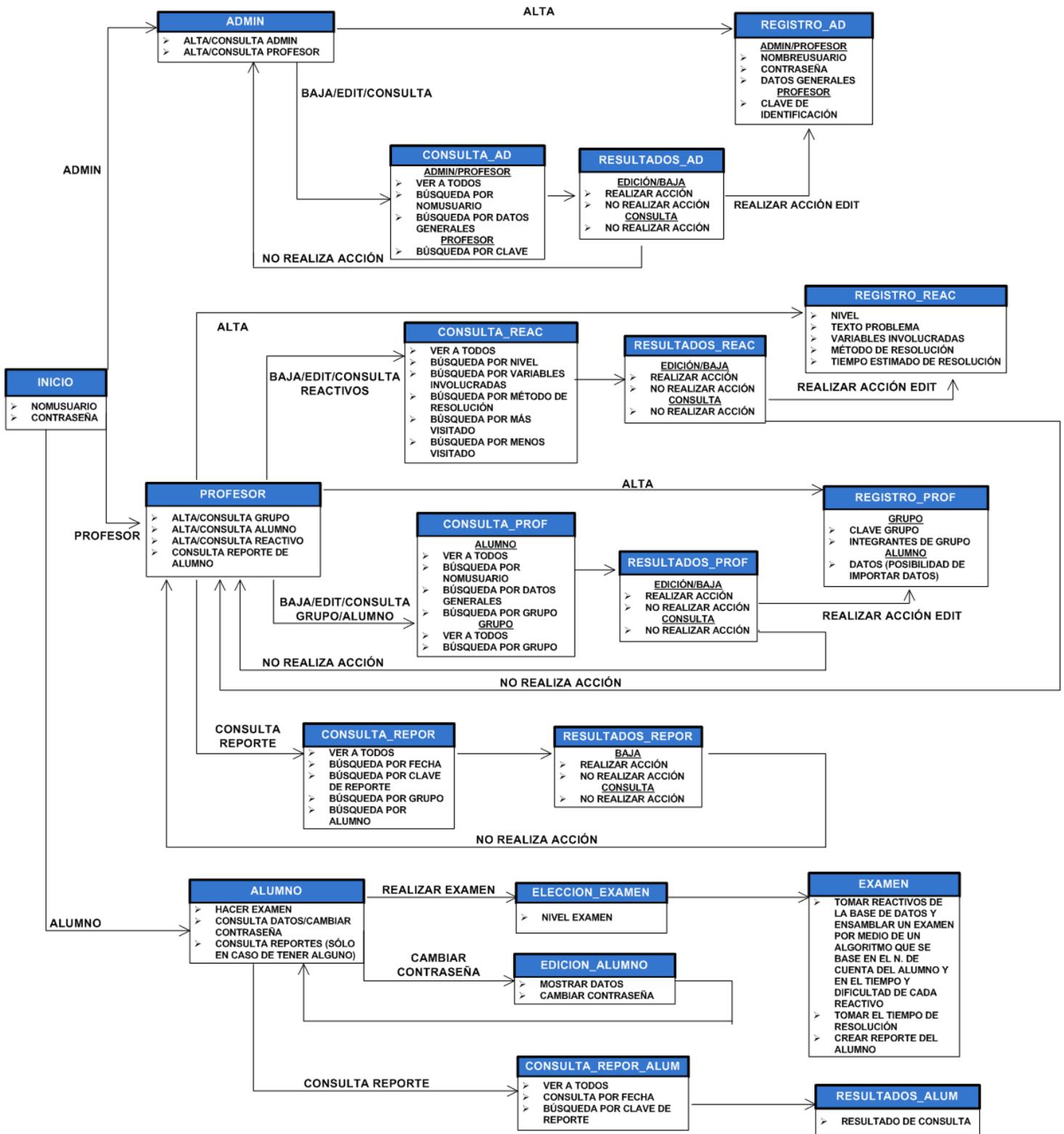


Figura 3.4. Diagrama General del SIEL.

Los criterios de búsqueda de información en los diferentes módulos están basados en los campos como nombre, tipo de usuario (Administrador/Profesor/Alumno), grupo, etc. Una vez que se muestren los resultados de la consulta el usuario puede, confirmar la información del registro, modificarla o borrarla.

3.2.1 MODELO DE REQUISITOS DEL SIEL

El SIEL cuenta con una Base Datos, entradas, salidas y procesos que permiten su funcionamiento, los actores principales dentro del SIEL son Alumno, Profesor y Administrador, dependiendo de los atributos otorgados por el sistema, cada uno de ellos es dirigido a la sesión correspondiente, en donde tienen acceso a módulos, tareas y actividades específicas.

Como se menciona anteriormente, el eje de información o de dominio del problema, explica los aspectos estructurales del sistema, como lo son los objetos y atributos. (Ver tabla 3.1).

Tabla 3.1. Objetos y Atributos del Dominio del Problema del SIEL.

Objetos	Atributos
Alumno / Profesor / Administrador	Nombre
RegistroUsuario	Número de Cuenta, Nombre, Usuario, Contraseña, Tipo.
RegistroReactivo	Diagrama de pregunta, texto de pregunta, Nivel examen, Puntos.
RegistroGrupo	Clave Profesor, Número de Grupo.
Reporte	Número de Cuenta Alumno, id de reactivos, respuesta del sistema, respuesta del alumno, calificación, fecha de realización, Duración.
Examen	Básico, Avanzado.
Tipos de Reactivos	Fijos, Dinámicos.

3.2.2 MODELO DE ANÁLISIS DEL SIEL

En el SIEL se tienen los actores principales Administrador, Profesor y Alumno, además se cuenta con actores secundarios como la base de datos.

El actor Administrador tiene la función de dar de alta, baja o editar; Profesores o Administradores, así como realizar consultas de los Profesores y Administradores.

El actor Profesor puede dar de alta a un Alumno/Grupo/Reactivo de Examen, dar de baja a un Alumno/Grupo/Reactivo de Examen, consultar información del Alumno, así como de los resultados del examen que realizaron los alumnos.

El actor Alumno puede elegir entre hacer un Examen Básico /Avanzado, consultar reporte de los exámenes realizados, así como cambiar la contraseña.

Teniendo en cuenta el concepto de actor, se identifican los siguientes actores dentro del sistema. (Ver tablas 3.2 a 3.5).

Tabla 3.2. Actor Administrador

Actor	Administrador.
Casos de Uso	Validar Usuario, Registrar Administrador, Registrar Profesor, Consultar Información Profesor/Administrador, Dar Baja Administrador/Profesor
Tipo	Primario
Descripción	Es uno de los actores primarios y es el encargado de las gestiones administrativas de los Administradores y Profesores.

Tabla 3.3. Actor Profesor

Actor	Profesor.
Casos de Uso	Validar Usuario, Registrar Alumno, Registrar Grupo, Registrar Reactivo, Consultar Información Alumno/Profesor, Dar Baja Grupo/Alumno/Reactivo.
Tipo	Primario
Descripción	Es uno de los actores primarios y es el encargado de dar de alta los reactivos de los exámenes, a alumnos dentro del sistema, para que puedan realizar el examen y consultar los reportes de los exámenes.

Tabla 3.4. Actor Alumno

Actor	Alumno.
Casos de Uso	Validar Usuario, Realizar examen Básico/Avanzado, Consultar Reporte de Exámenes.
Tipo	Primario
Descripción	Es uno de los actores primarios y su función principal es la de realizar el examen para ver su avance en la materia de Circuitos Eléctricos.

Tabla 3.5. Actor Base de Datos

Actor	Base de Datos
Casos de Uso	Almacenar Usuarios (Administrador/Profesor/Alumno), Dar de Baja Usuarios (Administrador/Profesor/Alumno). Almacenar Reactivos fijos Recuperación de Reactivos dinámicos Almacenar Reportes
Tipo	Secundario
Descripción	Es un actor secundario y representa la Base de Datos donde es almacenada la información de los usuarios, los reactivos a elegir del examen y los datos necesarios para los reportes.

El módulo de Inicio es el encargado de validar al usuario que desea ingresar al sistema y dependiendo de su tipo (Administrador/Profesor/Alumno), éste es dirigido al módulo correspondiente para realizar las actividades inherentes a su perfil con las herramientas que se presentan en el SIEL. (Ver figura 3.5).

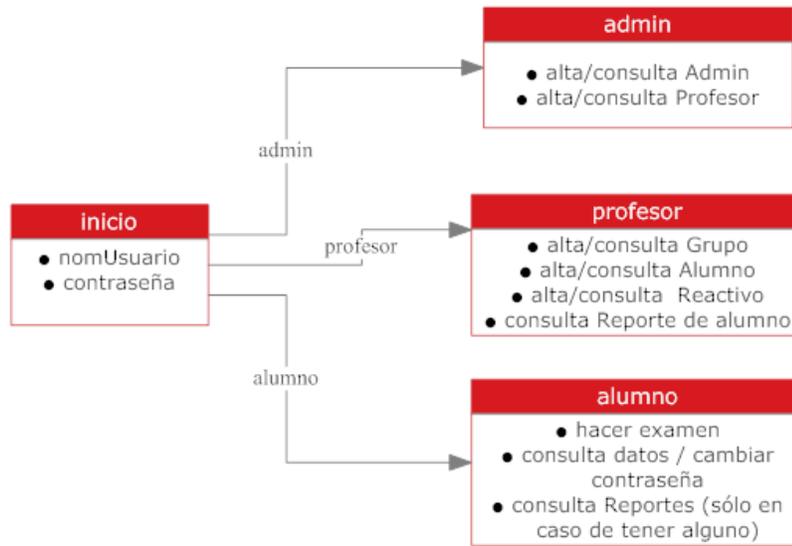


Figura 3.5. Diagrama de módulo de inicio.

Casos de Uso INICIO:

Tabla 3.6. Casos de Uso del módulo Inicio.

Caso de Uso	Autenticar Usuario
Actores	Usuario (Administrador/Profesor/Alumno), Base de Datos
Propósito	Validar a un usuario dado de alta con anterioridad para hacer uso del sistema (SIEL).
Resumen	El usuario proporciona los datos de usuario y contraseña, para que estos sean autenticados con la tabla de usuario correspondiente, para que pueda entrar al sistema y dependiendo de su tipo (Administrador/Profesor/Alumno) se direcciona al módulo correspondiente.
Precondiciones	Se requiere haber ejecutado anteriormente el Caso de Uso

	Registrar Administrador, Registrar Profesor o Registrar Alumno con el subflujo Dar de Alta. (Administrador/Profesor/Alumno).
Flujo Principal	Se presenta la pantalla principal donde el usuario proporcionará los datos de usuario y contraseña, para que sean autenticados y éste pueda ingresar al sistema, al presionar la actividad seleccionada de ingresar.
Subflujos	
Excepciones	E-1, no se autentifico el usuario, se le solicita al usuario ingresar nuevamente los datos.

El módulo de administrador permite al usuario llevar a cabo las siguientes tareas:
(Ver figura 3.6).

- Dar de alta a otro usuario con nivel de administrador o profesor.
- Dar de baja a un usuario con nivel administrador o profesor.
- Tener acceso a datos de profesores o administradores para realizar consultas.

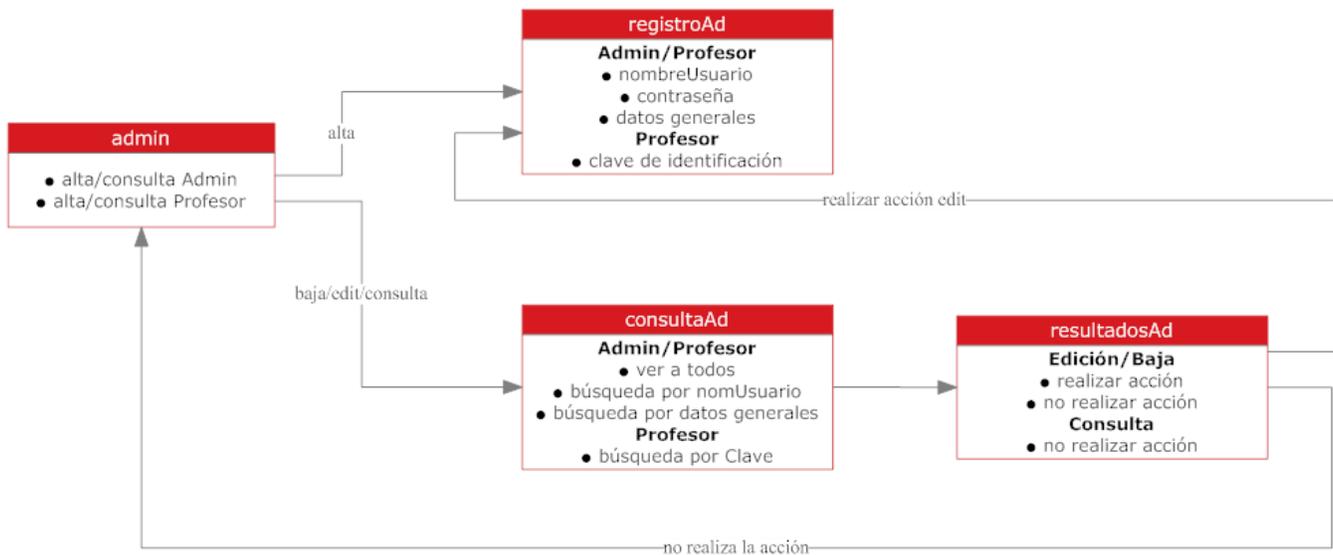


Figura 3.6. Módulo de Administrador.

Casos de Uso ADMINISTRADOR:

Tabla 3.7. Funciones de Administrador

Caso de Uso	Funciones de Administrador
Actores	Administrador, Base de Datos.
Propósito	Conectar los diferentes tipos de procesos para el Administrador.
Resumen	Son usados los diferentes procesos para la Administración de la materia de Circuitos Eléctricos.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado.
Flujo Principal	Se tiene la pantalla de Modulo de Administrador, donde continúan los subflujos de este Caso de Uso. Se presiona la actividad seleccionada la cual puede ser: Alta Administrador, Alta Profesor, Consulta Administrador, Consulta Profesor, Baja Administrador, Baja Profesor, Modificar Administrador y Modificar Profesor.
Subflujos	S-1 Dar de Alta Administrador. S-2 Dar de Alta Profesor S-3 Dar de Consulta de Administrador S-4 Dar de Consulta de Profesor S-5 Dar de Baja Administrador S-6 Dar de Baja Profesor

Tabla 3.7. Alta Administrador

Caso de Uso	Dar de alta Administrador
Actores	Administrador, Base de Datos.
Propósito	Dar de alta en el sistema a un nuevo administrador.
Resumen	El usuario inicia este Caso de Uso. Ofrece la funcionalidad de dar de alta a otro administrador.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado.

Flujo Principal	<p>Se tiene la pantalla de ingreso de datos, se ingresan los datos correspondientes, se continúa con los subflujos de este Caso de Uso.</p> <p>Se presiona la actividad seleccionada enviar, para que los datos sean validados para la correcta inserción del registro.</p> <p>Si la actividad seleccionada es volver, se regresa al Caso de Uso funciones del administrador.</p>
Subflujos	<p>S-1 Dar de alta al administrador.</p> <p>Se presenta la pantalla al administrador “Dar de alta administrador”, se tienen los campos que deben ser llenados por el administrador, la cual incluye nombre, usuario y contraseña.</p> <p>El administrador puede seleccionar las siguientes actividades, “Enviar” y “Volver”.</p> <p>Si el administrador selecciona “Enviar”, el sistema genera un registro nuevo. Se continúa con el subflujo Comprobar Registro (S-2).</p> <p>Si el administrador selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Comprobar Registro.</p> <p>Se presenta la página “Datos del Registro Insertado”</p> <p>Dicha página se encarga de validar los datos, provenientes de la página de “Dar de alta al administrador” (E1, E2).</p> <p>Se tiene la actividad “Volver”, la cual se encarga de regresar al Caso de Uso Funciones del administrador.</p>
Excepciones	<p>E1 Información incompleta.</p> <p>E2 Registro ya existe.</p> <p>Para ambas excepciones. Se presenta la página “Datos del Registro Insertado”. Se presenta el mensaje de error. Se tiene la actividad “Volver”, la cual permite regresar al Caso de Uso Funciones del Administrador.</p>

Tabla 3.8. Alta Profesor

Caso de Uso	Dar de Alta Profesor
Actores	Administrador, Base de Datos
Propósito	Dar de alta en el SIEL a un nuevo Profesor
Resumen	Ofrece la funcionalidad de dar de Alta a un Profesor, para así determinar el grupo. Un profesor puede tener varios grupos.
Precondiciones	Haber ejecutado el Caso de Uso, validar al Usuario y haber sido autenticado.
Flujo Principal	<p>Se tiene la pantalla de ingreso de datos, se ingresan los datos correspondientes, se continúa con los subflujos de este Caso de Uso.</p> <p>Se presiona la actividad seleccionada enviar, para que los datos sean validados para la correcta inserción del registro.</p> <p>Si la actividad seleccionada es “Volver”, se regresa al Caso de Uso funciones del administrador.</p>
Subflujos	<p>S-1 Dar de Alta al Profesor.</p> <p>Se presenta la pantalla al administrador “Dar de alta Profesor”, se tienen los campos que deben ser llenados por datos del Profesor, la cual incluye número de cuenta, apellido paterno, apellido materno, nombre, usuario , contraseña, Tipo.</p> <p>El Administrador puede seleccionar las siguientes actividades, “Enviar” y “Volver”.</p> <p>Si el administrador selecciona “Enviar”, el sistema genera un registro nuevo. Se continúa con el subflujo Comprobar Registro (S-2).</p> <p>Si el administrador selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Comprobar Registro.</p> <p>Se presenta la página “Datos del Registro Insertado”</p> <p>Dicha página se encarga de validar los datos, provenientes de la página de “Dar de alta al Profesor” (E1, E2).</p>

	Se tiene la actividad "Volver", la cual se encarga de regresar al Caso de Uso Funciones del administrador.
Excepciones	E1 Información incompleta. E2 Registro ya existe. Para ambas excepciones. Se presenta la página "Datos del Registro Insertado". Se presenta el mensaje de "Error". Se tiene la actividad "Volver", la cual permite regresar al Caso de Uso Funciones del Administrador.

Tabla 3.9. Consulta de Administrador

Caso de Uso	Consultar Información Administrador
Actores	Administradores, Base de Datos
Propósito	Consulta de información de los usuarios (Administrador)
Resumen	Ofrece la funcionalidad de proporcionar la información detallada de Administradores.
Precondiciones	Haber sido validado al usuario Administrador y haber registros de Administradores.
Flujo Principal	Se tiene la pantalla de datos de los usuarios (Administrador). Se presiona la actividad de "Volver" y se regresa al Caso de Uso funciones del administrador.
Subflujos	S-1 Consulta de Administrador. Se presenta la pantalla al administrador "Consulta de Administrador" donde se presentan los campos del Administrador: número de cuenta, apellido paterno, apellido materno, nombre, usuario, contraseña, tipo. El administrador selecciona la actividad de "Volver", donde se regresa al Caso de Uso Funciones del Administrador. S-2 Modifica Administrador. Si desea modificar, modifica los campos que desee cambiar y presiona el botón "Modifica", el cual se encarga de enviar los datos a ser modificados y actualizarlos en la base de datos.

Tabla 3.10. Consulta de Profesor

Caso de Uso	Consultar Información Profesor.
Actores	Administradores, Base de Datos
Propósito	Consulta de información de los usuarios (Profesores)
Resumen	Ofrece la funcionalidad de proporcionar la información detallada de Profesores.
Precondiciones	Haber sido validado al usuario Administrador y haber registros de Profesores.
Flujo Principal	Se tiene la pantalla de datos de los usuarios (Profesor). Se presiona la actividad de “Volver” y se regresa al Caso de Uso funciones del administrador.
Subflujos	S-1 Consulta de Profesor. Se presenta la pantalla al administrador “Consulta de Profesor”, donde se presentan los campos del Profesor: número de cuenta, apellido paterno, apellido materno, nombre, usuario, contraseña, tipo. El administrador selecciona la actividad de “Volver”, donde se regresa al Caso de Uso Funciones del Administrador. S-2. Modifica Profesor. Si el administrador desea cambiar los datos lo puede hacer, posteriormente presiona la actividad “Modifica”, la cual se encarga de enviar los datos y actualizarlos en la base de datos.

Tabla 3.11. Baja Administrador

Caso de Uso	Dar Baja Administrador
Actores	Administradores, Base de Datos
Propósito	Dar de Baja en el SIEL a un Administrador.
Resumen	El usuario inicia este Caso de Uso donde ofrece la funcionalidad de eliminar a un Administrador.
Precondiciones	Haber ejecutado el Caso de Uso, validar al Usuario (Administrador), y haber sido autenticado.

Flujo Principal	<p>Se tiene la pantalla de funciones de Administrador donde se puede seleccionar la acción de Eliminar Registro.</p> <p>Se puede seleccionar la actividad de “Borrar” para que los datos sean eliminados.</p> <p>Si la actividad seleccionada es “Volver”, se regresa al Caso de Uso funciones del administrador.</p>
Subflujos	<p>S-1 Dar de Baja al Administrador.</p> <p>Se presenta la pantalla al Administrador “Baja de Administrador”, en donde se selecciona el registro que se quiere eliminar.</p> <p>El administrador puede seleccionar las siguientes actividades, “Borrar” y “Volver”.</p> <p>Si el administrador selecciona “Borrar” los datos del SIEL son eliminados, y se continua con el subflujo de Datos Eliminados(S-2).</p> <p>Si el administrador selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Datos Eliminados.</p> <p>Se presenta la página “Datos Eliminados” y se regresa al Caso de Uso Funciones del Administrador.</p>

Los objetos Borde, Entidad y Control para el módulo del Administrador son:
(Ver tablas 3.12 a 3.14).

Tabla 3.12. Objetos Borde del Módulo Administrador

Casos de Uso	Actores	Objetos Borde
Validar Usuario.	Administrador, Profesor, Alumno, BD Registro de Usuario.	InterfazAdministrador, InterfazProfesor, InterfazAlumno, Interfaz Base de Datos, Pantalla Ingreso al Sistema.
Funciones Administrador.	Administrador, BD Registro de Usuario.	InterfazAdministrador, Pantalla Módulo Administrador.
Registrar	Administrador, BD	InterfazAltaAdministrador,

Administrador.	Registro de Usuario.		InterfazBDRegistroUsuario, InterfazComprobarRegistro, Pantalla Dar de Alta
Registrar Profesor.	Administrador, Registro de Usuario.	BD	InterfazAltaAdministrador, InterfazBDRegistroUsuario, InterfazComprobarRegistro, Pantalla Dar de Alta.
Consultar Información Profesor/Administrador.	Administrador, Registro de Usuario.	BD	InterfazConsultaAdministrador, InterfazBDRegistroUsuario, Pantalla Consulta.
Dar Baja Administrador/Profesor.	Administrador, Registro de Usuario.	BD	InterfazBajaAdministrador, InterfazBDRegistroUsuario, Pantalla Consulta.

Tabla 3.13. Objetos Entidad del Módulo de Administrador

Casos de Uso	Objetos Entidad
Validar Usuario.	RegistroUsuario.
Funciones Administrador.	
Registrar Administrador.	RegistroUsuario.
Registrar Profesor.	RegistroUsuario.
Consultar Información Profesor/Administrador.	Número de Cuenta, Apellido Paterno, Apellido Materno, Nombre, Usuario, Contraseña, Tipo.
Dar Baja Administrador/Profesor.	Número de Cuenta, Apellido Paterno, Apellido Materno, Nombre, Usuario, Contraseña, Tipo.

Tabla 3.14. Objetos Control del Módulo de Administrador

Casos de Uso	Objeto Control
Validar Usuario.	ManejadorValidarUsuario.
Funciones Administrador.	ManejadorFuncionesAdministrador.
Registrar Administrador.	ManejadorRegistroAdministrador.
Registrar Profesor.	ManejadorRegistroProfesor.
Consultar Información Profesor/Administrador.	ManejadorConsultaUsuario.
Dar Baja Administrador/Profesor.	ManejadorBajaUsuario.
	ManejadorPrincipal.

Para el Módulo de Profesor se mencionan las siguientes opciones:
(Ver figura 3.7).

- Altas de grupo, alumnos y reactivos de la evaluación en línea.
- Se puede realizar el proceso de baja de cada uno de los elementos mencionados en el punto anterior.
- Consulta del desempeño de los estudiantes y de los grupos.

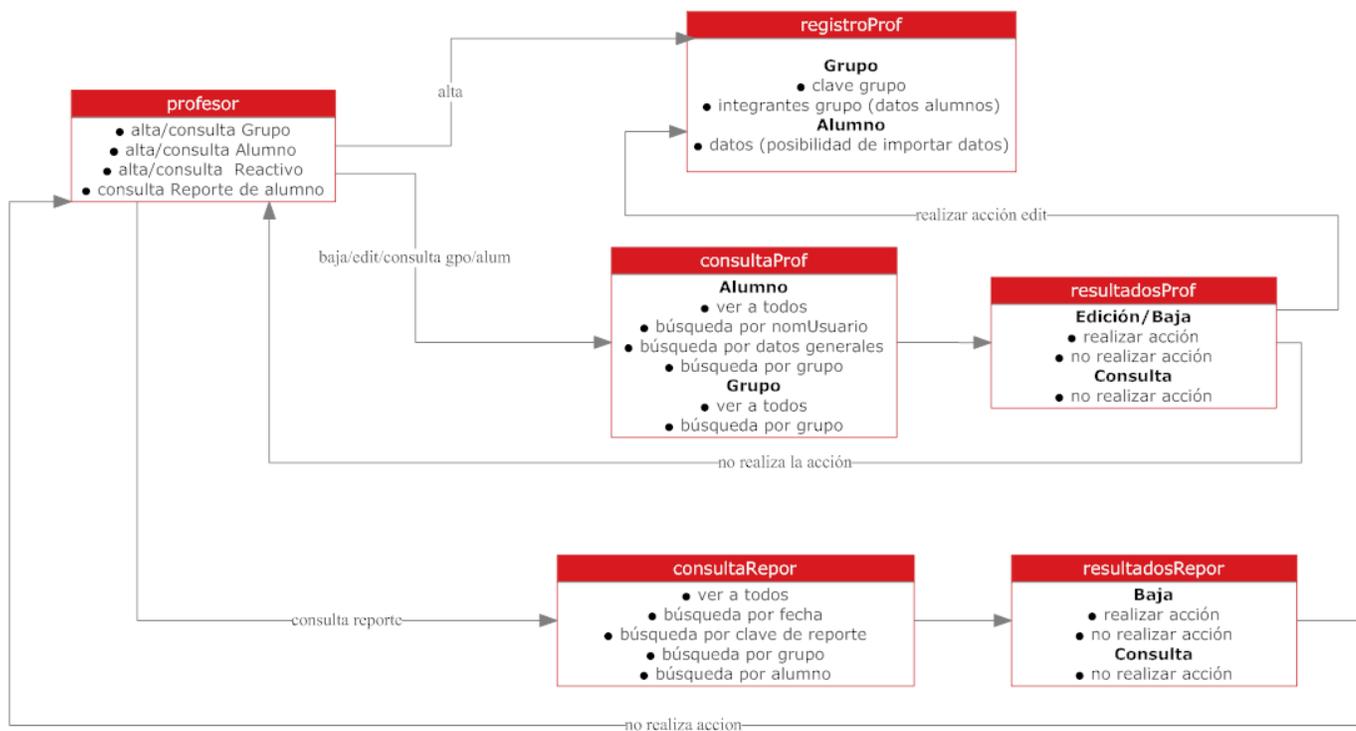


Figura 3.7. Módulo del Profesor (Altas/Bajas de Alumnos) y Reportes.

Dentro del módulo del profesor se tiene la opción para dar de alta o baja los reactivos, que también permite realizar consultas de los mismos. (Ver figura 3.8)



Figura 3.8. Módulo del Profesor detallando el Alta de reactivos así como su respectiva consulta.

Casos de Uso PROFESOR:

Tabla 3.15. Funciones de Profesor

Actor	Funciones de Profesor
Casos de Uso	Alta Grupo Alta Alumno Alta Reactivo Consulta Grupo Consulta Alumno Consulta Reactivo Consulta Reporte de alumno Editar Grupo Editar Alumno Editar Reactivo Baja Grupo Baja Alumno Baja Reactivo
Tipo	Secundario
Descripción	Es un actor secundario y representa los Casos de Uso que puede usar el Profesor.

Tabla 3.16. Alta Grupo

Caso de Uso	Dar de alta grupo de alumnos
Actores	Profesor, Base de Datos.
Propósito	Dar de alta en el sistema a un nuevo grupo de alumnos.
Resumen	El usuario inicia este Caso de Uso. Ofrece la funcionalidad de dar de alta a un grupo de alumnos.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado como profesor.
Flujo Principal	<p>Se tiene la pantalla de ingreso de datos, se ingresan los datos correspondientes, se continúa con los subflujos de este Caso de Uso.</p> <p>Se presiona la actividad seleccionada “Enviar”, para que los datos sean validados para la correcta inserción del registro.</p> <p>Si la actividad seleccionada es” Volver”, se regresa al Caso de Uso funciones del Profesor.</p>
Subflujos	<p>S-1 Dar de alta al grupo.</p> <p>Se presenta la pantalla al profesor “Dar de alta al Grupo”, se tienen los campos que deben ser llenados por datos del grupo, como son clave de grupo y horario.</p> <p>El profesor puede seleccionar las siguientes actividades, “Enviar” y “Volver”.</p> <p>Si el profesor selecciona “Enviar”, el sistema genera un registro nuevo. Se continúa con el subflujo Comprobar Registro (S-2).</p> <p>Si el profesor selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Comprobar Registro.</p> <p>Se presenta la página “Datos del Registro Insertado”</p> <p>Dicha página se encarga de validar los datos, provenientes de la página de “Dar de alta al grupo” (E1, E2).</p> <p>Se tiene la actividad “Volver”, la cual se encarga de regresar al Caso de Uso Funciones del profesor.</p>

Excepciones	<p>E1 Información incompleta.</p> <p>E2 Registro ya existe.</p> <p>Para ambas excepciones. Se presenta la página “Datos del Registro Insertados”. Se presenta el mensaje de “Error”. Se tiene la actividad “Volver”, la cual permite regresar al Caso de Uso Funciones del Profesor.</p>
--------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 3.17. Alta alumno

Caso de Uso	Dar de alta alumno(s)
Actores	Profesor, Base de Datos.
Propósito	Dar de alta en el sistema a un(os) nuevo(s) alumno(s).
Resumen	El usuario inicia este Caso de Uso. Ofrece la funcionalidad de dar de alta a un alumno o conjunto de alumnos por medio de la importación de una lista.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado como profesor, además de que por lo menos haya un grupo de alumnos dado de alta en el sistema.
Flujo Principal	<p>Se tiene la pantalla de ingreso de datos, se ingresan los datos correspondientes, se continúa con los subflujos de este Caso de Uso.</p> <p>Se presiona la actividad seleccionada enviar, para que los datos sean validados para la correcta inserción del registro.</p> <p>Si la actividad seleccionada es volver, se regresa al Caso de Uso funciones del Profesor.</p>
Subflujos	<p>S-1 Dar de alta al alumno.</p> <p>Se presenta la pantalla al profesor “Dar de alta al alumno”, se tienen los campos que deben ser llenados por datos del alumno, como lo son número de cuenta, apellido paterno, apellido materno y el grupo al que pertenece, dicho grupo es elegido de una lista de grupos previamente cargados al sistema, por lo que un alumno siempre está relacionado a un grupo.</p> <p>El profesor puede seleccionar las siguientes actividades,</p>

	<p>“Enviar” y “Volver”.</p> <p>Si el profesor selecciona “Enviar”, el sistema genera un registro nuevo. Se continúa con el subflujo Comprobar Registro (S-2).</p> <p>Si el profesor selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Comprobar Registró.</p> <p>Se presenta la página “Datos del Registro Insertado”</p> <p>Dicha página se encarga de validar los datos, provenientes de la página de “Dar de alta al alumnos” (E1, E2).</p> <p>Se tiene la actividad “Volver”, la cual se encarga de regresar al Caso de Uso Funciones del profesor.</p>
Excepciones	<p>E1 Información incompleta.</p> <p>E2 Registro ya existe.</p> <p>Para ambas excepciones. Se presenta la página “Datos del Registro Insertado”. Se presenta el mensaje de “Error”. Se tiene la actividad “Volver”, la cual permite regresar al Caso de Uso Funciones del Profesor.</p>

Tabla 3.18. Alta reactivo

Caso de Uso	Dar de alta reactivo
Actores	Profesor, Base de Datos.
Propósito	Dar de alta en el sistema a un nuevo reactivo.
Resumen	El usuario inicia este Caso de Uso. Ofrece la funcionalidad de dar de alta a un reactivo.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado como profesor.
Flujo Principal	<p>Se tiene la pantalla de ingreso de datos, se ingresan los datos correspondientes, se continúa con los subflujos de este Caso de Uso.</p> <p>Se presiona la actividad seleccionada “Enviar”, para que los datos sean validados para la correcta inserción del registro.</p> <p>Si la actividad seleccionada es “Volver”, se regresa al Caso de Uso funciones del Profesor.</p>

<p>Subflujos</p>	<p>S-1 Dar de alta al alumno.</p> <p>Se presenta la pantalla al profesor “Dar de alta el reactivo”, se tienen los campos que deben ser llenados por datos del reactivo, como lo son enunciado del reactivo, instrucciones, tipo de reactivo (dinámico, estático, opción múltiple), grado de complejidad, imagen del circuito, tiempo estimado de resolución, método necesario de resolución, matrices necesarias para la resolución y la información que se desea evaluar del alumno (voltajes, corrientes específicas, etc) .</p> <p>El profesor puede seleccionar las siguientes actividades, “Vista previa y “Volver”.</p> <p>Si el profesor selecciona “Vista previa”, el sistema genera una página con la vista previa del reactivo. Además se mostrará con el subflujo Vista previa del reactivo(S-3). El profesor puede elegir las siguientes actividades, “Enviar” y “Volver”.</p> <p>Si el profesor selecciona “enviar”, el sistema genera un registro nuevo. Se continúa con el subflujo “Comprobar Registro” (S-2).</p> <p>Si el profesor selecciona “Volver” se regresa al Caso de Uso Funciones del Administrador.</p> <p>S-2 Comprobar Registro.</p> <p>Se presenta la página “Datos del Registro Insertado”</p> <p>Dicha página se encarga de validar los datos, provenientes de la página de “Dar de alta el reactivo” (E1, E2).</p> <p>Se tiene la actividad “Volver”, la cual se encarga de regresar al Caso de Uso Funciones del profesor.</p> <p>S-3 Vista previa del reactivo.</p> <p>Se presenta la página “Vista Previa”. En esta página el profesor puede observar el reactivo tal y como será mostrado al alumno en el momento de la resolución del examen.</p>
<p>Excepciones</p>	<p>E1 Información incompleta.</p>

	<p>E2 Registro ya existe.</p> <p>Para ambas excepciones. Se presenta la página “Datos del Registro Insertado”. Se presenta el mensaje de “Error”. Se tiene la actividad “Volver”, la cual permite regresar al Caso de Uso Funciones del Profesor.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 3.19. Consulta de Grupo

Caso de Uso	Consultar Información Profesor
Actores	Profesor, Base de Datos
Propósito	Consulta de información de los grupos dados de alta previamente por el profesor
Resumen	Ofrece la funcionalidad de proporcionar la información detallada de los grupos.
Precondiciones	Haber sido validado al usuario Profesor y haber registros de grupos.
Flujo Principal	<p>Se tiene la pantalla de datos de los grupos (Profesor).</p> <p>Se presenta la opción de “Editar” o “Dar de baja el grupo”. Se llama entonces a los Casos de Usos, “Editar Grupo” y “Dar de baja grupo”, respectivamente.</p> <p>Se presiona la actividad de “Volver” y se regresa al Caso de Uso funciones del Profesor.</p>
Subflujos	<p>S-1 Consulta de grupo.</p> <p>Se presenta la pantalla al administrador “Consulta de grupo” donde se presentan los campos del grupo: clave, profesor, número de alumnos.</p> <p>El profesor selecciona la actividad de “volver”, donde se regresa al Caso de Uso Funciones del Profesor.</p>

A continuación se muestran los objetos Borde, Entidad y Control para el módulo del Alumno. (Ver tablas 3.20 al 3.22).

Tabla 3.20. Objetos Borde del Módulo Profesor.

Casos de Uso	Actores	Objetos Borde
Funciones Profesor.	Profesor, BD Registro de Usuario.	InterfazProfesor, InterfazRegistroUsuario, Pantalla Módulo de Profesor.
Registrar Grupo.	Profesor, BD Grupo, BD Registro de Usuario.	InterfazProfesor, InterfazBDGrupo, InterfazRegistroUsuario, Pantalla Registrar Grupo.
Registrar Alumno.	Profesor, BD Registro de Usuario.	InterfazProfesor, InterfazBDRegistroUsuario, Pantalla Registrar Alumno.
Registrar Reactivo.	Profesor, BD Reactivos Examen.	InterfazProfesor, InterfazBDReactivosExamen, Pantalla Registrar Reactivo.
Consultar Información Alumno/Grupo.	Profesor, BD Registro de Usuario, BD Grupo.	InterfazProfesor, InterfazBDRegistroUsuario, InterfazBDGrupo, Pantalla Consultar Información Alumno/Grupo
Consultar Información Reactivo.	Profesor, BD Registro Reactivos de Examen.	InterfazProfesor, InterfazBDReactivosExamen, Pantalla Consultar Información Reactivo.
Consultar Información Resultados (Reporte).	Profesor, BD Resultados de Examen.	InterfazProfesor, InterfazBDRResultadoExamen, Pantalla Consultar Información Resultados (Reporte).
Dar de Baja Alumno/Grupo.	Profesor, BD Registro de Usuario.	InterfazProfesor, InterfazBDRegistroUsuario, InterfazBDGrupo, Pantalla Dar

		deBaja Alumno/Grupo.
Dar de Baja Reactivos.	Profesor, BD Registro Reactivos de Examen.	InterfazProfesor, InterfazBDReactivosExamen, Pantalla Dar de Baja Reactivos.

Tabla 3.21. Objetos Entidad del Módulo Profesor.

Casos de Uso	Objeto Entidad
Funciones Profesor.	Registro Usuario.
Registrar Grupo.	Número de Grupo, Clave de Profesor.
Registrar Alumno.	Número de Cuenta, Apellido Paterno, Apellido Materno, Nombre, Usuario, Contraseña, Tipo.
Registrar Reactivo.	Diagrama de pregunta, texto de pregunta, nivel examen.
Consultar Información Alumno/Grupo.	Número de Cuenta, Apellido Paterno, Apellido Materno, Nombre, Usuario, Contraseña, Tipo, Número de Grupo, Clave de Profesor.
Consultar Información Reactivo.	Diagrama de pregunta, texto de pregunta, resultados.
Consultar Información Resultados (Reporte).	Diagrama de pregunta, texto de pregunta, Tipo de examen, resultados, valor de parámetros.
Dar de Baja Alumno/Grupo.	Número de Cuenta, Apellido Paterno, Apellido Materno, Nombre, Usuario, Contraseña, Tipo, Número de Grupo, Clave de Profesor.
Dar de Baja Reactivos.	Diagrama de pregunta, texto de pregunta, nivel examen.

Tabla 3.22. Objetos Control del Módulo Profesor.

Casos de Uso	Objeto Control
Funciones Profesor.	ManejadorFuncionesProfesor.
Registrar Grupo.	ManejadorRegistrarGrupo.
Registrar Alumno.	ManejadorRegistrarAlumno.
Registrar Reactivo.	ManejadorRegistrarReactivo.
Consultar Información	ManejadorConsultarInformaciónAlumno/Grupo.

Alumno/Grupo.	
Consultar Información Reactivo.	ManejadorConsultarInformaciónReactivo.
Consultar Información Resultados (Reporte).	ManejadorConsultarInformaciónResultado.
Dar de Baja Alumno/Grupo.	ManejadorDarBajaAlumno/Grupo.
Dar de Baja Reactivos.	ManejadorDarBajaReactivos.
	ManejadorPrincipal.

En el módulo “alumno”, el usuario ingresa al sistema y selecciona entre el nivel de examen Básico o Avanzado para evaluar su avance, teniendo para cada uno cierto número de intentos y tiempo limitado. El alumno puede consultar información sobre número de exámenes realizados en cada nivel, sus respectivas calificaciones y cambiar su contraseña. (Ver figura 3.9).

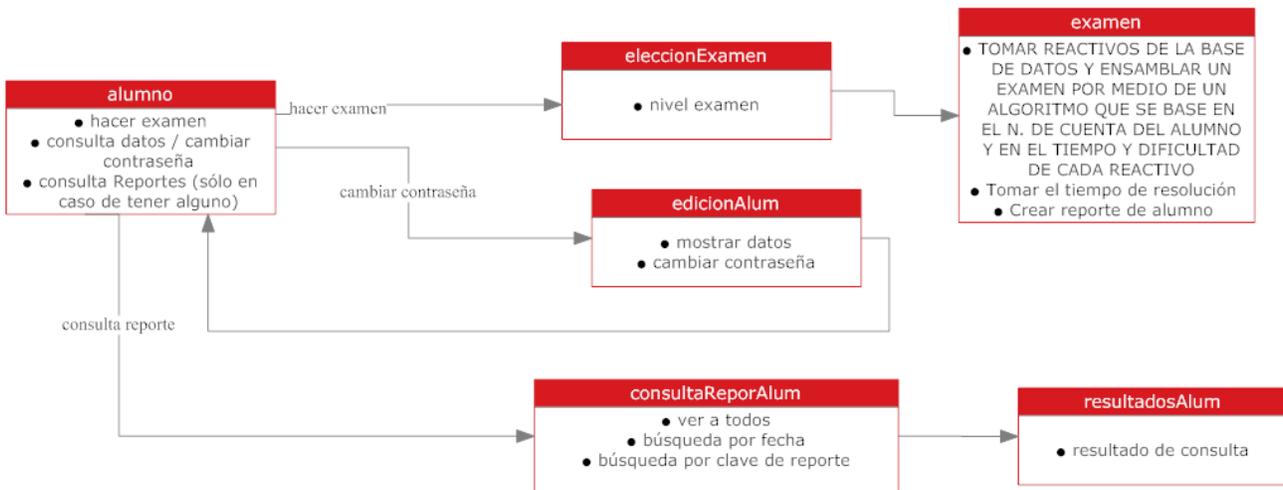


Figura 3.9. Módulo del Alumno.

Casos de Uso ALUMNO

Tabla 3.23. Funciones de Alumno

Caso de Uso	Funciones de Alumno
Actores	Alumno, Base de Datos.
Propósito	Conectar los diferentes tipos de procesos para el Alumno.
Resumen	Aquí el Alumno podrá escoger entre realizar un examen (Básico/Avanzado). Ver reportes que muestran los resultados de los exámenes. Llevar a cabo gestiones sencillas de administración como cambiar contraseña.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado.
Flujo Principal	Se tiene la pantalla de Módulo de Alumno, donde continúan los subflujos de este Caso de Uso. Se presiona la actividad seleccionada la cual puede ser: Realizar examen (Básico/Avanzado), consultar reporte, administración de cuenta.
Subflujos	S-1 Realizar examen Básico/Avanzado. S-2 Consultar Reporte. S-3 Administración Cuenta.

Tabla 3.24. Realizar Examen

Caso de Uso	Realizar examen.
Actores	Alumno, Base de Datos.
Propósito	El alumno podrá llevar a cabo una evaluación (Básica/Avanzada).
Resumen	Aquí el Alumno podrá escoger entre realizar un examen (Básico/Avanzado). Tomando en cuenta que únicamente tendrá dos intentos para hacer cada tipo de examen.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado.

Flujo Principal	<p>Se tiene la pantalla de Realizar Examen, se presentan dos actividades principales.</p> <p>Realizar Examen Básico: en donde el sistema escoge las preguntas que cumplan con este requisito, para que el alumno pueda contestarlas.</p> <p>Realizar Examen Avanzado: en donde el sistema escoge las preguntas que cumplan con este requisito, para que el alumno pueda contestarlas.</p>
Subflujos	<p>S-1 Realizar examen Básico.</p> <p>S-2 Realizar examen Avanzado.</p>

Tabla 3.25. Consultar Reporte

Caso de Uso	Consultar Reporte.
Actores	Alumno, Base de Datos.
Propósito	El alumno puede consultar el resultado de las evaluaciones realizadas (Básica/Avanzada).
Resumen	El alumno puede consultar información de sus evaluaciones presentadas.
Precondiciones	Haber ejecutado el Caso de Uso, Validar Usuario, y haber sido autenticado. Haber realizado por lo menos un examen para poder consultar información.
Flujo Principal	<p>Se tiene la pantalla Consultar Reporte, en donde aparecen todos los exámenes presentados.</p> <p>El alumno selecciona uno para ver el reporte del examen seleccionado.</p>
Subflujos	S-1 Seleccionar un examen para ver reporte.

Los objetos Borde, Entidad y Control para el módulo del Alumno se muestran a continuación. (Ver tablas 3.26 a 3.28).

Tabla 3.26. Objetos Borde del Módulo Alumno.

Casos de Uso	Actores	Objetos Borde
Funciones Alumno.	Alumno, BD Registro de Usuario.	InterfazAlumno, InterfazRegistroUsuario.
Realizar Examen Básico/Avanzado.	Alumno, BD Reactivos de Examen, BD Registro de Usuario.	InterfazAlumno, InterfazReactivosExamen, InterfazRegistroUsuario, InterfazRealizarExamen, Pantalla Realizar Examen.
Consultar Reporte.	Alumno, BD Respuesta de Examen.	InterfazAlumno, InterfazBDRespuestaExamen, Pantalla Consultar Reporte.
Administrar Cuenta.	Alumno, BD Registro de Usuario.	InterfazAlumno, InterfazBDRegistroUsuario, Pantalla Administrar Cuenta.

Tabla 3.27. Objetos Entidad del Módulo de Alumno.

Casos de Uso	Objetos Entidad
Funciones Alumno.	RegistroUsuario.
Realizar Examen Básico/Avanzado.	Diagrama de pregunta, texto de pregunta, resultados.
Consultar Reporte.	Diagrama de pregunta, texto de pregunta, resultados, parámetros utilizados, calificación.
Administrar Cuenta.	RegistroUsuario.

Tabla 3.28. Objetos Control del Módulo de Alumno.

Casos de Uso	Objeto Control
Funciones de Alumno.	ManejadorFuncionesAlumno.
Realizar Examen Básico/Avanzado.	ManejadorRealizarExamen.
Consultar Reporte.	ManejadorConsultarReporte.
Administrar Cuenta.	ManejadorAdministrarCuenta.
	ManejadorPrincipal.

El módulo donde se realiza el examen, tiene la función de generar la evaluación que el alumno contestará, una vez generada y realizada, se guardan los resultados en la Base de Datos para su posterior consulta.

Para la selección de los reactivos en los exámenes se usa un algoritmo que los va eligiendo de acuerdo al puntaje, hasta que los puntajes de los reactivos alcancen la calificación total del examen.

3.2.3 MODELO DE DISEÑO DEL SIEL

En el modelo de diseño se describen los diagramas de secuencia y las Tarjetas de Clase (Objeto). (Ver figuras 3.10 a 3.16).

Para el Módulo Administrador son:

- InterfazInicio.
- InterfazFuncionesAdministrador.
- InterfazConsultaAdministrador.
- InterfazModificarAdministrador.
- InterfazBajaAdministrador.
- InterfazAltaAdministrador.

Tarjeta para el Objeto InterfazInicio con responsabilidades, colaboraciones, contratos y especificaciones de algoritmos identificados de los Casos de Uso, Autenticar Usuario, Funciones Administrador, Funciones Profesor, Funciones Alumno (Ver tabla 3.29).

Tabla 3.29. Interfaz Inicio.

Objeto	InterfazInicio
Descripción.	Interacción del Usuario al ingresar al sistema.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Ingresar, Pantalla Módulo Administrador, Pantalla Módulo Profesor, Pantalla Módulo Alumno.
2. Enviar Evento.	
DatosUsuario(Datos) . Función encargada de enviar los datos de usuario y Contraseña enviados por el usuario.	
3.Manejar Evento.	
AutenticarUsuario (Datos) devuelve registro. Función encargada de verificar los datos de usuario y Contraseña y autenticar al usuario para enviarlo al Módulo correspondiente.	ManejadorValidarUsuario, InterfazBDRegistroUsuario, ManejadorFuncionesAdministrador, ManejadorFuncionesProfesor, ManejadorFuncionesAlumno.

Tarjeta para el Objeto InterfazFuncionesAdministrador con responsabilidades, colaboraciones y contratos de los Casos de Uso, Funciones de Administrador, Registrar Administrador, Consultar Información, Dar Baja Administrador. (Ver tabla 3.30).

Tabla 3.30. Interfaz Funciones Administrador

Objeto	InterfazFuncionesAdministrador
Descripción.	Menú con todas las funciones que puede realizar el Administrador.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Funciones de Administrador, Pantalla Alta Administrador, Pantalla Baja Administrador, Pantalla Consulta Administrador/Profesor.

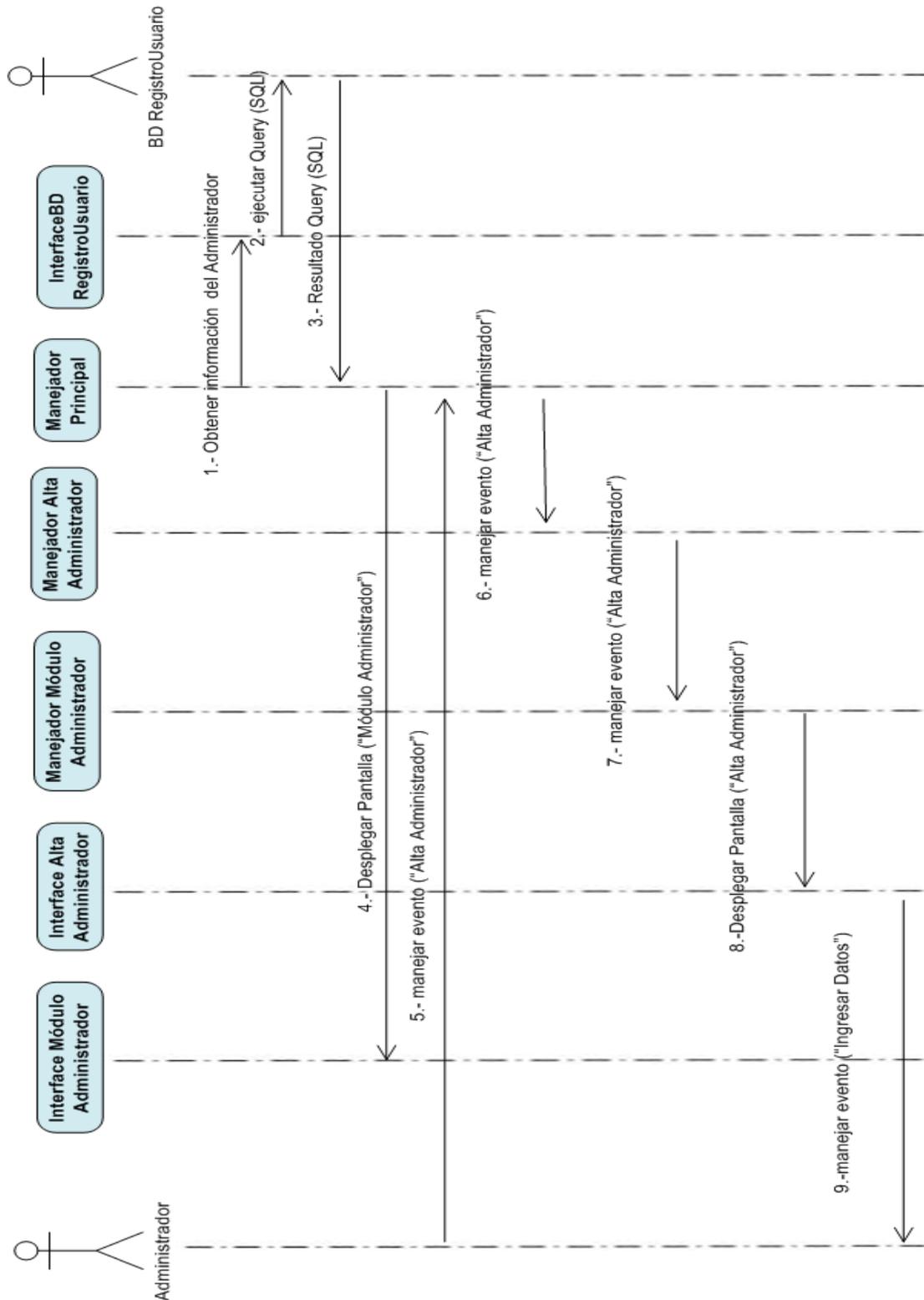


Figura 3.10. Diagrama de Secuencia de Diseño con Caso de Uso Funciones de Administrador y Caso de Uso Alta de Administrador como Subflujo.

Tarjeta para el Objeto InterfazConsultaAdministrador con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Consulta Administrador y Resultado Consulta. (Ver tabla 3.31).

Tabla 3.31. Consulta Administrador

Objeto	InterfazConsultaAdministrador.
Descripción.	El Administrador, introduce los criterios de búsqueda para efectuar la Selección de Registros.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Consulta Administrador, Pantalla Resultado Consulta.
2. Enviar Evento.	
CriteriosBúsqueda(Datos) Función que envía los datos que contienen los criterios de búsqueda.	
3. Manejar Evento.	
RealizarQuery(Datos) devuelve registro. Función que procesa los datos enviados para realizar una búsqueda en la Base de Datos y obtener los registros que concuerdan con los criterios.	InterfazBDRregistroUsuario, ManejadorConsultaAdministrador.
DesplegarDatos(Query). Función que despliega los datos del registro con formato para ser desplegado en html.	

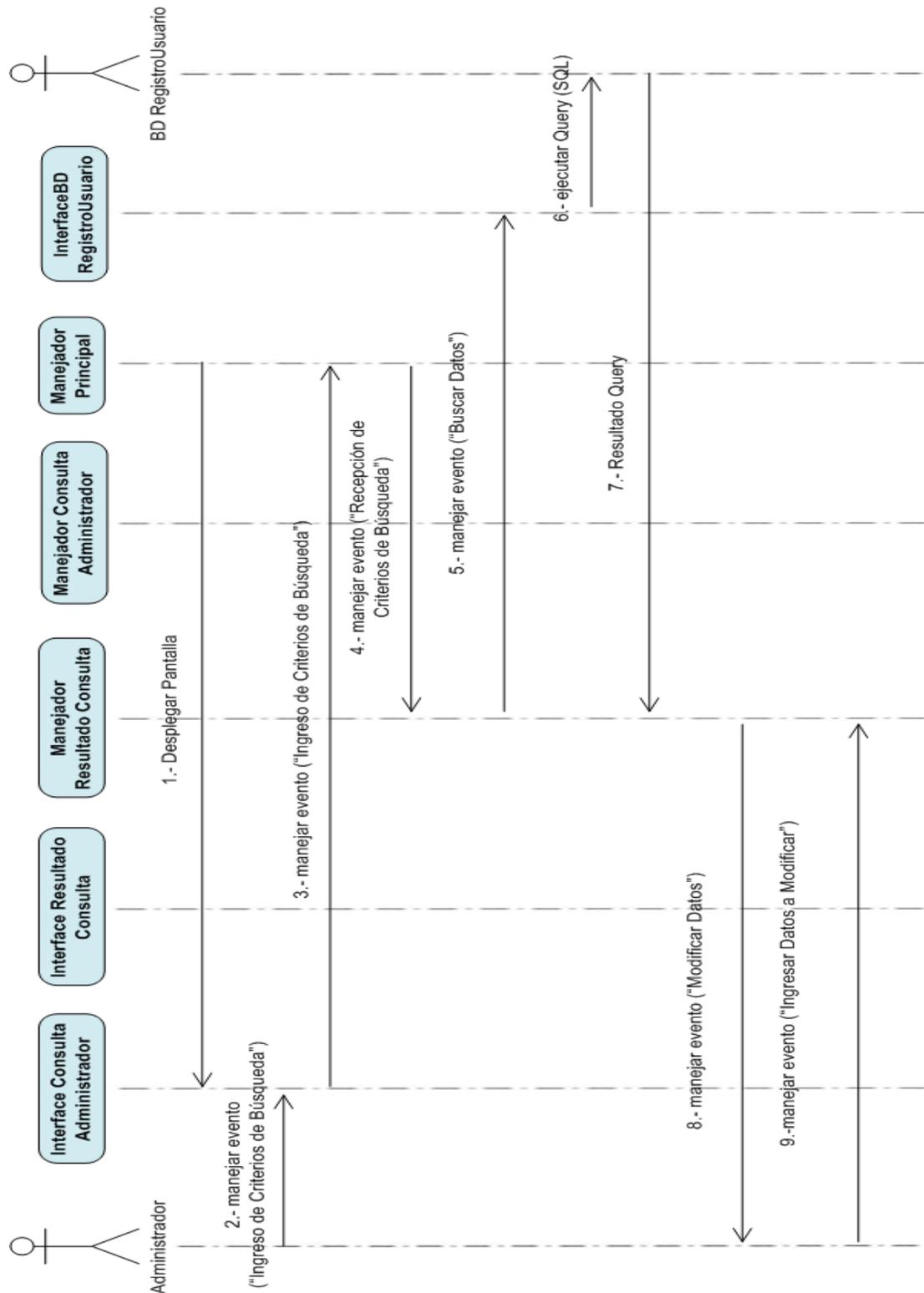


Figura 3.11. Diagrama de Secuencia de Diseño de Caso de Uso Consulta de Administrador con Subflujo 1 Resultado de Consulta y Subflujo 2 Modificar Datos.

Tarjeta para el Objeto InterfazModificarAdministrador con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Modificar Administrador. (Ver tabla 3.32).

Tabla 3.32. Modificar Administrador

Objeto	InterfazModificarAdministrador
Descripción.	El administrador puede cambiar los datos del registro original, en caso de algún error.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Modificar Administrador, Pantalla Confirmar Modificación.
2. Enviar evento.	
DatosModificados(Datos) Función que envía los datos para que sean modificados en la Base de Datos.	
3. Manejar Evento.	
DatosUpdate(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRegistroUsuario, ManejadorModificarAdministrador.

Tarjeta para el Objeto InterfazBajaAdministrador con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Baja Administrador. (Ver tabla 3.33).

Tabla 3.33. Baja Administrador

Objeto	InterfazBajaAdministrador
Descripción.	El Administrador da de Baja a un usuario Administrador del SIEL, en caso de ya no ser necesario.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página Web.	Pantalla Baja Administrador, Pantalla Confirmar Eliminación.
2. Enviar evento.	
DatosEliminadosAdm(Datos). Función que envía los datos que serán Eliminados de la Base de Datos.	
3. Manejar Evento.	
DatosDelete(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRegistroUsuario, ManejadorBajaAdministrador.

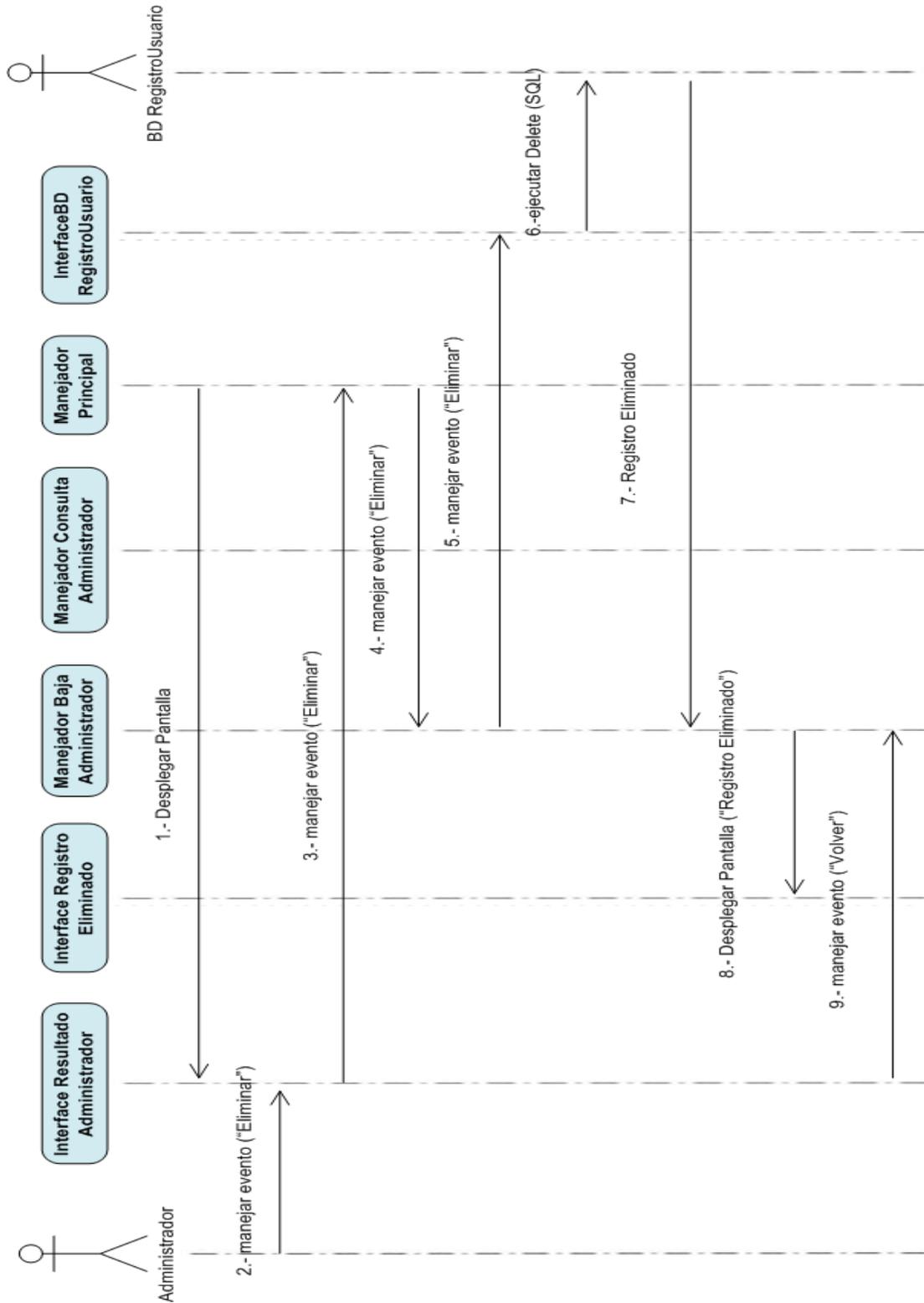


Figura 3.12. Diagrama de Secuencia de Diseño de Caso de Uso Consultar Información Administrador con Subflujo a Caso de Uso Dar Baja Administrador.

Tarjeta para el Objeto InterfazAltaAdministrador con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Alta Administrador. (Ver tabla 3.34).

Tabla 3.34. Alta Administrador

Objeto	InterfazAltaAdministrador
Descripción.	El Administrador da de Alta a un usuario Administrador en el SIEL.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página Web.	Pantalla Alta Administrador, Pantalla Confirmación de Alta.
2. Enviar evento.	
DatosAltaAdm(Datos). Función que envía los datos para que serán dados de Alta en la Base de Datos.	
3. Manejar Evento.	
DatosInsert(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRregistroUsuario, ManejadorAltaAdministrador.

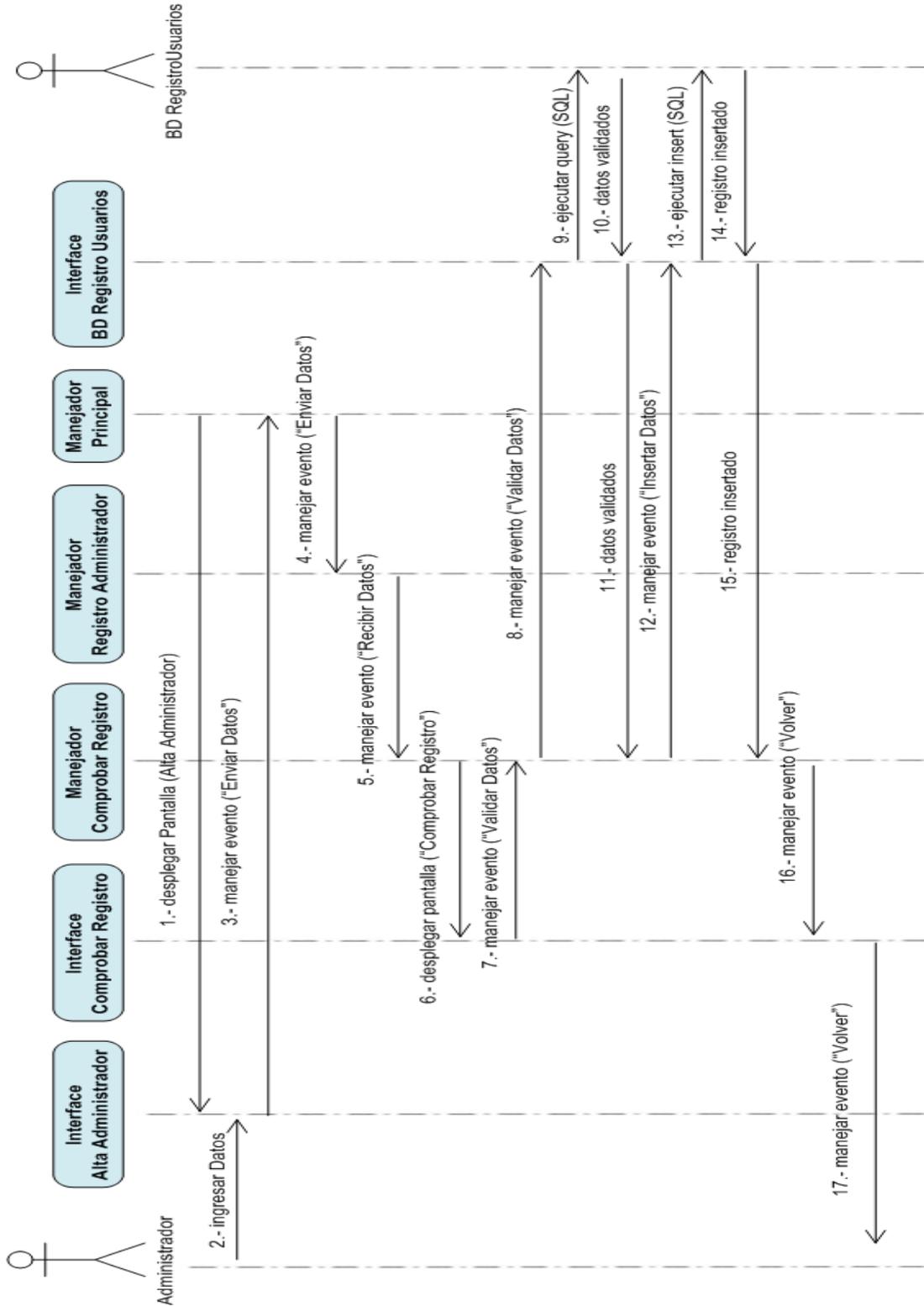


Figura 3.13. Diagrama de Secuencia de Diseño con Caso de Uso Dar de Alta Administrador con Subflujo Enviar Datos.

Las Tarjetas de Objetos para la Interfaz del Módulo Profesor son:

- InterfazEditarReactivo.
- InterfazConsultaReporte.
- InterfazAltaGrupo.

Tarjeta para el Objeto InterfazEditarReactivo con responsabilidades, colaboraciones, contratos y algoritmos identificados del Caso de Uso Editar Reactivo con Subflujo despliegue de datos. (Ver tabla 3.35).

Tabla 3.35. Editar Reactivo

Objeto	InterfazEditarReactivo
Descripción.	El profesor puede ingresar y editar reactivos que servirán para ensamblar el examen que realizará el alumno.
Contratos.	
1.Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Ingresar/Modificar Reactivo, Pantalla Confirmar Modificación, Pantalla Vista Previa
2. Enviar Evento.	
ReactivosModificados(Datos). ReactivosIngresados(Datos). Función que envía los datos para que sean modificados en la Base de Datos.	InterfazAltaReactivos
3. Manejar Evento.	
ReactivosUpdate(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDReactivos, ManejadorAltaDatos.
desplegarDatos(). Función que despliega los datos insertados.	InterfazBDReactivos, ManejadorEditarReactivos.

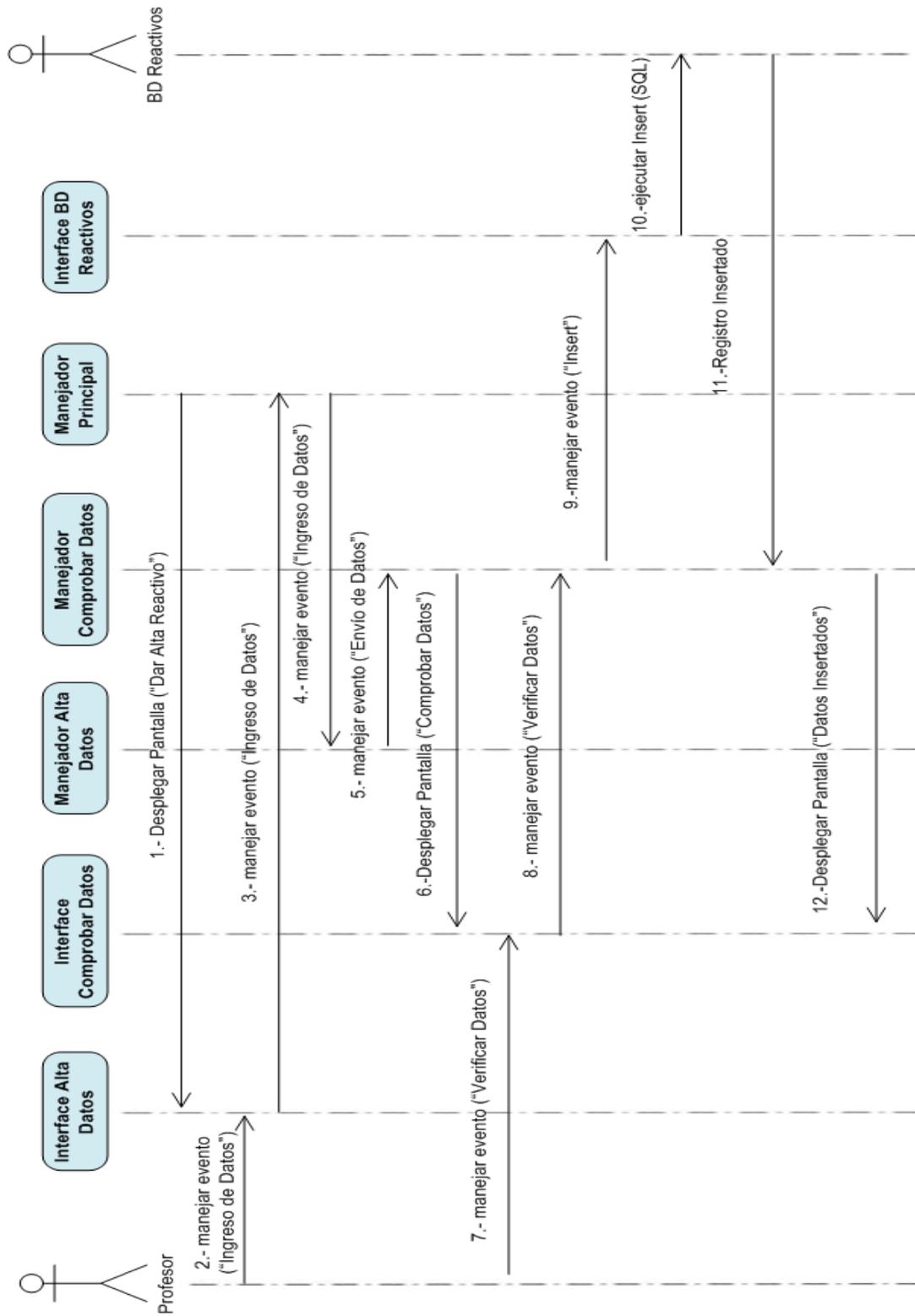


Figura 3.14. Diagrama de Secuencia de Diseño de Caso de Uso Editar Reactivo.

Tarjeta para el Objeto InterfazConsultaReporte con responsabilidades, colaboraciones, contratos y algoritmos identificados para el Caso de Uso Consulta Reporte Alumno con Subflujo despliegue de datos. (Ver tabla 3.36).

Tabla 3.36. Consulta Reporte

Objeto	InterfazConsultaReporte
Descripción.	El profesor puede revisar los reportes del desempeño de los alumnos que son generados cuando el alumno realiza el examen.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla de reporte por grupo. Pantalla de reporte por alumno.
2. Enviar Evento.	
DatosConsultaRep(Datos) Función que envía los datos para que se realice una búsqueda con el criterio elegido.	InterfazConsultaInformación
3. Manejar Evento.	
ResultadoReporte (Datos) devuelve registros. Función encargada de la búsqueda del reporte.	InterfazBDRResultadoReactivos, ManejadorDespliegueInformación, InterfazDespliegueReporte

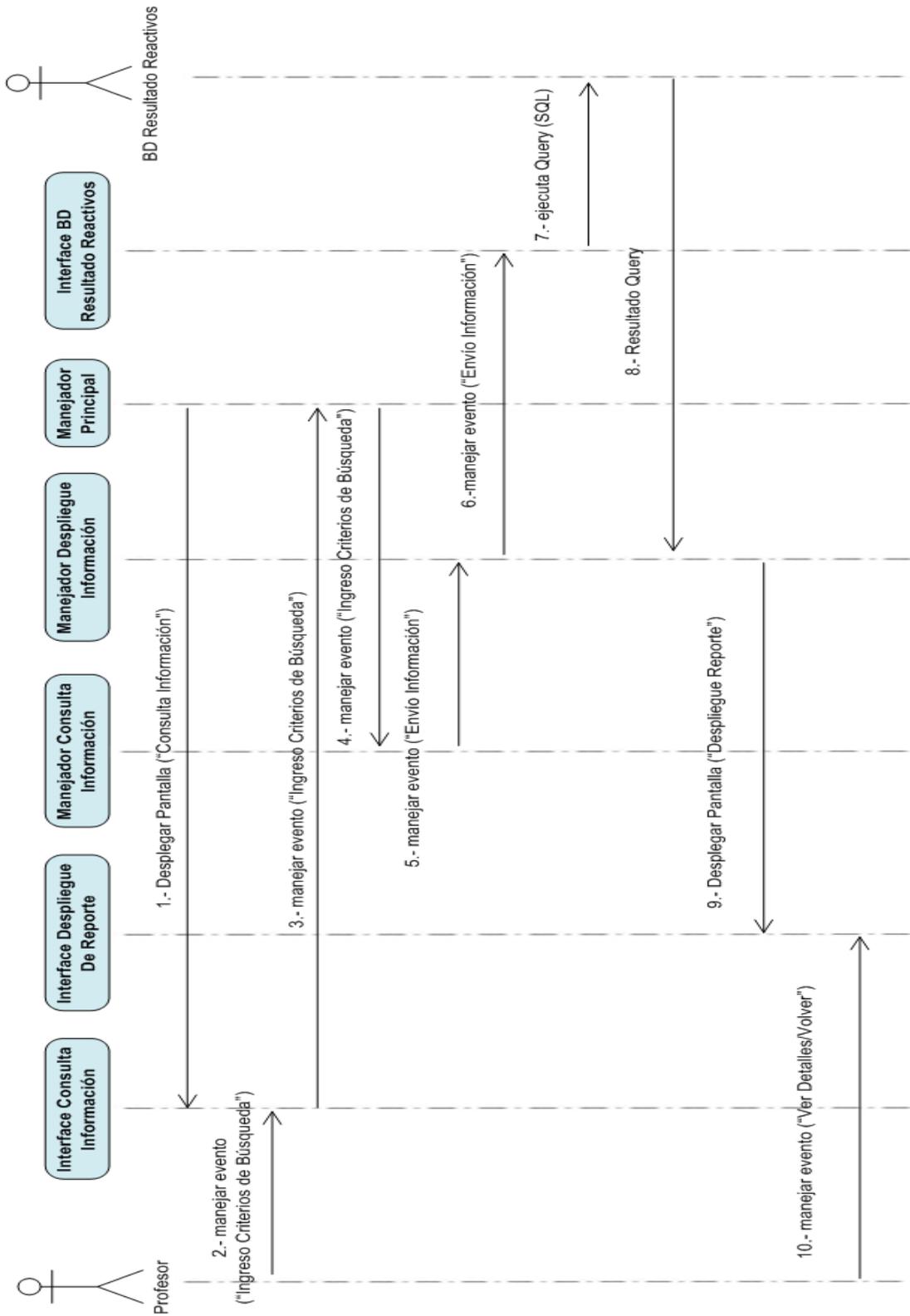


Figura 3.15. Diagrama de SecuenciFigura 3.15.Diagrama de Secuencia de Diseño con Caso de Uso Consulta Reporte Alumno con Subflujo Despliegue de Información.

Tarjeta para el Objeto InterfazAltaGrupo con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso Alta Grupo. (Ver tabla 3.37).

Tabla 3.37. Alta Grupo

Objeto	InterfazAltaGrupo
Descripción.	El profesor puede ingresar grupos
Contratos.	
1.Desplegar Pantalla.	
Nombre del archivo correspondiente a la página web.	Pantalla Agregar grupo.
2. Enviar evento.	
GrupoIngresado(Datos) Función que envía los datos para que sean modificados en la Base de Datos.	InterfazIngresoGrupo
3. Manejar Evento.	
GrupoUpdate(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRegistroGrupo, ManejadorIngresarGrupo.

Tarjeta para el Objeto InterfazBajaProfesor con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Baja Profesor. (Ver tabla 3.38).

Tabla 3.38. Baja Profesor

Objeto	InterfazBajaProfesor
Descripción.	El Administrador da de Baja a un usuario Profesor del SIEL, en caso de ya no ser necesario.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página Web.	Pantalla Baja Profesor, Pantalla Confirmar Eliminación.
2. Enviar evento.	
DatosEliminadosPro(Datos) Función que envía los datos que serán Eliminados de la Base de Datos.	
3. Manejar Evento.	
DatosDelete(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRegistroUsuario, ManejadorBajaProfesor.

Tarjeta para el Objeto InterfazAltaProfesor con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso, Alta Profesor. (Ver tabla 3.39).

Tabla 3.39 Alta Profesor

Objeto	InterfazAltaProfesor
Descripción.	El Administrador da de Alta a un usuario Profesor en el SIEL.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página Web.	Pantalla Alta Profesor, Pantalla Confirmación de Alta.
2. Enviar evento.	
DatosAltaPro(Datos). Función que envía los datos para que sean dados de Alta en la Base de Datos.	
3. Manejar Evento.	
DatosInsert(Datos). Función que realiza una actualización (update) a la Base de Datos.	InterfazBDRegistroUsuario, ManejadorAltaProfesor.

La Tarjeta de Objetos para la Interfaz del Módulo Alumno es:

- InterfazRealizarExamen.

Tabla para el objeto InterfazRealizarExamen con responsabilidades, colaboraciones, contratos y algoritmos identificados de los Casos de Uso Realizar Examen y Realizar Examen Básico. (Ver tabla 3.40).

Tabla 3.40. Interfaz Realizar Examen

Objeto	InterfazRealizarExamen.
Descripción.	El alumno elige el nivel de examen.
Contratos.	
1. Desplegar Pantalla.	
Nombre del archivo correspondiente a la página Web.	Pantalla Realizar Examen, Pantalla Examen Básico, Pantalla Examen Avanzado.
2. Enviar evento.	
ElecciónExamen(Datos) Función que direcciona al nivel de examen Básico o Avanzado.	
3. Manejar Evento.	
ElecciónReactivos(). Función que regresa los reactivos del examen.	InterfazBDReactivos, InterfazRealizarExamen, manejadorRealizarExamen.

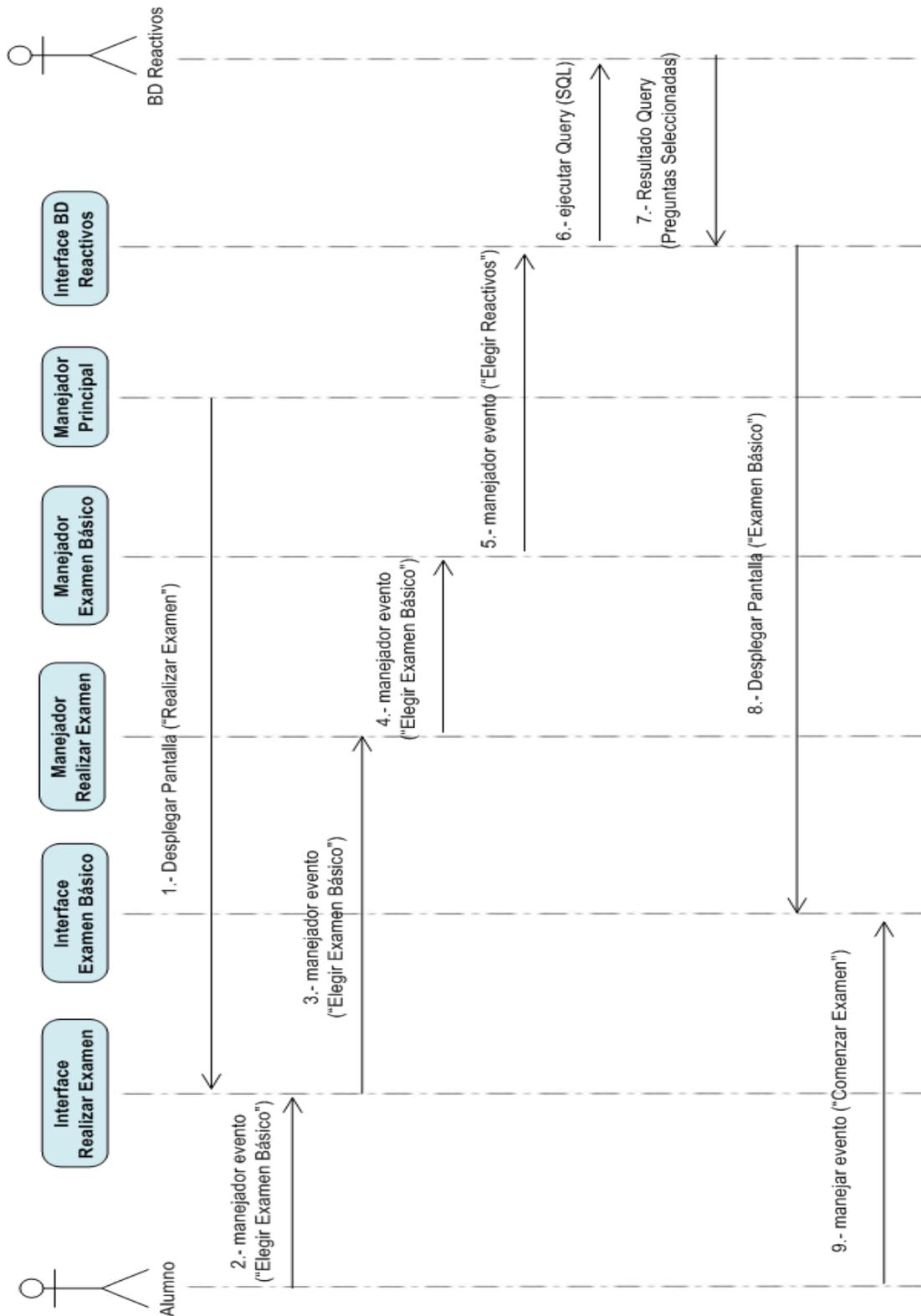
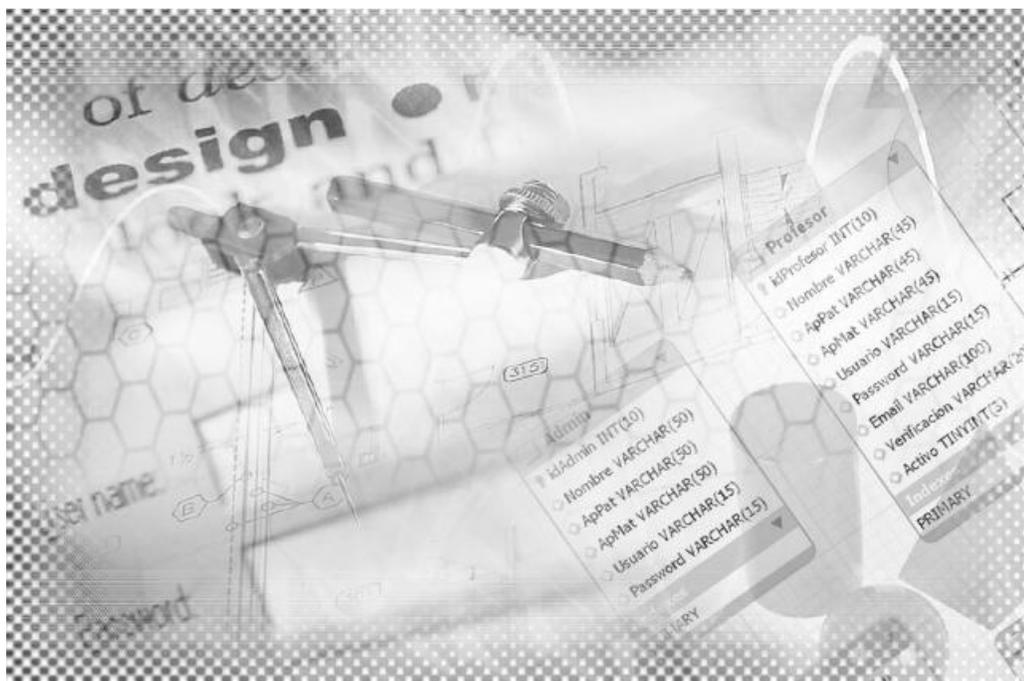


Figura 3.16. Diagrama de Secuencia de Diseño con Caso de Uso Realizar Examen con Subflujo Realizar Examen Básico.

4. DISEÑO E IMPLEMENTACIÓN



El diseño de un sistema es muy importante, un buen esbozo facilita y simplifica la implementación del proyecto de forma considerable, además de que se evitan errores como recurrencia u omisión de datos y objetos.

4.1 MODELO ENTIDAD-RELACIÓN

El modelo Entidad - Relación es una herramienta de modelado que nos permite representar de manera gráfica (Diagramas Entidad - Relación) la estructura lógica de una base de datos, siendo sus elementos las entidades, propiedades o atributos y las relaciones. Mediante éstos, el esquema conceptual abstracto puede ser mostrado gráficamente y mantener una independencia conceptual con respecto a la implementación propiamente dicha. En realidad, podemos hacer que los diagramas sean un reflejo fiel de las relaciones, interrelaciones y atributos del

modelo relacional de datos; así podemos englobar diversas relaciones en una sola entidad o conjunto de propiedades.

Los diagramas Entidad - Relación son parecidos a los diagramas de flujo (organigramas) clásicos en que utilizan rectángulos, rombos y óvalos, pero los significados de estos elementos son distintos.

En el modelo Entidad - Relación se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades.

Entidad: Objeto del mundo real del cual queremos almacenar información. La representación de una entidad es un rectángulo, para el caso de nuestra base de datos representaremos la entidad ALUMNO, véase figura 4.1:

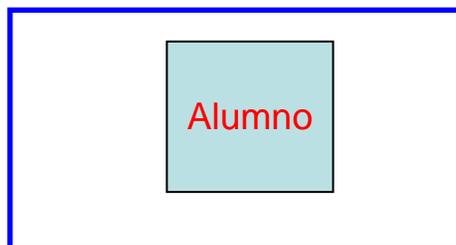


Figura 4.1. Entidad ALUMNO para un modelo Entidad Relación en una Base de Datos.

Las entidades están compuestas de Atributos que son los datos que definen el objeto. De entre los atributos habrá uno o un conjunto de ellos que no se repite; a este atributo o conjunto de atributos se le llama *clave* de la entidad. En toda entidad siempre hay al menos una clave que en el peor de los casos estará formada por todos los atributos de la tabla. Ya que puede haber varias claves y necesitamos elegir una, se hace atendiendo a las siguientes normas:

- Que sea única.
- Que se tenga pleno conocimiento de ella.
- Que sea mínima, ya que será muy utilizada por el gestor de base de datos.

Para nuestra base de datos se tienen los atributos de la entidad empleado.

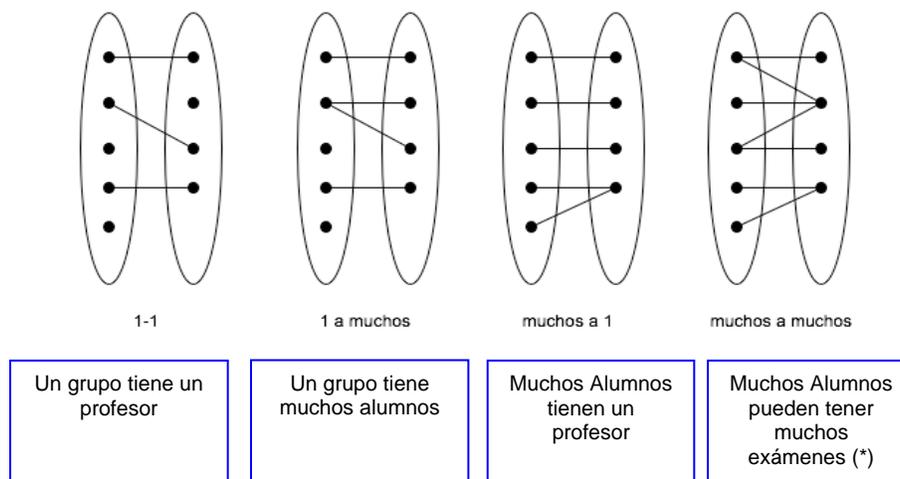
(ver figura 4.2)

Entidad : Alumno
idalumno
idgrupo
num_cuenta
nombre
usuario
password
email
intento_b
intento_a

Figura 4.2. ATRIBUTOS para la entidad ALUMNO para un modelo Entidad Relación en una Base de Datos.

Relación: Asociación entre entidades, sin existencia propia en el mundo real que estamos modelando, pero necesaria para reflejar las interacciones existentes entre entidades. Las relaciones pueden ser de cuatro tipos:

- **Relaciones “1 a 1”:** entidades que intervienen en la relación se asocian una a una.
- **Relaciones “1 a muchos”:** Una ocurrencia de una entidad está asociada con muchas de otra.
- **Relaciones “muchos a 1”:** muchas ocurrencias de una entidad están asociadas a una de otra entidad.
- **Relaciones “muchos a muchos”:** cada ocurrencia, en cualquiera de las dos entidades de la relación, puede estar asociada con muchas de la otra y viceversa. (Ver figura 4.3).



(*) Para nuestro caso se limita a 2 exámenes

Figura 4.3. Tipos de Relaciones en el modelo Entidad – Relación.

Conforme a lo anterior en la figura 4.4 veremos un ejemplo de un Diagrama del modelo Entidad-Relación.

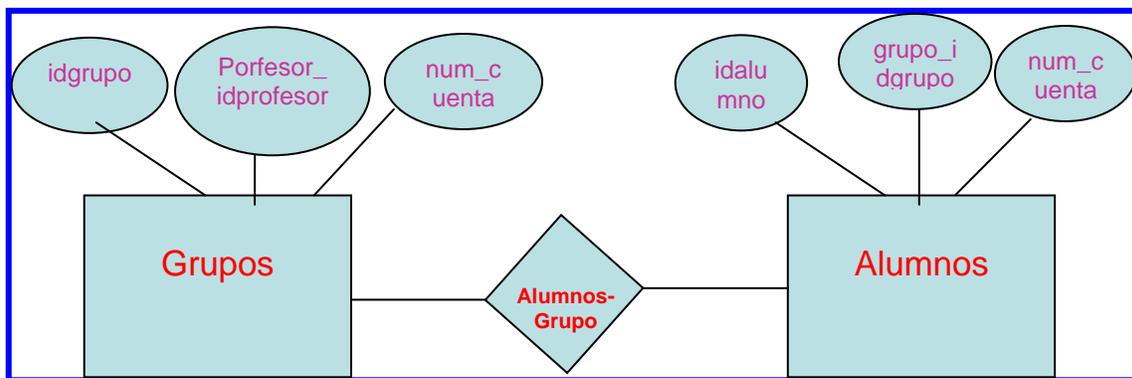


Figura 4.4. Ejemplo de Diagrama Entidad-Relación para el SIEL.

4.2 BASE DE DATOS DEL SIEL

La Base de Datos del SIEL está estructurada mediante el Modelo Entidad Relación con el cual determinamos las Entidades, Asociaciones entre Entidades, Tipos de Relaciones y Atributos de Entidades.

1) Entidades:

- admin
- profesor
- alumnos
- grupos
- reactivos
- reactivo fijo
- respuestas
- respuestas fijas
- respuestas dinámicas
- matrices
- literales
- reportes

2) Asociaciones:

- administrador-profesor
- profesor-grupo
- grupo-alumno
- profesor-reactivos
- reactivos-respuestas
- reactivos-reportes-respuestasfijas
- reactivos-reportes-respuestasdinamicas
- reactivos-literales
- reactivos-matrices
- alumnos-reportes.

3) Tipos de Relaciones:

- “1” administrador tiene “n” administradores.
- “1” administrador tiene “n” profesores
- “1” profesor tiene “1” administrador
- “1” profesor tiene “n” grupos
- “1” grupo tiene “1” profesor
- “1” alumno tiene “1” grupo
- “1” grupo tiene “n” alumnos
- “1” profesor tiene “n” reactivos
- “1” reactivo por “n” profesor
- “1” reactivo tiene “n” respuestas
- “1” respuestafija tiene “1” reactivo y “1” reporte
- “1” respuestadinamica tiene “1” reactivo y “1” reporte
- “1” reactivo tiene “n” literales
- “1” reactivo tiene “1” matriz
- “1” alumno tiene “n” reporte
- “1” reporte tiene “1” alumno.

4) Atributos de Entidades:

Para la Entidad **Administrador** se tiene:

- “idadmin”: que es el número de Identificación que tiene el Administrador y es la llave primaria.
- “nombre”: que es el nombre (ó nombres) del Administrador.
- “apat”: que es el Apellido Paterno del Administrador.
- “amat”: que es el Apellido Materno del Administrador.
- “usuario”: que es el nombre de usuario que usa el Administrador para entrar al SIEL.

- “password”: que es la contraseña de cada Administrador para entrar a su Módulo en el SIEL.
- “email”: es el correo electrónico del Administrador.
- “verificación”: se envía un conjunto de caracteres para compararse en la Base de Datos y poder activar al usuario para que se pueda registrar por Internet. Al registrarse el usuario su cuenta queda inactiva hasta que se ingrese al vínculo que se envía al usuario.
- “activo”: una vez dado de alta se pone como este campo como activo ‘1’. No es funcional hasta activar la cuenta.

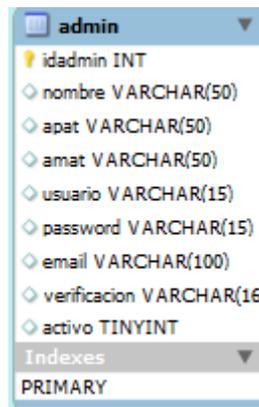


Figura 4.2. Entidad admin, contiene los datos del administrador.

Para la Entidad **Profesor** se tiene:

- “idprofesor”: que es el número de Identificación que tiene el Profesor y es la llave primaria.
- “nombre”: que es el nombre (o nombres) del Profesor.
- “apat”: que es el Apellido Paterno del Profesor.
- “amat”: que es el Apellido Materno del Profesor.
- “usuario”: que es el nombre que usa el Profesor para entrar al SIEL.

- “password”: que es la contraseña personal de cada Profesor para entrar a su Módulo en el SIEL.
- “email”: que es el correo electrónico del Profesor.
- “verificación”: se envía un conjunto de caracteres para compararse en la Base de Datos y poder activar al usuario para que se pueda registrar por Internet. Al registrarse el usuario su cuenta queda inactiva hasta que se ingrese al vínculo que se envía al usuario.
- “activo”: una vez dado de alta se pone como este campo como activo ‘1’. No es funcional hasta activar la cuenta.

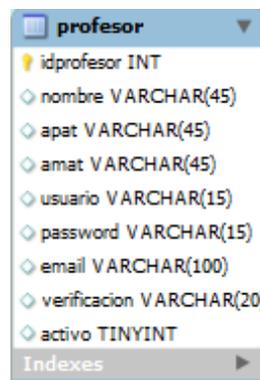


Figura 4.3 Muestra la Entidad Profesor.

Para la Entidad **Alumno** se tiene:

- “idalumno”: que es el número de Identificación que tiene el Alumno y es la llave primaria.
- “num_cuenta”: que es el Número de Cuenta de cada alumno.
- “nombre”: que es el nombre (o nombres) así como el apellido Paterno y Materno del Alumno.
- “usuario”: que es el nombre que usa el Alumno para entrar al SIEL.

- “password”: que es la contraseña personal de cada Alumno para entrar a su Módulo en el SIEL.
- “email”: que es el correo electrónico del Alumno.
- “intento_b”: que es el número de intentos realizados en el Examen Básico (2 oportunidades).
- “intento_a”: que es el número de intentos realizados en el Examen Avanzado (2 oportunidades).
- “grupo_idgrupo”: llave foránea a la entidad grupo.

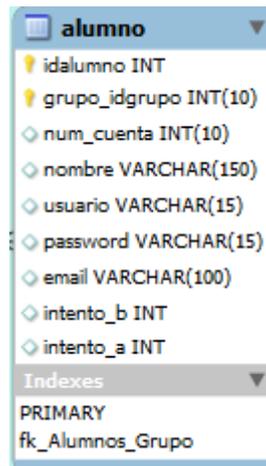


Figura 4.4 Entidad alumno, contiene los datos del alumno.

Para la Entidad **Grupo** se tiene:

- “idgrupo”: que es el número de Identificación que tiene el Grupo y es la llave primaria.
- “grupo”: que son los grupos de Circuitos Eléctricos.
- “horario”: que es el horario de la materia.
- “profesor_idprofesor”: llave foránea a la entidad Profesor.

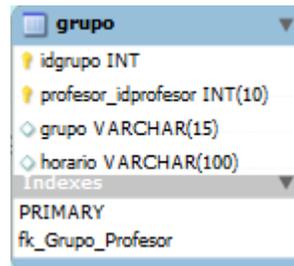


Figura 4.5. Entidad grupo.

Para la Entidad **Reactivos** se tiene:

- “idreactivo”: que es el número de Identificación que tiene el Reactivo y es la llave primaria.
- “enunciado”: que es el enunciado del reactivo de Circuitos Eléctricos.
- “Imagen”: que es la figura del Reactivo.
- “nivel”: que es el nivel de examen a desarrollar (Básico - Avanzado).
- “tipo”: que es el tipo de examen a desarrollar para (0) Básico ó (1) Avanzado.
- “tema”: que es el tema del examen de Circuitos Eléctricos.
- “puntos”: es el valor que tiene el Reactivo.
- “fecha”: que es la fecha en que se dio de alta el reactivo.
- “imagen_enunciado”: es la imagen del Reactivo.
- “profesor_idprofesor”: llave foránea a la entidad profesor.

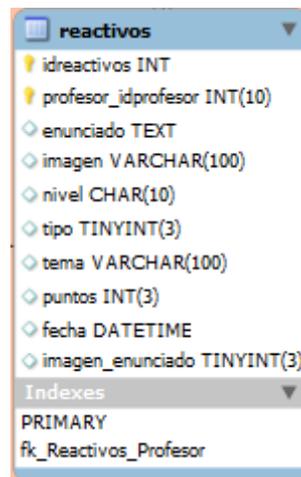


Figura 4.6 Entidad reactivos, contiene el banco de preguntas.

Para la Entidad **Reactivo Fijo** se tiene:

- “idreactivosfijos”: que es el número de Identificación que tiene el Reactivo Fijo y es la llave primaria.
- “opción1”: que es la opción 1 de respuesta.
- “opción2”: que es la opción 2 de respuesta.
- “opción3”: que es la opción 3 de respuesta.
- “opción4”: que es la opción 4 de respuesta.
- “opcion_correcta”: que es la opción Correcta de la respuesta del Reactivo Fijo.
- “reactivos_idreactivos”: llave foránea a la entidad reactivos.

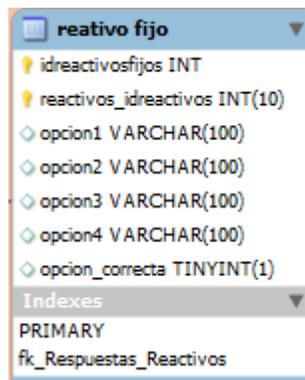


Figura 4.7. Entidad reactivo fijo, contiene las respuestas de los reactivos fijos.

Para la Entidad **respuestasfijas** se tiene:

- “idrespuestasfijas”: que es el número de Identificación que tienen las Respuestas Fijas de un Reactivo Fijo y es la llave primaria.
- “opcion_elegida”: que es la opción seleccionada por el Alumno.
- “puntaje”: es la calificación obtenida de este Reactivo Fijo.
- “reportes_idreportes”: llave foránea a la entidad reportes.
- “reactivos_idreactivos”: llave foránea a la entidad reactivos.

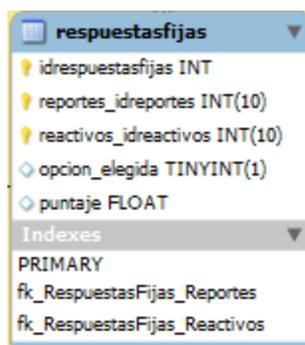


Figura 4.8. Entidad respuestasfijas, aquí se guarda la opción que elige el alumno en cada reactivo.

Para la Entidad **respuestasdinamicas** se tiene:

- “idrespuestasdinamicas”: que es el número de Identificación que tiene la Respuesta Dinámica de un Reactivo Dinámico y es la llave primaria.
- “opción1”: que es la opción 1 de la Respuesta Dinámica.
- “opción2”: que es la opción 2 de la Respuesta Dinámica.
- “opción3”: que es la opción 3 de la Respuesta Dinámica.
- “opción4”: que es la opción 4 de la Respuesta Dinámica.
- “opcion_elegida”: que es la respuesta seleccionada por el Alumno.
- “opcion_correcta”: que es la opción Correcta de la respuesta del Reactivo Dinámico.
- “puntaje”: es la calificación obtenida en este Reactivo Dinámico.
- “cadena_generadora”: es una cadena de caracteres obtenidos a partir de un algoritmo de mezcla y desorden de los caracteres que componen el nombre y el número de cuenta del alumno.
- “factor”: es un valor numérico por el que el valor ASCII¹ de cada carácter de la cadena generadora es multiplicado.
- “reportes_idreportes”: llave foránea a la entidad reportes.
- “reactivos_idreactivos”: llave foránea a la entidad reactivos.

¹ American Standard Code for Information Interchange.

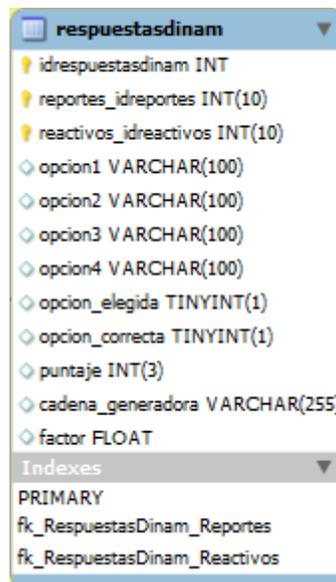


Figura 4.9. Entidad respuestasdinam, almacena las 4 respuestas generadas para cada reactivo.

Para la Entidad **matrices** se tiene:

- “idmatrices”: es la llave primaria.
- “nombre”: Identificador de la variable.
- “valor”: valor del identificador, se guarda la matriz en sí.
- “reactivos_idreactivos”: llave foránea a la entidad reactivos.

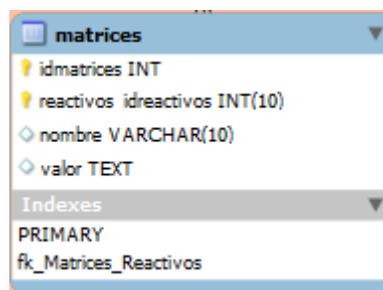


Figura 4.10. Entidad matrices, se guarda la cadena de texto que contiene las ecuaciones de malla o nodo del reactivo dinámico.

Para la Entidad **literales** se tiene:

- “idliterales”: es la llave primaria de la entidad.
- “literales”: son las variables que el problema tiene o las peticiones que el profesor hace. En un ejemplo de ecuaciones de nodo las variables serán las resistencias y las peticiones las fuentes de corriente.
- “tipo”: muestra si es una variable o petición. Para un caso donde se tengan las ecuaciones de nodo, las resistencias tienen el valor de 0, y las fuentes de corriente el valor de 1.
- “reactivos_idreactivos”: llave foránea a la entidad reactivos.

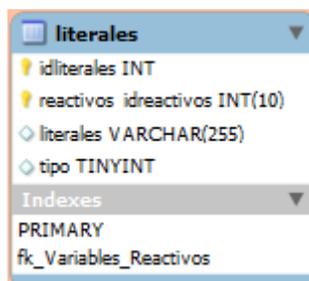


Figura 4.11. Entidad literales, tabla que guarda las variables que se usan en el reactivo.

Para la Entidad **reportes** se tiene:

- “idreportes”: que es el número de Identificación que tiene el Reporte del Examen y es la llave primaria.
- “calificación”: que es la calificación del Examen de Circuitos Eléctricos.
- “tiempo_resolucion”: que es el tiempo en que se realiza el Examen de Circuitos Eléctricos.
- “fecha”: que es la fecha en que se realiza el Examen de Circuitos Eléctricos.
- “tipo”: determina si es un examen Básico o Avanzado.
- “alumnos_idalumnos”: llave foránea a la entidad alumnos.

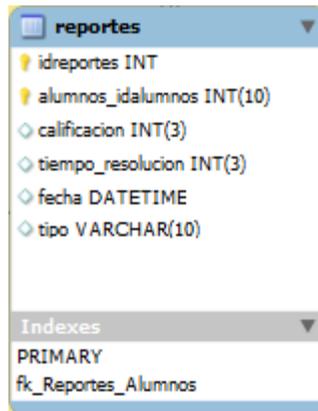


Figura 4.12. Entidad reportes, tabla que contiene la calificación del examen, tiempo de resolución, fecha de realización y tipo de examen.

En la figuras 4.16, 4.17. 4.18 y 419 se muestra el diagrama de Entidad-Relación del SIEL.

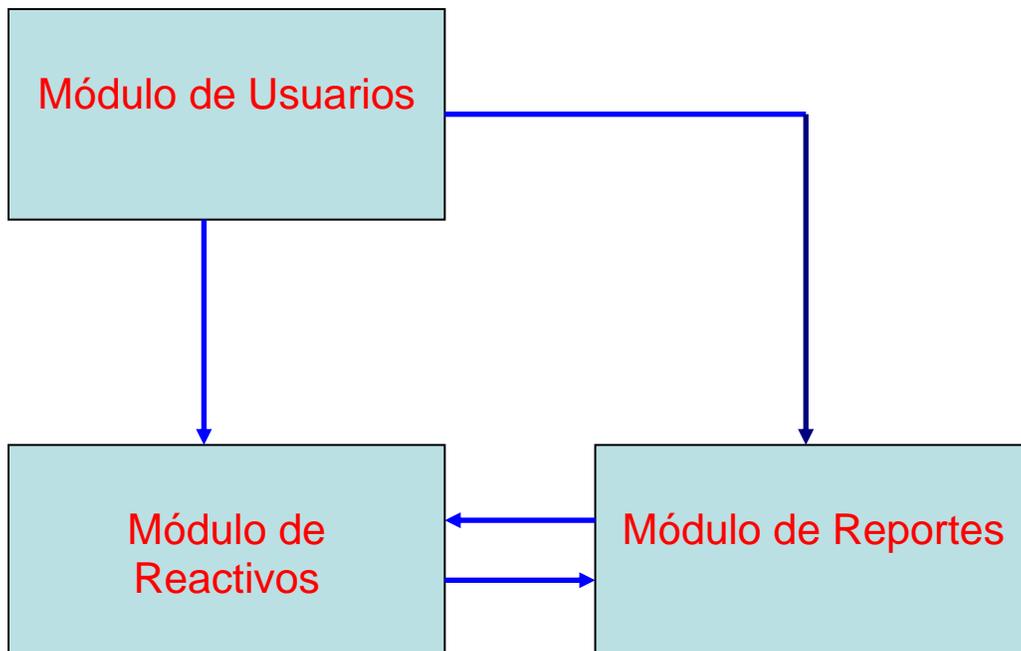


Figura 4.16. Diagrama Entidad-Relación a Nivel Módulos

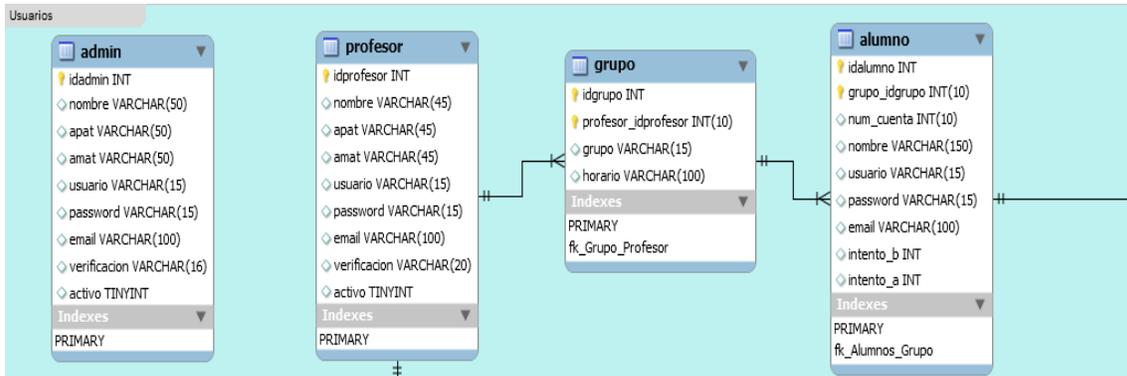


Figura 4.17. Diagrama Entidad-Relación a Nivel Módulo Usuarios

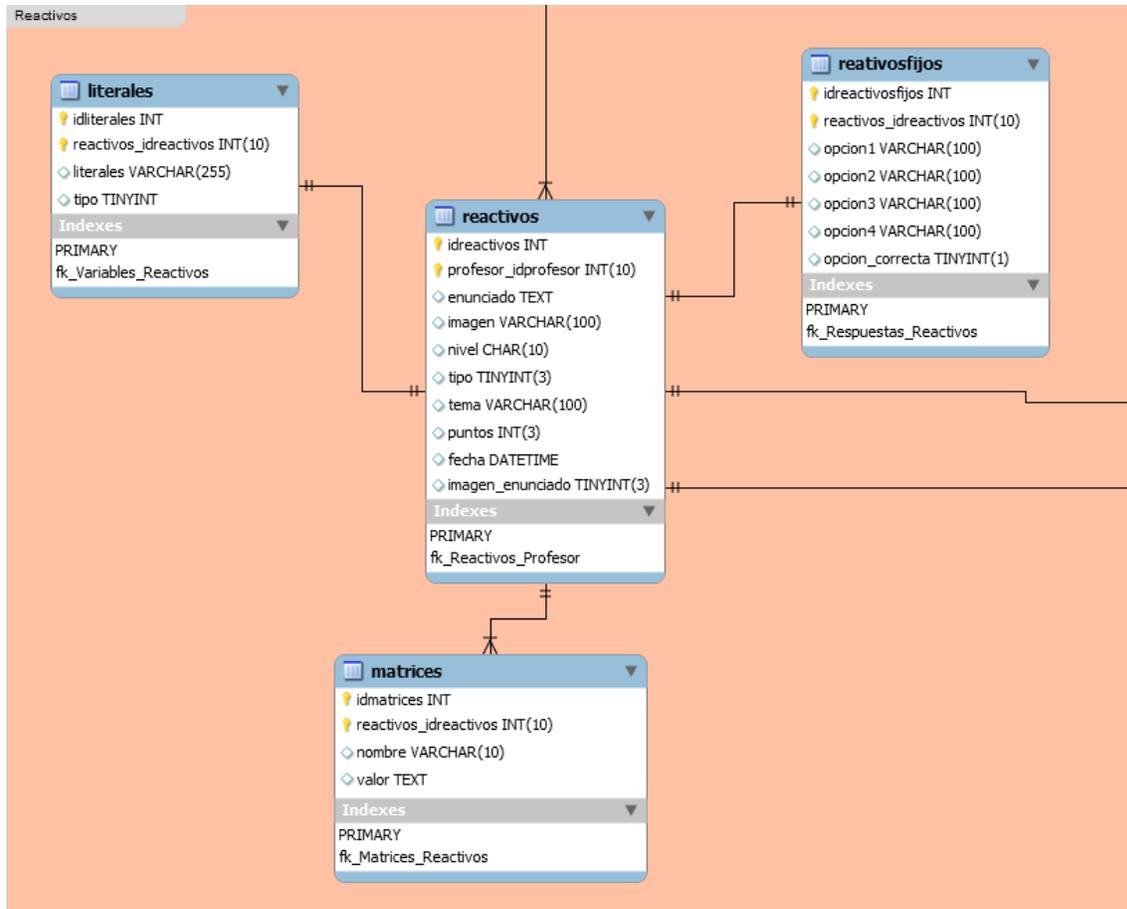


Figura 4.18. Diagrama Entidad-Relación a Nivel Módulo Reactivos

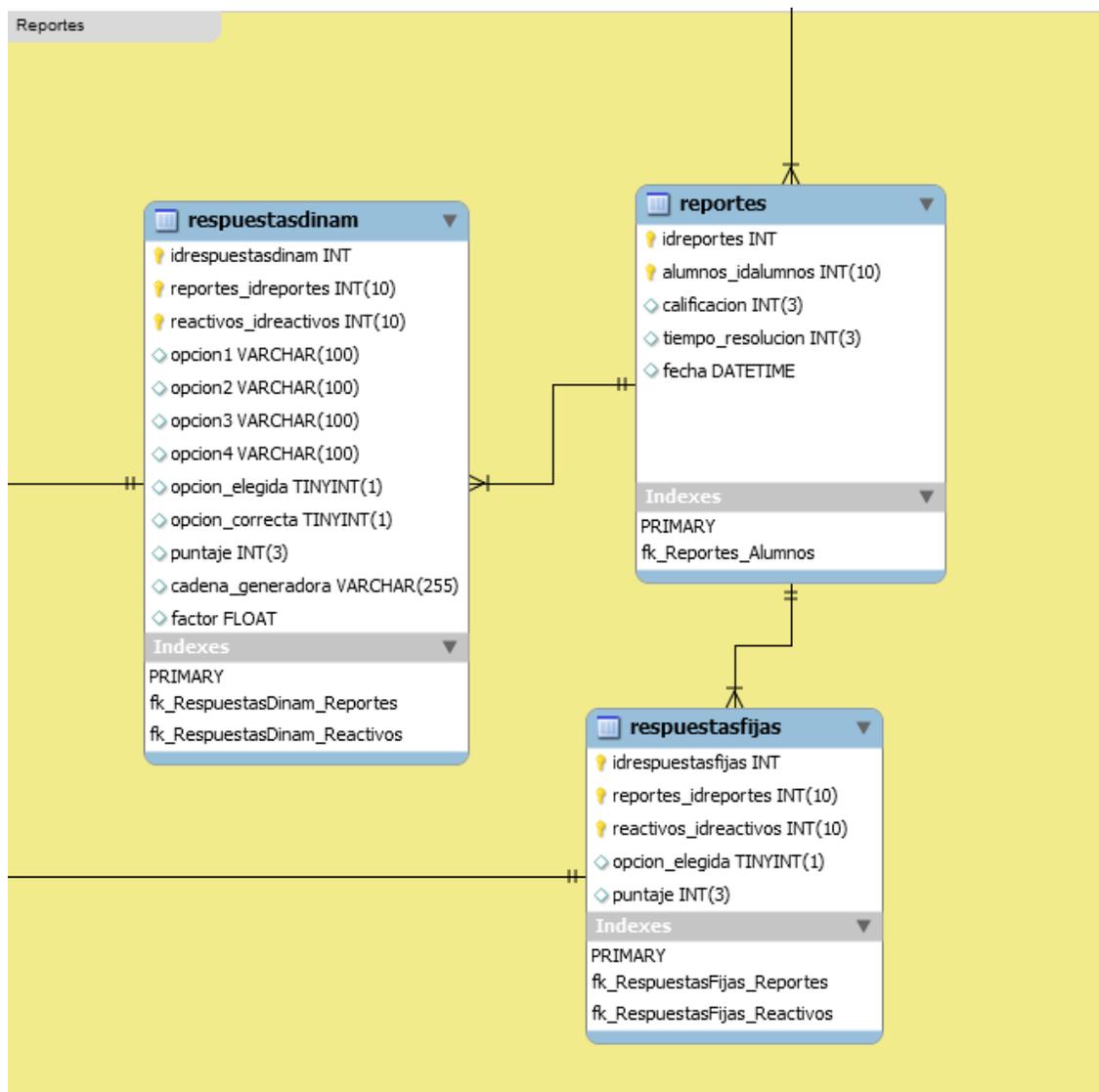


Figura 4.19. Diagrama Entidad-Relación a Nivel Módulo Reportes

4.3 IMPLEMENTACIÓN DEL SIEL

En la siguiente sección se detallan las interfaces que se desarrollaron de los principales Casos de Uso así como las funciones que se realizaron para llevar a cabo las funcionalidades de los módulos que componen el sistema.

Para este módulo se explican además de las interfaces de los Casos de Uso, la InterfazInicio, la cual representa la pantalla de inicio al sistema en donde se autentifica al usuario para que este tenga acceso a su módulo correspondiente.

Modulo Administrador:

- InterfazInicio (Tabla 4.1).
- InterfazFuncionesAdministrador (Tabla 4.2).
- InterfazConsultaAdministrador (Tabla 4.3).
- InterfazModificarAdministrador (Tabla 4.4).
- InterfazBajaAdministrador (Tabla 4.5).
- InterfazAltaAdministrador (Tabla 4.6).

Tabla 4.1. Se detalla la implementación del Caso de Uso: Autenticar Usuario.

Nombre de la Interfaz:	InterfazInicio.
Funciones:	Autenticar a los diferentes tipos de usuarios (Administrador/ Profesor/Alumno), para que ingrese al sistema en su respectivo módulo.
Desarrollo:	<p>Se despliega la página de inicio index.php, donde se tiene información general del desarrollo de la evaluación en línea.</p> <p>El usuario dependiendo de su nivel, podrá acceder al módulo correspondiente a su nivel, para eso presionará el link->vínculo correspondiente (Profesor, Alumno, Admin). Desplegándose la página correspondiente:</p> <ol style="list-style-type: none"> 1. loginadmin.php: se ejecuta la búsqueda en la tabla admin, comparando los campos usuario y password con los ingresados por el usuario, para verificar la veracidad de los datos. Una vez autenticado se despliega la página inicio_admin.php donde puede acceder a todas las funcionalidades del módulo de administrador.

	<p>2. loginprofesor.php: se ejecuta la búsqueda en la tabla profesor, comparando los campos usuario y password con los ingresados por el usuario, para verificar la veracidad de los datos. Una vez autenticado se despliega la página inicio_profesor.php donde puede acceder a todas las funcionalidades del módulo del profesor.</p> <p>3. loginalumno.php: se ejecuta la búsqueda en la tabla alumnos, comparando los campos usuario y password con los ingresados por el usuario, para verificar la veracidad de los datos. Una vez autenticado se despliega la página inicio_alumno.php donde puede acceder a todas las funcionalidades del módulo del alumno.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.2. Se detalla la implementación del Caso de Uso: Funciones de Administrador.

Nombre de la Interfaz:	InterfazFuncionesAdministrador.
Funciones:	Se despliega la página inicio_admin.php donde el administrador cuenta con un menú en el cual puede elegir las opciones a realizar (consulta/alta/baja).
Desarrollo:	<p>Se cuenta con un conjunto de vínculos, hacia las diferentes páginas donde puede realizar la acción deseada.</p> <ol style="list-style-type: none"> 1. Vínculo a la página consulta_admin.php: es una página que cuenta con un menú con las opciones donde puede elegir consultar, modificar o dar de alta a otro administrador. 2. Vínculo a la página nuevo_admin.php: aquí se ingresan los datos del nuevo administrador para ser guardados en la tabla admin. 3. A su vez se cuentan con los vínculos para consultar, modificar, bar de baja y alta a profesores.

Tabla 4.3. Se detalla la implementación del Caso de Uso: Consultar Información Administrador.

Nombre de la Interfaz:	InterfazConsultaAdministrador.
Funciones:	Dentro de esta página el administrador tendrá la opción de consultar la información de otro administrador, modificar dicha información y también dar de baja registros de la tabla admin.
Desarrollo:	<ol style="list-style-type: none"> 1. Al seleccionar el vínculo de consulta se despliega la página consulta_admin1.php, el cual despliega los administradores vigentes en una tabla. Se realiza una búsqueda en la tabla admin, para desplegar los campos de nombre, apat, amat, los tres anteriores concatenados para desplegar el nombre, así como el usuario y el password. 2. Se selecciona un registro para ver la información detallada, se despliega la pantalla consulta_admin2.php, se ejecuta la búsqueda en la tabla admin usando como parámetro el identificador del administrador del cual se quieren obtener la información en la tabla admin, se despliegan los campos de nombre, apat, amat, usuario, password y email.

Tabla 4.4. Se detalla la implementación del Caso de Uso: Consultar Información Administrador con Subflujo Modificar.

Nombre de la Interfaz:	InterfazModificarAdministrador.
Funciones:	Modificar los datos del administrador en la tabla admin.
Desarrollo:	<p>Para llegar a la página donde se realiza esta acción se sigue la siguiente secuencia de vínculos:</p> <ol style="list-style-type: none"> 1. En la pantalla inicio_admin.php, seleccionar el vínculo Consultar Administrador. 2. En la pantalla consulta_admin.php, seleccionar el vínculo Consulta Administrador.

	<ol style="list-style-type: none"> 3. En la pantalla consulta_admin1.php, se despliegan los administradores actuales, seleccionar el registro a modificar. 4. Se despliega la pantalla consulta_admin2.php, se despliegan los datos del registro del administrador seleccionado. 5. Una vez que se cambian los datos, se envían a la página consulta_admin3.php, la cual procesa los datos y realizar una actualización de los datos en la tabla admin, con los campos que se hayan cambiado.
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.5. Se detalla la implementación del Caso de Uso: Dar Baja Administrador.

Nombre de la Interfaz:	InterfazBajaAdministrador.
Funciones:	Borrar de la tabla admin, el registro del administrador seleccionado.
Desarrollo:	<p>Para realizar la acción deseada se sigue la siguiente secuencia de vínculos:</p> <ol style="list-style-type: none"> 1. En la página inicio_admin.php se selecciona el vínculo de Consultar Administrador. 2. En la página consulta_admin.php se selecciona el vínculo de Dar de Baja Administrador. 3. En la página BajaAdmin.php se realiza una búsqueda en la tabla admin, para desplegar todos los administradores presentes en la base de datos, en cada registro se cuenta con un control checkbox, el cual nos permite seleccionar el registro a eliminar. 4. Una vez seleccionado el registro, se presiona el control que envía el identificador del registro del administrador a la

	<p>página encargada de realizar la secuencia de sql que permite borrar el registro de la tabla.</p> <p>5. En la página BajaAdmin1.php, se obtiene el identificador del registro del administrador, lo cual permite hacer una búsqueda de dicho registro para ser borrado de la tabla admin.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.6. Se detalla la implementación del Caso de Uso: Dar Alta Administrador.

Nombre de la Interfaz:	InterfazAltaAdministrador.
Funciones:	Dar de alta un nuevo registro con los datos del nuevo administrador en la tabla admin.
Desarrollo:	<ol style="list-style-type: none"> 1. En la página inicio_admin.php se selecciona el vínculo Registrar nuevo Administrador. 2. En la página nuevo_admin.php, se ingresan los datos del nuevo administrador, nombre, apellido paterno, apellido materno, usuario, password y email. 3. La página nuevo_admin1.php, es la encargada de recibir los datos del nuevo registro del administrador y crear un nuevo registro en la tabla admin, al guardar la información en los campos (nombre, apat, amat, usuario, password, e-mail).

Módulo Profesor

En este módulo se contemplan tres interfaces principales para desarrollar sus respectivos Casos de Uso, dar de alta los reactivos, consultar los reportes y dar de

alta un grupo. Para el caso de consulta de reportes se explican las funciones que se implementaron para llevar a cabo la búsqueda de la información de los reportes generados por el alumno tanto al hacer el examen básico (Tabla 4.9); como el avanzado (Tabla 4.10).

- InterfazEditarReactivo (Tabla 4.7).
- InterfazConsultaReporte (Tabla 4.8).
- InterfazAltaGrupo (Tabla 4.11).

Tabla 4.7. Se detalla la implementación del Caso de Uso: Editar Reactivo.

Nombre de la Interfaz:	InterfazEditarReactivo.
Funciones:	Dar de alta reactivos fijos así como dinámicos en la base de datos.
Desarrollo:	<p>Para dar de alta reactivos fijos:</p> <ol style="list-style-type: none"> 1. En la página inicio_profesor.php seleccionar el vínculo de Registrar nuevo Reactivo. 2. En la página nuevo_reactivo.php seleccionar el vínculo Reactivo Fijo. 3. En la página ReactivoFijo.php, se lleva a cabo el ingreso de los datos del reactivo así como sus respectivas respuestas. Diagrama (archivo de imagen que contiene el circuito del problema a realizar), enunciado de la pregunta, Nivel (Básico /Avanzado), Tema, Puntos, Respuesta1, Respuesta2, Respuesta3, Respuesta4. 4. Se envía la información a la página ReactivoFijo1.php, la cual se encarga de guardar la información en las tablas reactivos (información de la pregunta) y reactivofijo (información concerniente a las respuestas). <p>Para dar de alta reactivos dinámicos:</p> <ol style="list-style-type: none"> 1. En la página nuevo_reactivo.php seleccionar el vínculo de

	<p>Reactivo Dinámico.</p> <ol style="list-style-type: none">2. En la página ReactivoDinamico.php se selecciona la imagen que corresponde a la pregunta que se va a desplegar, se indica el número de renglones y columnas de la matriz de admitancia/impedancia y el tipo de vector Corriente/Voltaje.3. En la página ReactivoDinámico1.php se ingresa la información que se refiere al enunciado de la pregunta, Nivel (Básico /Avanzado), Tema, Puntos, Matriz de Datos (cadena de texto que contiene la información de las ecuaciones de nodo/malla), tipo de matriz (Admitancia/Impedancia), las expresiones del Vector Corriente/Voltaje, número de resistencias y fuentes presentes en el circuito.4. En la página ReactivoDinamico2.php, se guarda la información enviada, en la tabla reactivos, se guarda el identificador del profesor que ingreso el reactivo, el enunciado de la pregunta, en nombre de la imagen del circuito del problema, nivel, tipo, tema, puntos y la fecha en que se ingreso el reactivo. En la tabla matrices se guarda el identificador del reactivo, el tipo de matriz Admitancia/Impedancia, y la cadena de texto que contiene las ecuaciones de malla/nodo del sistema. En la tabla de literales se guarda secuencialmente el número de resistencias y fuentes presentes en el reactivo (R1, R2, IS1, IS2, etc.).
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.8. Se detalla la implementación del Caso de Uso: Consulta Reporte de Alumno.

Nombre de la Interfaz:	InterfazConsultaReporte.
Funciones:	Ver cuales son los alumnos que han presentado exámenes y revisar sus respectivos resultados.
Desarrollo:	<ol style="list-style-type: none"> 1. En la página inicio_profesor.php, seleccionar el vínculo Consultar Reportes. 2. En la página ProfesorReporte.php, se despliega la lista de todos los alumnos que han presentado exámenes. Se realiza primero una búsqueda en la tabla reportes para obtener una lista de los identificadores de los alumnos que han hecho exámenes, con esa lista de identificadores de alumnos en la tabla alumnos se busca la información personal de cada uno. De la lista se selecciona el alumno del cual se quiere revisar sus reportes. 3. En la página ProfesorReporte1.php, se busca en la tabla reportes y alumnos, la información del alumno del cual se desea conocer los informes, como parámetro de búsqueda se utiliza el identificador del alumno. De la tabla alumnos, se selecciona el nombre, de la tabla reportes tipo (tipo de examen), calificación, fecha y el identificador del reporte. 4. En la página ProfesorReporte2.php, de acuerdo al identificador de reporte e identificador de alumno, se ejecutan las funciones que despliegan los reportes de acuerdo al tipo de examen. 5. Para un examen Básico se ejecuta la función RProfesorBasico(\$reporteTipo,\$id). 6. Para un examen Avanzado se ejecuta la función RProfesorAvanzado(\$reporteTipo,\$id).

Tabla 4.9 Se detalla la implementación de la Función RProfesorBasico.

Nombre de la Función:	RProfesorBasico(\$reporteTipo,\$id).
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$reporteTipo: identificador de reporte. 2. \$id: identificador de alumno.
Parámetros de salida:	No se cuenta con parámetros de salida.
Objetivos:	<p>En base al identificador de alumno y de reporte realizar una búsqueda para desplegar los datos del alumno, los datos de las preguntas de acuerdo al identificador de reporte y la calificación obtenida.</p> <ol style="list-style-type: none"> 1. Buscar en la tabla alumnos de acuerdo al identificador del alumno, los datos específicos. 2. Realizar una búsqueda en las tablas reactivos y respuestasfijas, para obtener los datos del reactivo así como la respuesta elegida por el alumno y el puntaje obtenido. 3. De la tabla reativofijo, obtener las respuestas asociadas al reactivo y desplegar únicamente el que tenga el indicador de la respuesta elegida. 4. De la tabla reportes obtener la calificación total del examen.

Tabla 4.10. Se detalla la implementación de la Función RProfesorAvanzado.

Nombre de la Función:	RProfesorAvanzado(\$reporteTipo,\$id).
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$reporteTipo: identificador de reporte. 2. \$id: identificador de alumno.
Parámetros de salida:	No cuenta con parámetro de salida.
Objetivo:	Buscar la información en las tablas respectivas de acuerdo al

	<p>identificador de reporte y alumno, para desplegar el reporte del examen avanzado:</p> <ol style="list-style-type: none"> 1. Se busca en la tabla alumnos la información de quien realizo el examen de acuerdo con el identificador. 2. En la tabla reactivos y respuestasdinam se busca la información del reactivo y la respuesta seleccionada de los campos enunciado, imagen, opcion1, opcion2, opcion3, opcion4, opcion_elegida, opcion_correcta, cadena_generadora, puntaje
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.11. Se detalla la implementación del Caso de Uso: Alta Grupo.

Nombre de la Interfaz:	InterfazAltaGrupo.
Funciones:	Leer el archivo con formato xml, que contiene los datos de todos los alumnos del grupo, guardar la información en la tabla alumnos.
Desarrollo:	<ol style="list-style-type: none"> 1. En la página inicio_profesor.php, seleccionar el vínculo Registrar nuevo Grupo. 2. En la página nuevo_grupo.php se tiene un control fileupload, que nos permite subir al servidor el archivo con formato xml que contiene la información de los alumnos del grupo. 3. En la página nuevo_grupo1.php, se lee el archivo renglón por renglón, guardando los registros nuevos en la tabla alumnos, validando el número de cuenta ya existente, para evitar repetir alumnos.

Módulo Alumno:

- Implementación de la Interfaz Realizar Examen:

Existen dos tipos de examen Básico y Avanzado. El primero (Básico) va a contar con respuestas fijas, es decir las respuestas ya están guardadas en la Base de Datos, en cada examen el sistema va a presentar estas respuestas en forma aleatoria.

La explicación de la implementación de la interfaz del examen Básico se detalla en la Tabla 4.12. En donde se mencionan las funciones que se implementaron para llevar a cabo el despliegue del examen. En primer lugar la obtención de las preguntas que se van a desplegar en el examen seleccionándolas del banco de datos (Tabla 4.13), actualizando el número de intentos del alumno, generando el identificador de reporte del examen en la Base de Datos y guardando los identificadores de las preguntas (Tabla 4.14). Una vez con los identificadores de las preguntas se busca en las tablas de reactivos fijos la información concerniente a cada uno de los reactivos (Tabla 4.15), por cada pregunta se despliegan las cuatro respuestas del reactivo utilizando *radio buttons* (botones radiales) (Tabla 4.16). Para el caso de que el alumno una vez en la sesión de examen, recargara la página, el sistema detecta esta acción volviendo a desplegar las preguntas previamente guardadas en la tabla de reportes buscando su respectivo identificador en lugar de generar un nuevo conjunto de preguntas (Tabla 4.17). Una vez que el alumno finaliza el examen presiona el botón “Finalizar” para que se despliegue el reporte del examen previamente realizado, se guardan las respuestas del alumno a cada una de sus respectivas preguntas, se guarda la calificación de cada reactivo, así como la calificación total de la evaluación (Tabla 4.18). Posteriormente se despliega la información guardada y que corresponde al identificador de reporte que se acaba de hacer (Tabla 4.19).

El examen Avanzado cuenta con reactivos dinámicos, en cada examen el sistema genera valores diferentes para las variables que intervienen en el problema. Las tablas donde está guardada la información de los reactivos dinámicos son:

1. **Reactivos:** información general como enunciado, nombre de diagrama, nivel, tipo, tema, puntos, etc.
2. **Matrices:** aquí se va a guardar la información de las ecuaciones de malla o nodo según sea el caso del problema. Las matrices de Admitancia, Resistencia van a tener el siguiente formato para ser guardada en la base de datos: `matriz(fila(..., ...), fila(..., ...))`, lo anterior para el caso de una matriz de 2x2. Y los vectores de corriente/voltaje van a tener el siguiente formato: `Corriente(IS1, IS2)`.
3. **Literales:** en esta tabla se guarda la información de las variables del problema (resistencias, voltajes, corrientes).

Cuando un reactivo dinámico sea seleccionado, elegirá sus respectivos valores de matriz y vector, para formar la matriz de admitancia/resistencia y el vector de corriente/voltaje según sea el caso, y formar así su sistema de ecuaciones de nodo o malla, el cual se resuelve por el método iterativo de Jacobi. Esto se realiza cuatro veces para formar respuestas que el alumno debe escoger, aleatoriamente se escoge una para que sea la correcta.

La explicación de las funciones que se realizaron para implementar la interfaz del examen Avanzado se detalla en la Tabla 4.20. De la tabla 4.21 a la tabla 4.40, se explican las funciones que hacen posible que se lleve a cabo el despliegue del examen Avanzado. Entre las funciones que podemos mencionar esta la función que lleva a cabo la creación del registro donde se guarda el identificador del reporte, así como la inserción en la Base de Datos de los identificadores, de los reactivos dinámicos que se van a desplegar en el examen del alumno (Tabla 4.22). Una vez guardados los identificadores se procede a desplegar cada uno de los reactivos (Tabla 4.23). Se obtiene la cadena de texto que contiene las

ecuaciones de la matriz de admitancia/inductancia (Tabla 4.24), para posteriormente obtener la cadena de texto que contiene la cadena de texto que contiene el vector de corriente/voltaje según sea el caso (Tabla 4.25). Se asignan los valores de las resistencias (Tabla 4.30), a continuación se hace lo mismo con la asignación de valores de las fuentes involucradas en el circuito (Tabla 4.31). Para el despliegue de los valores de las variables, se escoge aleatoriamente dentro de un bucle iterativo de 4 (número de respuestas de las cuales el alumno debe elegir la correcta Tabla 4.32). Dada la cadena de texto que contiene la información de la ecuaciones del sistema, se procede a obtener cada uno de estos elementos para crear un vector que contiene la expresión de cada uno de ellos, para posteriormente determinar las variables que están presentes, buscar los valores y realizar la suma algebraica (Tabla 4.33), para obtener como parámetro de salida la matriz que contiene los valores de dicha ecuación. De manera análoga de la cadena de texto que contiene la información del vector de corriente/voltaje se obtienen cada uno de los elementos para identificar las fuentes que intervienen, asignarles el valor y realizar la suma algebraica, obteniendo el vector con los valores numéricos (Tabla 4.34). Con los valores de la matriz y el vector se procede a resolver el sistema de ecuaciones usando el método iterativo de Jacobi (Tabla 4.35). Se realizan cuatro respuestas con los respectivos valores del sistema de ecuaciones, se guardan en la Base de Datos en forma de cadena de texto, también se guarda la opción correcta, la cual es elegida aleatoriamente (Tabla 4.36). Las cuatro respuestas son desplegadas en forma de *radio buttons* (botones radiales) para que el alumno pueda elegir una (Tabla 4.37). En caso de que el alumno recargue la página no se vuelven a buscar otro grupo de preguntas, el sistema como en el examen Básico detecta que se ha realizado una recarga para que busque en la Base de Datos las preguntas guardadas de acuerdo con el identificador de reporte (Tabla 4.38). Para terminar el examen el alumno oprime el botón “Finalizar”, se pasa a la pantalla de reportes donde en primer lugar se guardan la opción elegida del alumno de cada respuesta los cuales se actualizan

en la Base de Datos para su posterior consulta (Tabla 4.39). Como último paso se despliega la calificación obtenida en cada reactivo así como la calificación final obtenida por el alumno (Tabla 4.40).

Para los exámenes se crearon las siguientes variables de sesión: `$_SESSION['examen']` y `$_SESSION['reporte']`. La primera no permite que se guarden varias veces las preguntas seleccionadas para el examen, provocando la creación de reportes con datos incompletos al realizar una actualización o recarga de la página que muestra las preguntas del examen. Lo anterior se consigue evaluando el valor de la variable al ejecutarse por primera vez la página, ya que al visualizarse tendrá el valor por default de falso (*FALSE*), ejecutándose por completo el código cambiará al valor de verdadero (*TRUE*), de esta manera si el usuario solicita actualizar o recargar la página ya no realiza la ejecución del código. Por lo que respecta a la segunda variable tiene la función de dar más seguridad en el envío de la información que se hace de página a página, en esta variable se guarda el identificador (idreportes) del reporte que contiene la calificación total entre otros datos, misma que es transparente al alumno.

Para asegurar que los valores asignados a las variables que participan en cada uno de los problemas dinámicos sean diferentes, se desarrolló un algoritmo muy sencillo basado en el número de cuenta y el nombre del alumno. Se garantiza que los valores serán diferentes aún en el caso de que el mismo alumno entre al sistema varias veces, lo último debido a que este procedimiento cuenta con una rutina de mezcla y además, los valores se ven modificados por un factor que multiplica el código ASCII de cada carácter.

La manera en que desarrolló el algoritmo en lenguaje coloquial es el siguiente:

Función de mezcla.

Inicialmente se cuenta con el nombre del alumno y su número de cuenta, ambos datos deben ser ingresados con cadenas de texto. Los caracteres de la cadena

del nombre son duplicadas y almacenadas en la misma cadena, por lo que si el alumno se llama “Juan Gutiérrez”, la cadena pasa a “JuanGutiérrezJuan Gutiérrez”, lo anterior con la idea de generar un número considerable de posibles valores. Se cuentan el número de caracteres de la cadena de número de cuenta y se hace lo mismo con la cadena modificada del nombre, se divide el número de caracteres de la cadena del nombre entre el número de caracteres de la cadena del número de cuenta, el valor obtenido se redondea hacia abajo, dicho valor servirá como intervalo para seccionar la cadena del nombre en rangos y colocar, uno a uno cada número de la cadena del número de cuenta, así por ejemplo si el valor es 3, la cadena que se obtiene de “JuanGutiérrezJuanGutiérrez” y el número de cuenta “098745632” será “Jua0n G9uti8érr7ezJ4uan5 Gu6tié3rre2z”.

La cadena generada hasta este punto será la misma cada ocasión que el alumno ingrese al sistema, entonces se requiere hacer que no sea así, por lo que se aplica una función que la desordena de manera aleatoria, manteniendo los mismos caracteres pero en posiciones diferentes, a esta cadena se le nombra “Cadena generadora”.

Función de conversión de texto a número.

La cadena generadora es modificada a valores numéricos, para hacer lo anterior se recorre toda la cadena, convirtiendo cada caracter a su respectivo valor ASCII, el valor obtenido es entonces multiplicado por un factor común para toda la cadena, el factor se obtiene de manera aleatoria en un rango de entre 10 y 15. El resultado de la multiplicación es redondeado hacia arriba y almacenado en un vector, por lo que al final el vector tendrá una longitud igual al número de caracteres de la cadena generadora, este vector será el que ofrecerá los valores a las variables involucradas en el problema.

Tabla 4.12. Se muestran las funciones principales para implementar el Caso de Uso: Realizar Examen (Básico).

	Objetivo	Funciones o Instrucciones
1.-	El sistema despliega las opciones que tiene el alumno para cada tipo de examen, y solo permite realizar aquellos que tiene menos de dos intentos.	<p>Se despliega pantalla iniciar_exam.php</p> <p>Se busca en la base de datos, el número de intentos de cada examen.</p> <p>En base a esto se hacen visibles las opciones de los exámenes a realizar.</p>
2.-	Se genera el examen (Básico).	<p>Se despliega pantalla Pregunta1_SIEL.php</p> <p>Se obtiene la combinación de las preguntas:</p> <p>\$preguntas=Combinacion(\$tipo);</p> <p>Se modifica el número de intentos de exámenes en la base de datos, se crea el reporte y se guardan las preguntas en la Base de Datos:</p> <p>InsertBasico (\$id,\$basico,\$preguntas,\$tipo);</p> <p>Se despliegan las preguntas:</p> <p>PreguntaBasico(\$pregunta, \$indice);</p> <p>Se hace un reordenamiento aleatorio y se despliegan las cuatro respuestas de la pregunta.</p> <p>DespliegueRadio1 (\$radio,\$respuesta);</p> <p>Si el alumno hace una recarga de la página (se vuelve a desplegar la pantalla Pregunta1_SIEL.php):</p> <p>Se despliegan las preguntas que ya están guardadas en la base de datos:</p> <p>PreguntaBasicoPost(\$idreporte);</p>

		Se envían las respuestas al formulario de reporte.
3.-	Se genera el reporte.	Se despliega la pantalla Reporte.php ó Reportel.php, según sea el caso: Se reciben los valores de las respuestas y se guardan en la base de datos: RespuestaBasico(\$idreporte,\$respuesta); Se despliegan los datos del reporte: ReporteBasico(\$idreporte, \$respuesta); Se termina la sesión del alumno.

Las funciones anteriores se aplican para el examen Básico las cuales se detallan a continuación:

Tabla 4.13. Se detalla la Función Combinación.

Nombre de Función	\$preguntas=Combinación(\$tipo)
Parámetros de entrada	La variable de entrada en el sistema se llama \$tipo, la cual tiene por valores Básico, indica el tipo de examen que se va a hacer.
Parámetros de salida	Es arreglo de cuatro elementos, los cuales contienen los identificadores de las preguntas que se seleccionaron para hacer el examen.
Objetivo	<ol style="list-style-type: none"> 1. Se seleccionan todos los reactivos que cumplan con la condición que sean Básicos. 2. Se crean combinaciones de 4 elementos, que contienen los identificadores de los reactivos, de los cuales aleatoriamente se selecciona una combinación, el cual es el parámetro de salida.

Tabla 4.14. Se detalla la Función InsertBasico.

Nombre de Función		InsertBasico (\$id,\$basico,\$preguntas,\$tipo)
Parámetros de entrada:	de	Esta función tiene 4 parámetros de entrada: <ol style="list-style-type: none"> 1. \$id: identificador del alumno. 2. \$basico: número de intento actual. 3. \$preguntas: arreglo con los identificadores de las preguntas seleccionadas. 4. \$tipo: tipo de examen .
Parámetros de salida:		No cuenta con parámetros de salida.
Objetivos:		<ol style="list-style-type: none"> 1. Hace la sumatoria en uno del intento de examen que se lleve. 2. Se crea el registro de reporte en la tabla de reportes, insertando los datos, (identificador de alumno, fecha actual, tipo de examen) 3. Se guardan los identificadores de las respuestas junto con el identificador del reporte en la tabla respuestasfijas. 4. Se crea la variable de sesión cuyo contenido es el identificador del reporte.

Tabla 4.15. Se detalla la Función PreguntaBasico.

Nombre de Función:		PreguntaBasico(\$pregunta, \$indice)
Parámetros de entrada:	de	<ol style="list-style-type: none"> 1. Arreglo de 4 elementos que indica los identificadores de las preguntas. 2. Contador que nos indica el número de la pregunta que se va a desplegar.
Parámetros de salida:		No cuenta con parámetros de salida.
Objetivos:		<ol style="list-style-type: none"> 1. Se realiza una búsqueda de información (query) del

	<p>reactivo que se va desplegar. Dicha información incluye enunciado del problema, nombre del diagrama que se va a desplegar, las cuatro respuestas de este reactivo se guardan en un arreglo.</p> <ol style="list-style-type: none"> 2. Se despliega el número de la pregunta con la información del reactivo. 3. Se hace un reordenamiento aleatorio de las respuestas que se encuentran en el arreglo, para su posterior despliegue. 4. Se llama a la función <code>DespliegueRadio1 (\$radio,\$respuesta)</code>; para que haga el despliegue de las respuestas en forma de radio buttons (botones radiales).
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.16. Se detalla la Función `DespliegueRadio1`.

Nombre de Función:	<code>DespliegueRadio1 (\$radio,\$respuesta)</code>
Parámetros de entrada:	<ol style="list-style-type: none"> 1. El parámetro <code>\$radio</code>, sirve para llenar a su vez el parámetro <code>Name</code> del control <code>html radio button</code>. 2. <code>\$respuesta</code> es un arreglo de cuatro elementos que contiene las respuestas al reactivo en cuestión. Se despliegan 4 controles <code>radio buttons</code> (botones radiales), a cuyo parámetro <code>VALUE</code> se le asignará cada uno de los valores del arreglo.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivos:	Mediante un ciclo iterativo <code>for</code> , se despliegan las cuatro respuestas del reactivo en controles <code>radio buttons</code> (botones radiales).

Tabla 4.17. Se detalla la Función PreguntaBasicoPost.

Nombre de Función:	PreguntaBasicoPost(\$idreporte)
Parámetros de entrada:	\$idreporte: identificador del reporte actual.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivos:	<ol style="list-style-type: none"> 1. Con el identificador \$idreporte, buscar la información referente a los identificadores de preguntas asociados a ese reporte en la tabla respuestasfijas. 2. Una vez obtenidos los identificadores de preguntas asociados a ese \$idreporte, buscar la información de ese reactivo (enunciado, nombre de diagrama, respuestas de ese reactivo).

Tabla 4.18. Se detalla la Función RespuestaBasico.

Nombre de Función:	RespuestaBasico(\$idreporte,\$respuesta)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. Identificador de reporte \$idreporte. 2. Arreglo que contiene las 4 respuestas de las preguntas del examen.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivos:	<ol style="list-style-type: none"> 1. En la página que va a desplegar el reporte se guardan los parámetros enviados en el examen en la variable \$respuesta. 2. Se hace una búsqueda para saber cuales son los identificadores de pregunta asociados al identificador de reporte \$idreporte y obtener la respuesta correcta asociada a las preguntas. 3. Verificar si la respuesta seleccionada por el alumno corresponde a la correcta.

	<p>4. En base a esto se asigna la calificación a cada pregunta, y por otro lado se va haciendo una sumatoria para obtener la calificación total.</p> <p>5. Se modifica la tabla respuestasfijas guardando la opción que selecciono el alumno para esa pregunta así como su calificación individual.</p> <p>6. Se modifica la tabla reportes guardando la calificación total del examen.</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.19. Se detalla la Función ReporteBasico.

Nombre de Función:	ReporteBasico(\$idreporte, \$respuesta)
Parámetros de entrada:	<ol style="list-style-type: none"> Identificador de reporte \$idreporte. Arreglo que contiene las 4 respuestas de las preguntas del examen.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivos:	<ol style="list-style-type: none"> Se busca la información a desplegar en las tablas reactivos y respuestasfijas. Se presentan las preguntas numeradas con la calificación individual y la calificación total.

Tabla 4.20. Muestra las funciones principales para generar un examen Avanzado.

	Objetivo	Funciones o Instrucciones
1.-	Generar examen Avanzado.	<p>Se despliega en pantalla ExamAvanzado.php.</p> <p>Se actualiza el número de intentos: SetIntentos(\$intento,\$id)</p> <p>Se seleccionan las preguntas y crea el nuevo registro con los datos del reporte: \$idreporte=ReporteAvanzado(\$id,\$reactivos)</p>

		<p>Se crea la variable de sesión ['reporte'] que contiene el identificador de reporte.</p> <p>Se despliegan las preguntas: DesplieguePregunta(\$pregunta, \$indice,\$idreporte)</p>
2.-	Generar reporte.	<p>Se despliega la página ReporteDinam.php.</p> <p>Se reciben los valores de las respuestas y se guardan en la base de datos: AlumnoRespuestas(\$respuesta, \$idreporte)</p> <p>Se despliega el reporte con los datos del examen realizado: ResutadoReporte(\$idreporte)</p>

Tabla 4.21. Se detalla la Función SetIntentos.

Nombre de Función:	SetIntentos(\$avanzado,\$id)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$avanzado: número de intentos en el examen avanzado. 2. Identificador del alumno.
Parámetros de salida:	No cuenta con parámetro de salida.
Objetivos:	Se hace una búsqueda del registro del alumno, para actualizar el número de intentos en el examen avanzado.

Tabla 4.22. Se detalla la Función ReporteAvanzado.

Nombre de Función:	\$idreporte=ReporteAvanzado(\$id,\$reactivos)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$id: identificador del alumno. 2. \$reactivos: arreglo que contiene los identificadores de las preguntas del examen.
Parámetros de salida:	\$idreporte: identificador con el número del reporte.
Objetivos:	<ol style="list-style-type: none"> 1. En la tabla reportes se crea el nuevo registro con los datos del identificador del alumno, tipo de examen

	(avanzado), fecha actual.
	2. En la tabla respuestasdinam se crea un nuevo registro con los identificadores de las preguntas del examen y el identificador del reporte.

Tabla 4.23. Se detalla la Función DatosPregunta.

Nombre de Función:	\$row=DatosPregunta(\$id_reactivos)
Parámetros de entrada:	\$id_reactivos: identificador de la pregunta.
Parámetros de salida:	\$row: registro con los datos de la búsqueda. 1. Nombre de la imagen del problema. 2. Enunciado del problema. 3. Matriz asociado al problema. 4. Nombre de la matriz(Admitancia/Impedancia).
Objetivos:	Hacer una búsqueda en las tablas reactivos y matrices para encontrar los datos necesarios.

Tabla 4.24. Se detalla la Función CadenaMatriz.

Nombre de la función:	\$MatrizValor = CadenaMatriz(\$cadena)
Parámetros de entrada:	\$cadena: cadena de texto que contiene las ecuaciones de nodo o malla dependiendo del problema con el siguiente formato: matriz(fila(..., ...), fila(..., ...))
Parámetros de salida:	\$MatrizValor: vector que contiene las expresiones a evaluar de las ecuaciones de nodo o malla. Ejemplo: $\frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3}$
Objetivos:	1. De la cadena que contiene las ecuaciones, se realiza una subdivisión usando como delimitador la cadena fila, obteniendo con esto el número de renglones.

	<p>2. Posteriormente dependiendo del número de renglones que se obtengan, a cada uno se realiza otra subdivisión usando como delimitador el carácter ‘,’; para obtener cada uno de los elementos de la matriz de admitancia/resistencia, y guardarlos en un vector de nombre \$MatrizValor.</p> <p>3. Se saca la raíz cuadrada al número de elementos del vector para obtener el número de renglones y columnas de la matriz.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.25. Se detalla la Función SelectVector.

Nombre de Función:	\$cadena = SelectVector(\$id_reactivos)
Parámetros de entrada:	\$id_reactivos: identificador del reactivo.
Parámetros de salida:	<p>\$cadena: cadena de texto que contiene los datos del vector de voltaje o corriente.</p> <ol style="list-style-type: none"> 1. El primer elemento del arreglo indica si es un vector de voltaje o corriente. 2. El segundo elemento del arreglo indica la cadena de texto con la información del vector.
Objetivos:	Realizar una búsqueda en la tabla matrices para buscar la cadena de texto que contiene la información, e indicar si es un vector de voltaje o corriente.

Tabla 4.26. Se detalla la Función Cadena Vector.

Nombre de Función:	\$VectorValor = CadenaVector(\$cadena1)
Parámetros de entrada:	\$cadena1: cadena de texto que indica la información del vector de voltaje/corriente.
Parámetros de salida:	\$VectorValor: vector que contiene las expresiones a evaluar de cada uno de los elementos del vector.
Objetivos:	De la cadena de texto que contiene la información del vector

	voltaje/corriente, se va a subdividir usando como delimitador el carácter “,”, obteniendo un vector \$VectorValor que contendrá los elementos individuales para ser evaluados.
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.27. Se detalla la Función GetNumResistencias.

Nombre de Función:	\$resistencia = GetNumResistencias(\$id_reactivos)
Parámetros de entrada:	de \$id_reactivos: identificador del reactivo.
Parámetros de salida:	\$resistencia: valor entero que contiene el número de resistencias que contiene el reactivo.
Objetivos:	Se realiza una búsqueda en la tabla literales, para realizar un contero del número de resistencias asociadas al reactivo. Se busca las literales de tipo '0'.

Tabla 4.28. Se detalla la Función GetNumFem.

Nombre de la Función:	\$fem = GetNumFem(\$id_reactivos)
Parámetros de entrada:	de \$id_reactivos: identificador del reactivo.
Parámetros de salida:	\$fem: valor entero que nos indica el número de fuentes asociadas al reactivo.
Objetivos:	Se realiza una búsqueda en la tabla literales para contabilizar el número de fuentes asociadas al reactivo. Se busca las literales de tipo '1'.

Tabla 4.29. Se detalla la Función VectorOpciones.

Nombre de Función:	\$opcion = VectorOpciones (\$ renglon, \$columna, \$resistencia, \$MatrizValor, \$TipoMatriz, \$peticion, \$VectorValor, \$fem,\$reporte,\$id_reactivos)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$renglon: número de renglones de la matriz (Admitancia/Resistencia). 2. \$columna: número de columnas de la matriz (Admitancia/Resistencia). 3. \$resistencia: número de resistencias en el reactivo. 4. \$MatrizValor: vector que contiene cada uno de los

	<p>elementos de la matriz (Admitancia/Resistencia) de acuerdo a las ecuaciones de nodo/malla.</p> <ol style="list-style-type: none"> 5. \$TipoMatriz: indica si es una matriz de Admitancia/Resistencia. 6. \$peticion: indica si es un vector de corriente/voltaje. 7. \$VectorValor: contiene los elementos del vector voltaje/corriente. 8. \$fem: número de fuentes asociadas al reactivo. 9. \$reporte: identificador del reporte. 10. \$id_reactivo: identificador del reactivo.
<p>Parámetros de salida:</p>	<p>\$opcion: arreglo de cuatro elementos que contienen las cuatro respuestas del reactivo.</p>
<p>Objetivos:</p>	<ol style="list-style-type: none"> 1. Generar las cuatro respuestas del reactivo y elegir una como correcta. 2. Se elige un número aleatorio entre 0-3, para que esa sea la respuesta correcta. 3. Se asignan valores a las resistencias ValorResistencias(\$resistencia). 4. Se asignan valores a las fuentes ValorPeticones(\$fem). 5. Se crea una cadena de texto que contiene los datos del reactivo: resistencias y fem ShowVariables(\$resistencia,\$VResistencias, \$fem,\$VPeticiones, \$peticion). 6. Una vez que las resistencias tienen valor se crea la matriz de Admitancia/Resistencia MatrizExpresion1(\$renglon, \$columna, \$resistencia, \$MatrizValor, \$TipoMatriz, \$VResistencias,\$show). 7. Una vez que se tiene el valor de las fem se crea el vector Corriente/Voltaje VectorUno(\$columna,\$renglon, \$peticion ,

	<p>$\\$VectorValor$, $\\$VPeticiones$, $\\$show$).</p> <p>8. Teniendo la matriz y el vector se procede a resolver el sistema por el método de Jacobi Jacobi($\\$renglon$, $\\$columna$, $\\$V$, $\\$M$).</p> <p>9. Se guardan los respuestas del sistema en una cadena de texto que a su vez es guardado en un arreglo de 4 elementos (las cuatro respuestas del reactivo)</p> <p>10. En la tabla respuestasdinam se guardan las respuestas generadas por el sistema, la opción correcta y los valores de resistencias y fem asociadas a la respuesta correcta</p> <p>Del paso 3 al 9, se repiten 4 veces para formar las cuatro respuestas.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.30. Se detalla la Función ValorResistencias.

Nombre de la Función:	$\$VResistencias = ValorResistencias(\$resistencia)$
Parámetros de entrada:	$\$resistencia$: valor entero que indica el número de resistencias presentes en el reactivo.
Parámetros de salida:	$\$VResistencias$: vector que indica el valor de cada una de las resistencias presentes en el circuito.
Objetivos:	Se asignan n números aleatorios que representan los valores de las resistencias.

Tabla 4.31. Se detalla la Función ValorPeticiones.

Nombre de la Función:	$\$VPeticiones = ValorPeticiones(\$fem)$
Parámetros de entrada:	$\$fem$: número que indica las fuentes presentes en el reactivo.
Parámetros de salida:	$\$VPeticiones$: vector de n elementos, los cuales dependen del parámetro de entrada, que indican los valores asignados a las fuentes.
Objetivos:	Se asignan n números aleatorios que representan los valores de las fuentes.

Tabla 4.32. Se detalla la Función ShowVariables.

Nombre de la Función:		\$CadenaGeneradora= ShowVariables(\$resistencia, \$VResistencias, \$fem,\$VPeticiones, \$peticion)
Parámetros de entrada:	de	<ol style="list-style-type: none"> 1. \$resistencia: valor entero que indica el número de resistencias. 2. \$VResistencias: vector que contiene los valores de cada una de las resistencias. 3. \$fem: valor entero que indica el número de fuentes presentes en el circuito. 4. \$VPeticiones: vector que contiene los valores de cada una de las fuentes. 5. \$peticion: Indica si es un vector de corriente o voltaje.
Parámetros de salida:		\$CadenaGeneradora: cadena de texto que indica los datos del problema, el número de resistencias, fuentes y los valores asignados a cada uno.
Objetivos:		<p>Formar una cadena de texto que contenga la información de los datos de las variables del problema nombre-valor-unidad de cada uno.</p> <ol style="list-style-type: none"> 1. Primeramente en una cadena de texto que se inicializa como vacía, se le van a concatenar nombre-valor-unidad, de las resistencias que están presentes en el circuito. En un ciclo iterativo for, se van a buscar los valores de las resistencias presentes en el vector \$VResistencias. 2. A esto se va a seguir concatenando los valores de las fuentes, nombre-valor-unidad, dependiendo si es un vector de corriente o voltaje se asignaran las unidades correspondientes. Esta información viene en la variable \$peticion. De igual forma en un ciclo for, se van a buscar los valores de las fuentes presentes en el vector \$VPeticiones.

Tabla 4.33. Se detalla la Función MatrizExpresion1.

Nombre de la Función:	\$M=MatrizExpresion1(\$ renglon, \$columna, \$resistencia, \$MatrizValor, \$TipoMatriz,\$VResistencias)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$renglon: indica el número de renglones de la matriz. 2. \$columna: indica el número de columnas de la matriz. 3. \$resistencia: indica el número de resistencias presentes en el reactivo. 4. \$MatrizValor: vector que contiene los elementos de cada uno de la matriz de Admitancia/Resistencia. 5. \$TipoMatriz: indica si la matriz es de Admitancia/Resistencia. 6. \$VResistencias: vector que contiene los valores de cada una de las resistencias presentes en el circuito.
Parámetros de salida:	\$M: matriz de \$renglonX\$columna, que ya contiene los valores de la matriz de Admitancia/Resistencia.
Objetivos:	<p>Dado el vector \$MatrizValor, , determinar cuáles elementos representan una suma algebraica, y en esa suma algebraica determinar cuáles son las resistencias que intervienen (R1, R2, etc). Una vez encontradas las resistencias buscar en el vector \$VResistencias, el valor a esa resistencia. Para así determinar el valor numérico y obtener la matriz de Admitancia/Resistencia. En caso de no encontrarse ninguna resistencia el valor por default será 0.</p> <ol style="list-style-type: none"> 1. Dentro de un ciclo iterativo for, analizar cada uno de los elementos del vector \$MatrizValor, para determinar si existen resistencias. 2. El elemento del vector es una cadena de texto, usando del delimitador ' ', se separarán las resistencias existentes para crear subcadenas, las cuales se guardaran en un arreglo, en caso que se encuentren. 3. Se verifica cuantas resistencias tiene el elemento del vector contando el número de elementos que tiene el

	<p>arreglo creado.</p> <p>4. En otro ciclo iterativo for en base al número de elementos encontrados, se va a determinar el signo del elemento, se busca el carácter '-', en caso de encontrarlo se pondrá signo 'menos', en otro caso 'mas'. A continuación se va a determinar cuál es el número de resistencia, haciendo una búsqueda secuencial recorriendo desde R1 hasta Rn, una vez encontrada la resistencia se procede a buscar el valor en el vector \$VResistencias para hacer la suma o resta según sea el caso. Se verifica si la matriz es de Admitancia ó Resistencia, para hacer la suma algebraica del inverso o del valor de la resistencia. En caso de buscar R1, se verifica si el siguiente carácter es otro número para evitar errores y no confundirlo con R11, R12, etc.</p> <p>5. Una vez realizada la sumatoria, se asigna el valor a la matriz, en caso de no encontrarse ninguna resistencia se asigna 0.</p>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.34. Se detalla la Función VectorUno.

Nombre de Función:	\$V=VectorUno(\$renglon, \$peticion , \$VectorValor,\$VPeticiones)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$renglon: indica el número de renglones del vector. 2. \$peticion: indica si es un vector de corriente o voltaje. 3. \$VectorValor: vector que indica las expresiones del vector de corriente o voltaje según las ecuaciones de nodo o malla. 4. \$VPeticiones: vector que indica los valores de las fuentes presentes en el circuito.
Parámetros de salida:	\$V: vector con los valores del vector de voltaje o corriente

Objetivos:	<p>Obtener los valores del vector corriente/voltaje, en base a las expresiones algebraicas y los valores de las fuentes, localizadas en el vector \$VectorValor, \$VPeticiones.</p> <ol style="list-style-type: none"> 1. Determinar el tipo de vector corriente/voltaje para determinar si se buscan IS ó VS, en las expresiones del \$VectorValor. 2. Dentro de un ciclo for, analizar cada uno de los elementos del vector \$VectorValor para determinar si se encuentran fuentes (IS ó VS). 3. Se realiza una subdivisión del elemento del vector usando como delimitador ' ', teniendo como resultado un vector que contiene n elementos dependiendo del número de fuentes, con su respectivo signo. 4. Dentro de otro ciclo for, que va a depender del número de elementos que se encontraron, se va hacer una búsqueda secuencial de las fuentes existentes, por ejemplo (VS1, VS2, etc.), se va a buscar el carácter '-', para determinar el signo, 'negativo' , 'positivo' y así realizar una suma o resta del valor asociado a dicha fuente.
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.35. Se detalla la Función Jacobi.

Nombre de la Función:	\$Xfinal=Jacobi(\$renglon,\$columna,\$V,\$M)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$renglon: número de renglones de la matriz y del vector. 2. \$columna: número de columnas de la matriz. 3. \$V: vector de corriente/voltaje con los valores. 4. \$M: matriz de Admitancia/Resistencia con los valores.
Parámetros de salida:	\$Xfinal: arreglo que contiene los valores que se obtuvieron al resolver el sistema de ecuaciones por el método de Jacobi.
Objetivo:	Resolver el sistema de ecuaciones teniendo la matriz de Admitancia/Resistencia y el vector de Corriente/Voltaje.

	<p>En el anexo C, se detalla el método de Jacobi.</p> <p>Se concatenan las respuestas para formar una cadena de texto, que indica, variable-valor-unidades. Que se guarda en un arreglo que contiene las cuatro soluciones del reactivo.</p>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 4.36. Se detalla la Función RespuestaDinamica.

Nombre de Función:	RespuestaDinamica(\$correcta,\$opcion, \$CadenaGeneradora,\$reactivos, \$reporte)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$correcta: indica el número de la opción correcta. 2. \$opcion: arreglo que contiene las cuatro respuestas del reactivo. 3. \$CadenaGeneradora: cadena de texto que contiene los nombres de los datos-valor-unidades del reactivo. 4. \$reactivos: identificador del reactivo. 5. \$reporte: identificador del reporte.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivo:	Guardar en la tabla respuestasdinam las cuatro opciones de respuestas, la opción correcta, la cadena de texto que contiene las variables con sus respectivos valores, buscando el registro por el identificador de reporte.

Tabla 4.37. Se detalla la Función DespliegueRadio.

Nombre de la Función:	DespliegueRadio (\$radio,\$opcion)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$radio: nombre del control radio button (botones radiales) (respuesta1, respuesta2, etc.). 2. \$opcion: arreglo con la cadena de texto que contiene las respuestas del reactivo.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivo:	Desplegar cuatro controles radio buttons (botones radiales) que contienen las respuestas al reactivo.

Tabla 4.38. Se detalla la Función PreguntaAvanzado.

Nombre de la Función:	PreguntaAvanzado(\$idreporte)
Parámetros de entrada:	\$idreporte: identificador de reporte.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivo:	<p>Una vez desplegada la página de ExamAvanzado.php, si el alumno recarga la página se desplegarán los reactivos y sus respuestas, los cuales ya se guardaron la primera vez de visualizar la página.</p> <p>Se realiza una búsqueda en las tablas reactivos y respuestasdinam, para obtener los datos de los reactivos y las respuestas de cada reactivo.</p>

Tabla 4.39. Se detalla la Función AlumnoRespuestas.

Nombre de la Función:	AlumnoRespuestas(\$respuesta, \$idreporte)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$respuesta: arreglo que contiene las respuestas enviadas del examen avanzado. 2. \$idreporte: identificador del reporte, obtenido de la variable de sesión 'reporte'.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivo:	<p>Buscar en la tabla respuestasdinam los reactivos y la opción correcta de acuerdo al identificador de reporte para compararlas con las respuestas elegidas por el alumno, y determinar la calificación individual de cada reactivo así como la calificación total.</p> <p>Se realizan actualizaciones en la tabla respuestasdinam en los campos opcion_elegida y en la tabla reportes con el campo Calif_Total.</p>

Tabla 4.40. Se detalla la Función ResultadoReporte.

Nombre de la Función:	ResutadoReporte(\$idreporte)
Parámetros de entrada:	\$idreporte: identificador de reporte.
Parámetros de salida:	No se cuenta con parámetro de salida.
Objetivo:	Se realiza una búsqueda en las tablas reactivos y respuestasdinam para obtener la información de los campos referentes a la información del reactivo y las respuestas respectivas.

Tabla 4.41. Se detalla la Función SeleccionReactivos.

Nombre de la Función:	SeleccionReactivos()
Parámetros de Entrada:	No contiene parámetros de entrada.
Parámetros de Salida:	Arreglo con los identificadores de los reactivos seleccionados.
Objetivo:	<p>Buscar reactivos, para seleccionar un grupo que contenga tanto reactivos de alto puntaje como de bajo puntaje.</p> <ol style="list-style-type: none"> 1. Se selecciona un reactivo de alto puntaje (5, 4 puntos). 2. Se seleccionar un grupo de reactivos de bajo puntaje ((2,2,1), (2,2,2),(2,1,1,1)), dependiendo del puntaje del reactivo de alto puntaje para que se complete la calificación total. 3. Los reactivos seleccionados son guardados en un arreglo que sirve como parámetro de salida.

Tabla 4.42. Se detalla la Función mezcla.

Nombre de la Función:	mezcla(\$nombre,\$numCuenta)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$nombre: Nombre del alumno. 2. \$numCuenta: Número de cuenta del alumno.
Parámetros de salida:	Cadena de texto que contiene los caracteres del nombre y número de cuenta en forma desordenada.
Objetivo:	<p>Obtener una cadena de texto única formada por el nombre del alumno y el número de cuenta.</p> <p>El número de cuenta se va a insertar en la cadena de texto que contiene el nombre, cada determinado número de espacios, para posteriormente desordenar esa cadena. La cual va a ser el parámetro de salida.</p> <p>Ver diagrama de flujo en ANEXO B.</p>

Tabla 4.43. Se detalla la Función TextoANúmero.

Nombre de la Función:	textoANúmero(\$cadena, \$factor)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$cadena: Cadena de texto que contiene el número de cuenta y el nombre del alumno, en forma desordenada. 2. \$factor: factor multiplicativo a cada uno de los caracteres de la cadena de texto.
Parámetros de salida:	\$arr: arreglo de valores numéricos que contienen valores aleatorios que son utilizados por los reactivos dinámicos.
Objetivo:	<p>A partir de una cadena de texto, obtener un arreglo de valores enteros.</p> <ol style="list-style-type: none"> 1. La cadena de texto se convierte en un arreglo de caracteres. 2. Se recorre toda el vector de caracteres, cada elemento se convierte a su respectivo valor ASCII el valor obtenido es multiplicado por un factor obteniendo como resultado un vector con valores numéricos. 3. Ver diagrama de flujo en ANEXO B.

Para el despliegue de Reactivos de Fasores en la siguiente tabla se despliegan las funciones para supervisar este tipo de reactivos, en la pantalla de seleccionar Reactivo Dinámico se selecciona Reactivo con Fasores, posteriormente aparecen los Reactivos con Fasores, al igual que los Reactivos Dinámicos de números reales, se selecciona uno para verificar el correcto desempeño; verificando que se haya realizado correctamente la solución del sistema de ecuaciones y que se haya insertado la información correcta concerniente a la matriz y vector de la ecuación a solucionar, igualmente se usa el método iterativo de Jacobi, adaptado para números complejos (Tabla 4.44), dicha función hace uso de funciones de suma (Tabla 4.45), resta (Tabla 4.46), multiplicación (Tabla 4.47) y división (Tabla 4.48) de números complejos:

	Objetivos:	Funciones o Instrucciones:
1	Revisión de Reactivos Dinámicos.	<p>En la página de inicio_profesor.php , seguir el siguiente vínculo</p> <p>Consultar Reactivo->Revisión Reactivo Dinámico.</p> <p>Seleccionar reactivos con fasores, para revisar la correcta resolución del sistema de ecuaciones con fasores.</p>
2	Selección de Reactivos con Fasores.	<p>Una vez seleccionados este tipo de reactivos, se elige el reactivo en específico a revisar.</p> <p>Se presiona el botón seleccionar registro.</p> <p>Se siguen en su mayoría los mismos pasos que para desplegar un reactivo dinámico, como se explicó anteriormente.</p> <p>Se hacen modificaciones al algoritmo de Jacobi para que resuelva el sistema de ecuaciones con números complejos.</p> <p>Se utiliza la función <code>JacobiC(\$indicei, \$indicej,\$Vector,\$Matriz)</code>, para resolver el sistema de ecuaciones.</p>

Tabla 4.44. Función que resuelve un sistema de ecuaciones por el método de Jacobi utilizando números complejos:

Nombre de la Función:	JacobiC(\$índicei, \$índicej,\$Vector,\$Matriz)
Parámetros de Entrada:	<ol style="list-style-type: none"> 1. \$índicei: número de renglones de la matriz. 2. \$índicej: número de columnas de la matriz. 3. \$Vector: arreglo con los valores del vector independiente. 4. \$Matriz: arreglo bidimensional con los valores de la matriz.
Parámetros de salida:	Arreglo con la solución del sistema.
Objetivos:	<p>Resolver el sistema de ecuaciones de números complejos: Se realiza una asignación de valores a un vector de solución inicial. Se elige el número de iteraciones. Se realizan dos ciclos iterativos para recorrer renglón y columnas del renglón. Al realizar el recorrido por las columnas de un renglón se realiza la multiplicación del elemento del vector inicial de índice j con el elemento de la matriz j que le corresponde. No se realiza la multiplicación cuando el índice i del renglón es igual al índice j de la columna. Dichas multiplicaciones de elementos se van acumulando en una sumatoria. Al tener la sumatoria de las multiplicaciones se realiza una resta con el índice del vector independiente correspondiente. Posteriormente se realiza una división con el elemento de la matriz $A_{ij, i=j}$. Se realiza esto con todos los renglones y se guardan los nuevos valores iniciales.</p>

Tabla 4.45. Función que realiza una suma de complejos.

Nombre de la Función:	suma1(\$primero,\$segundo)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$primero: primer número complejo a ser sumado. 2. \$segundo: segundo número complejo a ser sumado.
Parámetros de salida:	Regresa la suma de los dos números complejos.
Objetivos:	Suma las partes reales y complejas de los parámetros de entrada.

Tabla 4.46. Función que realiza una resta de números complejos.

Nombre de la Función:	resta1(\$primero, \$segundo)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$primero: minuendo de la resta. 2. \$segundo: sustraendo de la resta.
Parámetros de salida:	Regresa la diferencia de los dos números complejos.
Objetivos:	Resta las partes reales y complejas de los parámetros de entrada.

Tabla 4.47. Función que realiza la multiplicación de dos números complejos.

Nombre de la Función:	function mult1(\$primero,\$segundo)
Parámetros de entrada:	<ol style="list-style-type: none"> 1. \$primero: primer factor a multiplicar. 2. \$segundo: segundo factor a multiplicar.
Parámetros de salida:	Regresa el producto de los dos factores.
Objetivos:	Se realiza la suma algebraica de la parte real del primer factor con la parte real e imaginaria del segundo factor, y la parte imaginaria del primer factor con la parte real e imaginaria del segundo factor.

Tabla 4.48. Función que realiza la división de dos números complejos.

Nombre de la Función:	div1(\$primero,\$segundo)
Parámetros de entrada:	1. \$primero: dividendo. 2. \$segundo: divisor.
Parámetros de salida:	Resultado de la división.
Objetivos:	Se multiplica al divisor por su conjugado obteniéndose un número real. Se multiplica al dividendo por el conjugado del divisor. Se agrupan términos reales y complejos, y se realiza la división teniendo como divisor el número real obtenido en la primera multiplicación.

5. PROCEDIMIENTOS DE METODOLOGÍA DE PRUEBAS



Existen una variedad de pruebas que pueden ser aplicadas a las diferentes actividades del proceso de desarrollo, los cuales demandan de tiempo y presupuesto adicionales y pueden llegar a representar un alto porcentaje del costo total.

Existen diferentes tipos de pruebas empleadas durante las diferentes actividades del proceso de desarrollo, las cuales requieren de tiempo y presupuesto adicionales, que pueden llegar a significar un alto porcentaje del costo total. Por tal motivo, el modelo de pruebas debe ser planificado con anticipación y de manera integral junto con el desarrollo, ya que no se puede lograr software de alta calidad si no es mediante pruebas finales y depuraciones.

De manera general existen conceptos conocidos informalmente como “*bugs*”, los cuales son errores presentes en el sistema, los cuales se hacen evidentes cuando

se desarrollan pruebas al sistema, a continuación se mencionan algunas definiciones que particularizan estos errores dentro de un programa :

- Una falla (*failure*) sucede cuando un programa no se comporta de manera apropiada. La falla es una propiedad (estadística) de un sistema en ejecución.
- Una falta (*fault*) se lleva a cabo en el código del programa; la existencia de una falta en el programa puede ocasionar una falla (*failure*) en el sistema.
- Un error es una tarea humana que provoca que un software contenga una falta; un error puede significar la existencia de una falta en el programa, al escribir código que contenga faltas involuntarias hace que el sistema falle.

Un aspecto importante relacionado con los conceptos anteriores es que no se puede garantizar ni probar que un sistema no falle, solamente se puede demostrar que contiene faltas.

Dada la dificultad de probarlo y de encontrar faltas, el encargado de detectarlas en el código es una persona distinta al desarrollador del sistema, Probadores o *Testers*. Esto también significa un costo adicional en el desarrollo de éste, por lo cual a veces sólo se prueban las partes principales del sistema.

Los dos tipos principales de pruebas se dividen de manera general en pruebas de verificación y validación. En el primer caso se revisa si el resultado corresponde a la especificación del sistema, es decir, si se está construyendo el sistema de manera correcta, algo que por sí sólo no garantiza la satisfacción de los clientes. En el segundo caso, se revisa si el resultado es realmente lo que el cliente quiere, en otras palabras, si se está construyendo el sistema correcto, de manera que tanto la especificación como el resultado lo sean.

Las técnicas utilizadas para realizar las pruebas son muy variadas, pero se pueden destacar las siguientes:

- **Prueba de regresión:** Tiene el propósito de verificar el sistema después de haberle introducido cambios; por ejemplo, al corregir una falta, de manera que se mantenga la funcionalidad especificada originalmente.
- **Prueba de operación:** Su objetivo es verificar el sistema de operación por un largo período bajo condiciones normales de uso. Este tipo de prueba mide la confiabilidad (*reliability*) del sistema.
- **Prueba de escala completa:** Trata de verificar el sistema en su carga máxima, mediante la asignación de los parámetros a su valor límite y la interconexión del sistema con un máximo de equipos y usuarios simultáneos. Su máxima expresión es la prueba de estrés (*stressing*), que significa que se prueba el sistema en los límites extremos para determinar su nivel de tolerancia y verificar ocurre algún tipo de falla.
- **Prueba de rendimiento (*performance*) o de capacidad:** Tiene como propósito medir la capacidad de procesamiento del sistema bajo diferentes cargas, incluyendo el espacio de almacenamiento y utilización de la unidad de procesamiento y así valores medidos se comparan con los valores requeridos.
- **Prueba negativa:** Tiene como propósito medir el estrés del sistema en situaciones inesperadas, como Casos de Uso que normalmente no serían utilizados de manera simultánea; aquí el sistema se usa intencional y sistemáticamente de forma incorrecta, este maltrato debe ser planeado de forma cuidadosa para probar aspectos especialmente críticos.
- **Prueba basada en requisitos o prueba de Casos de Uso:** Lleva a cabo pruebas basadas directamente en la especificación de requisitos; pueden utilizarse los mismos Casos de Uso originales como casos de prueba, así

mismo también pueden ser utilizadas para verificar las especificaciones de rendimiento o de escala completa, se persigue verificar que el sistema final cumple con las especificaciones funcionales descritas por los Casos de Uso originales.

Existen tres niveles principales para aplicar las diversas técnicas de pruebas:

- **Prueba de unidad:** Mediante esta prueba sólo una unidad es probada como tal; como una clase, un paquete de servicio o un subsistema.
- **Prueba de integración:** En ella se verifica que las unidades trabajen juntas correctamente, ambas pueden ser realizadas mediante Casos de Uso de pruebas, los cuales pueden ser aplicados a clases, paquetes de servicio, subsistemas y el sistema completo.
- **Prueba de sistema:** Verifica el sistema completo o su aplicación como tal; se toma el punto de vista del usuario final y los Casos de Uso de pruebas ejecutan acciones típicas del usuario.

Este proyecto se desarrollo para cubrir las necesidades de la Prueba de Interrogación, que ha su vez incluye la Prueba de Unidad; así mismo se optó por el Caso de Uso Realizar Examen, por ser el tema central del desarrollo del proyecto.

5.1 PRUEBAS DE INTEGRACIÓN

Para realizar la prueba de integración se realiza se utiliza la prueba de Casos de Uso, en específico el Caso de Uso Realizar Examen con Subflujo 1 (Examen Básico). Donde se va a verificar que se cumplen las siguientes acciones y especificaciones:

- Funcionamiento de los vínculos de las páginas index.php, loginalumno.php, inicio_alumno.php, iniciar_exam.php, Pregunta1_SIEL.php, Reporte.php.
- Verificar que la información que se despliega en las páginas al realizar una búsqueda en la base de datos, en verdad concuerda con la existente en la tabla en cuestión.
- Revisar que se cree el registro en cuestión de reporte en la base de datos al momento de inicializar el examen.
- Comprobar que se guarden las opciones elegidas por el alumno, en cada uno de los reactivos; así como su respectivo puntaje en la tabla respuestasfijas.
- Confirmar que en la tabla reportes, se guardó la calificación del reporte de examen.

En la página de inicio el alumno debe elegir el vínculo de alumno para ingresar a su respectivo módulo y llevar a cabo el examen.

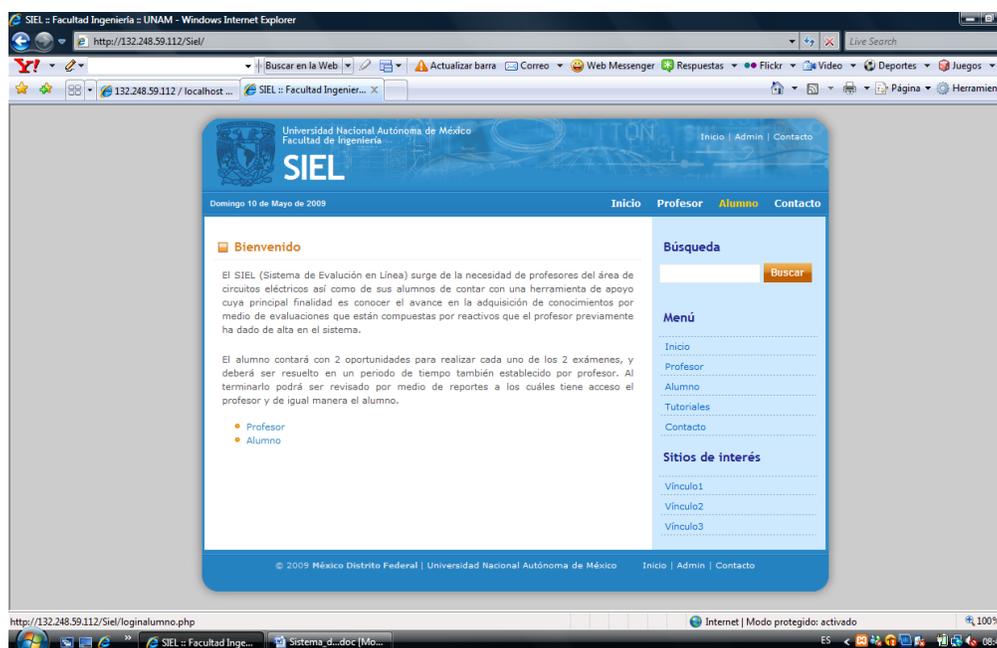


Figura 5.1. Página de Inicio del Sistema SIEL.

Se despliega la página `loginalumno.php` donde al alumno introduce su usuario y `password`; para que el sistema verifique en la base de datos, los datos introducidos en la tabla `alumnos`.

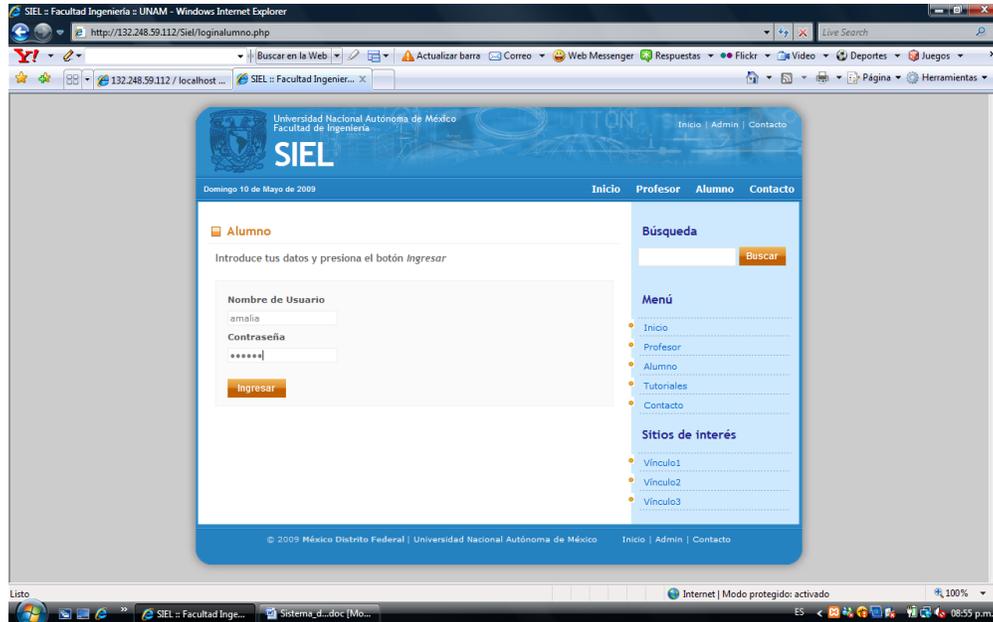


Figura 5.2. El alumno teclea su usuario y password para ser autenticado por el sistema.

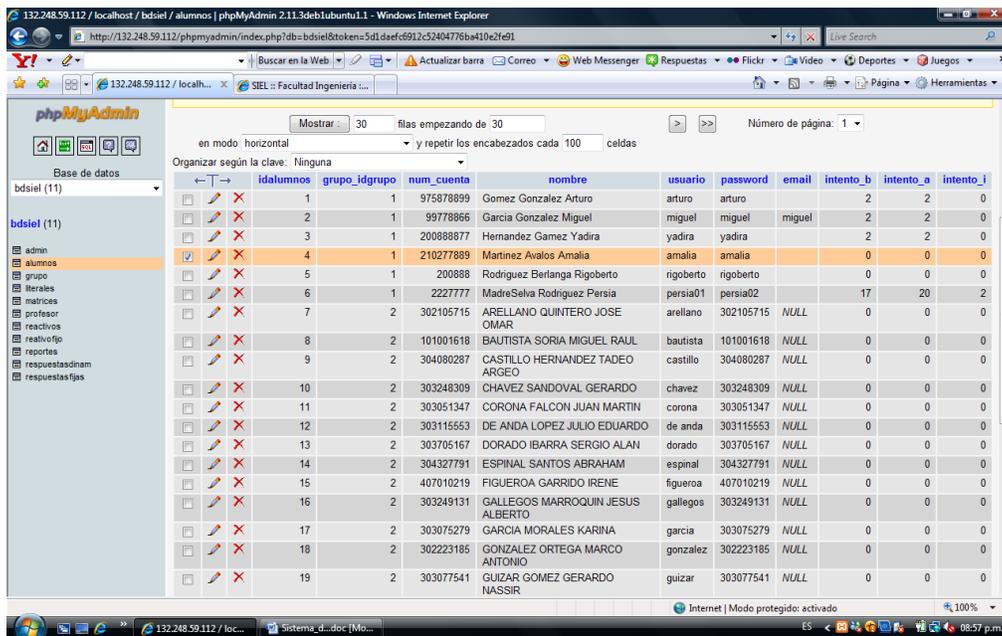


Figura 5.3. El sistema busca en la tabla alumnos, y verifica que existen y concuerdan los datos ingresados por el alumno.

Una vez autenticado el alumno se despliega la página inicio_alumno.php donde el alumno puede elegir una acción a realizar.

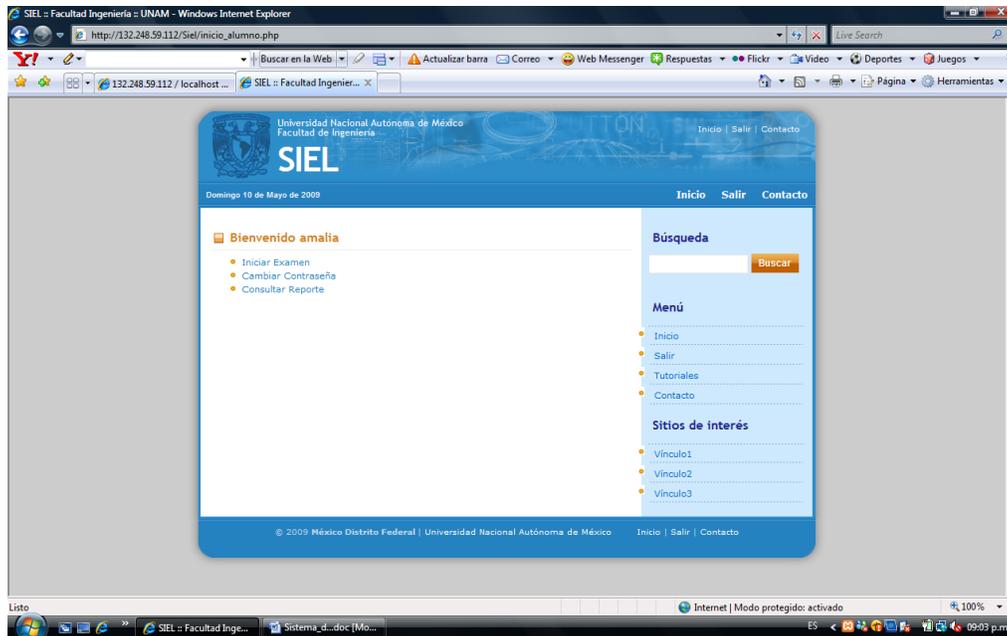


Figura 5.4. Al ser autenticado el alumno, se despliega la página inicio_alumno.php.

Al seleccionar el vínculo Iniciar Examen, se despliega la pantalla iniciar_exam.php; Se despliega información relacionada con el número de intentos que tiene el alumno en los diferentes tipos de exámenes, la cual se obtiene de la tabla alumnos de la base de datos para ser desplegada en la página.

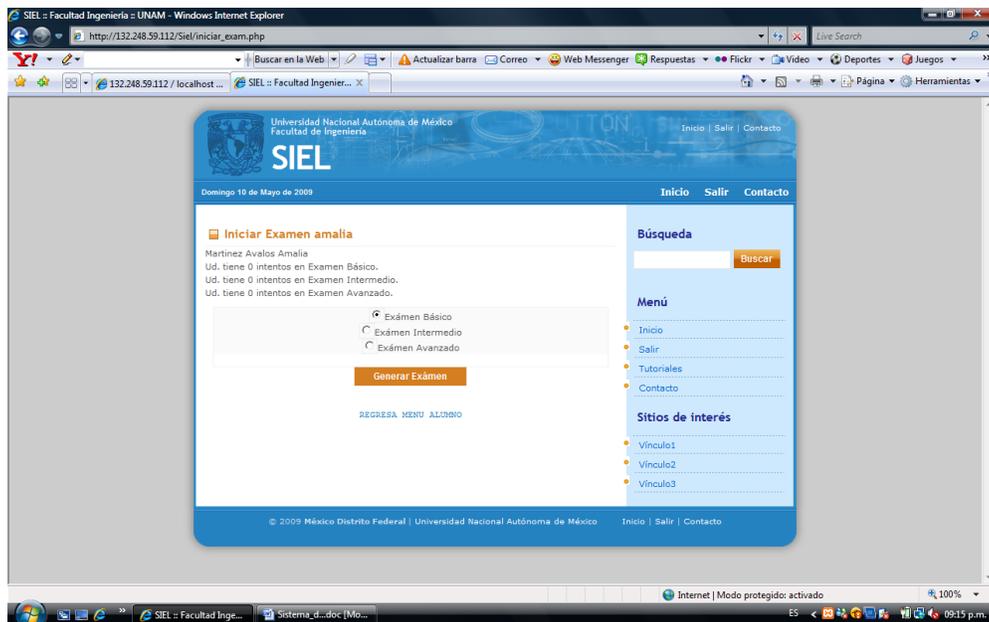


Figura 5.5. Se despliegan los intentos que tiene el alumno en cada uno de los exámenes, el alumno puede elegir cualquiera de los tres exámenes.

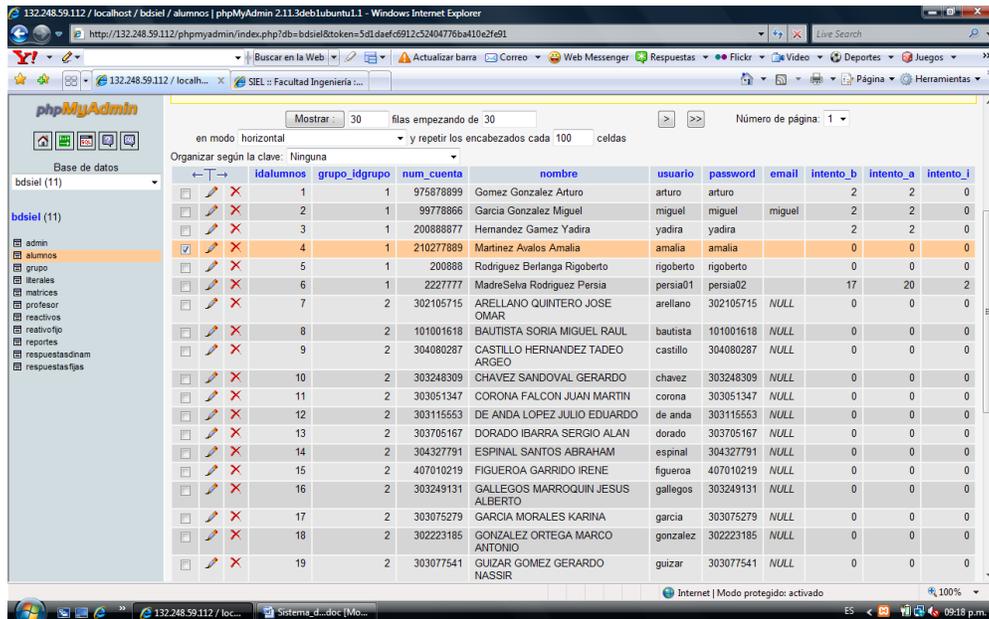


Figura 5.6. El sistema hace una búsqueda en la base de datos; tabla alumnos, para buscar la información concerniente al número de intentos del alumno en cada uno de los exámenes. Se selecciona el Examen Básico, se presiona el botón Generar Examen, y se despliega la página Pregunta1_SIEL.php, con las preguntas seleccionadas de la tabla de reactivos, es cuál es el banco de preguntas con que se cuenta.

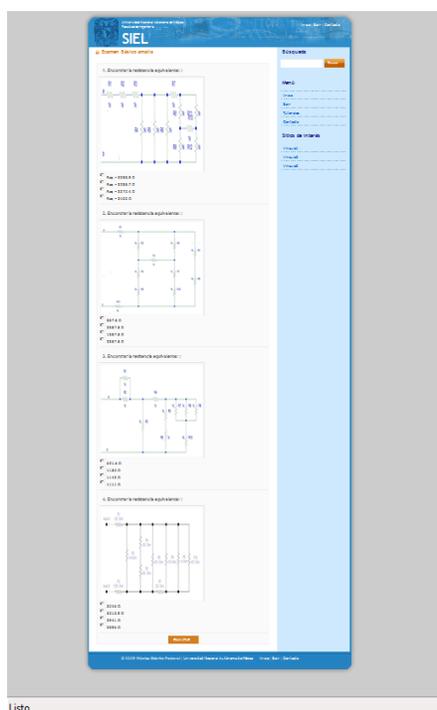


Figura 5.7. El sistema elige 4 reactivos del banco de preguntas y las despliega con sus respuestas para que el alumno pueda elegir la correcta.

Se verifica en la tabla reportes que se creó el registro concerniente al examen que se está realizando, verificando que concuerdan los datos de identificador de alumno, fecha y tipo de examen (Básico). Vemos que la calificación permanece como NULL¹, porque el valor predeterminado queda libre para un registro nuevo; al finalizar el examen, se guarda la calificación del mismo. Para este caso vemos que se creó el reporte con número de identificador 99.

¹ En SQL, NULL no es un valor. Es un estado que el valor es desconocido o no existente.

The screenshot shows the phpMyAdmin interface for a database named 'bdsiel'. The table 'reportes' is selected, and its data is displayed in a table. The columns are: idreportes, alumnos_idalumnos, calificacion, tiempo_resolucion, fecha, and tipo. The last row, with idreportes 99, is highlighted in orange, indicating it is a new record. The interface also shows a sidebar with database navigation options and a bottom status bar.

idreportes	alumnos_idalumnos	calificacion	tiempo_resolucion	fecha	tipo
78	6	7	NULL	2009-04-20 15:23:15	Básico
79	6	0	NULL	2009-04-20 18:45:22	Avanzado
80	6	0	NULL	2009-04-20 18:49:34	Avanzado
81	6	NULL	NULL	2009-04-20 18:53:41	Avanzado
82	6	0	NULL	2009-04-20 18:58:02	Avanzado
83	6	3	NULL	2009-04-20 19:13:12	Avanzado
84	6	0	NULL	2009-04-20 19:26:47	Avanzado
85	6	6.66667	NULL	2009-04-20 19:28:03	Avanzado
86	6	6.66667	NULL	2009-04-20 19:29:18	Avanzado
87	6	6.66667	NULL	2009-04-20 19:32:19	Avanzado
88	6	0	NULL	2009-04-20 19:34:02	Básico
89	6	NULL	NULL	2009-04-20 19:46:35	Básico
90	6	0	NULL	2009-04-21 08:28:47	Básico
91	6	0	NULL	2009-04-21 10:19:25	Básico
92	6	2.5	NULL	2009-04-21 19:27:49	Básico
93	6	5	NULL	2009-04-21 19:47:30	Básico
94	6	0	NULL	2009-04-21 20:25:41	Intermedio
95	6	6.66667	NULL	2009-04-21 21:17:33	Avanzado
96	6	2.5	NULL	2009-04-23 17:22:06	Intermedio
97	6	5	NULL	2009-04-23 17:33:21	Básico
98	6	0	NULL	2009-04-23 19:09:33	Básico
99	4	NULL	NULL	2009-05-10 21:20:26	Básico

Figura 5.8. Se crea un nuevo registro en la tabla reportes con los datos del identificador del alumno, fecha y tipo de examen (Básico/Avanzado).

Verificamos también que en la tabla respuestasfijas se guardaron los identificadores de las preguntas seleccionadas del banco de preguntas que corresponden al identificador de reporte 99.

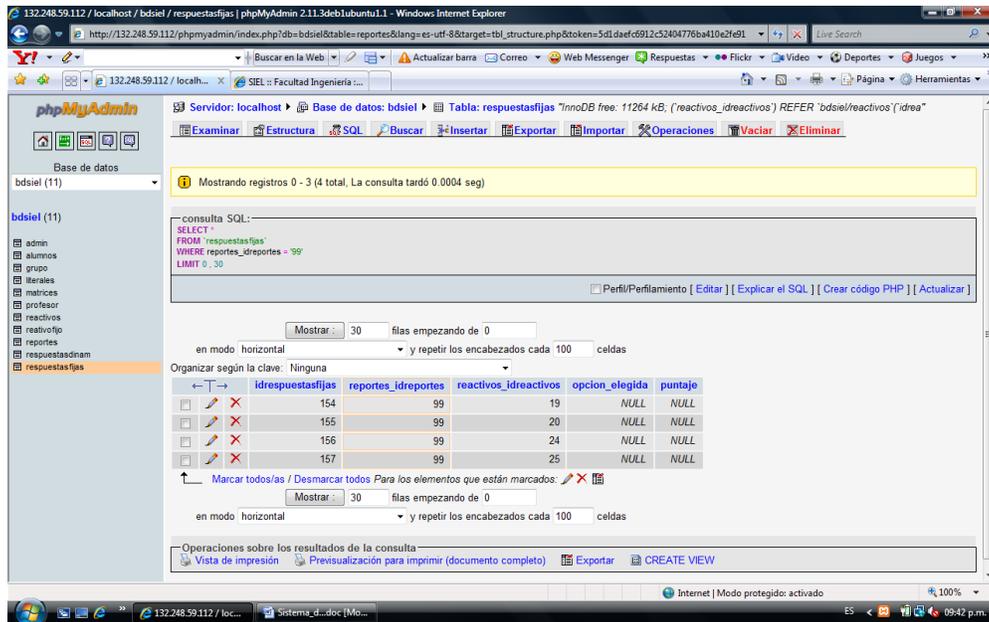


Figura 5.9. En la tabla respuestasfijas se guardan el identificador de reporte junto con los identificadores de las preguntas seleccionadas para el examen.

Una vez que el alumno seleccionó las respuestas, procede a elegir el botón “FINALIZAR”, el cual da por terminado el examen y se procede a realizar el reporte del examen, desplegando la página Reporte.php.

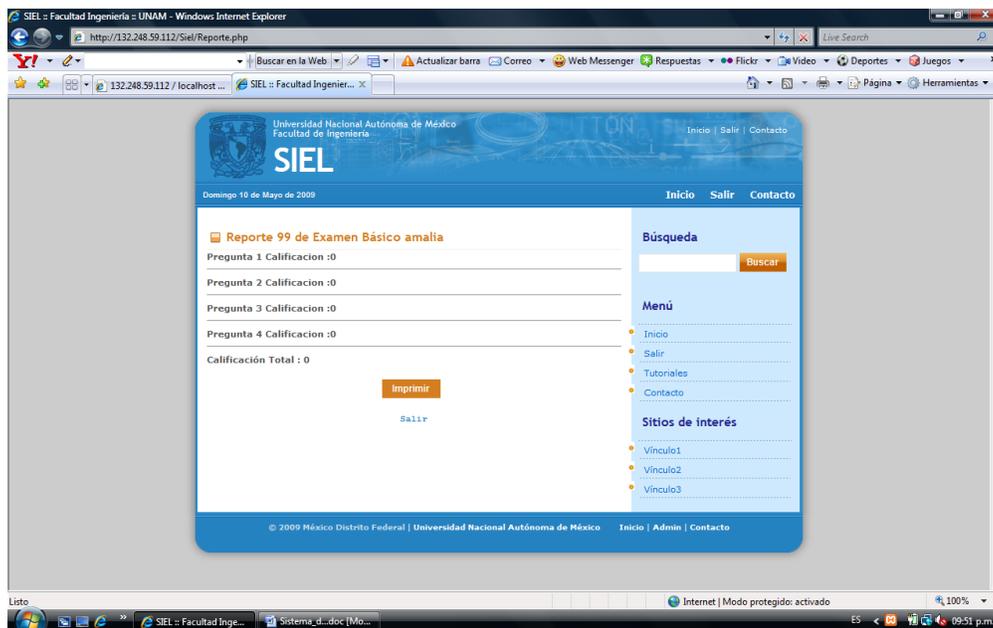


Figura 5.10. Al presionar el botón FINALIZAR del examen, se despliega la página Reporte.php, la cual muestra el puntaje por reactivo así como la calificación total del examen.

Se verifica la tabla respuestasfijas dónde vemos que se guardó la opción que el alumno selecciono así como su calificación de cada reactivo.

Mostrando registros 0 - 3 (4 total. La consulta tardó 0.0004 seg)

```
consulta SQL:
SELECT *
FROM `respuestasfijas`
WHERE reportes_idreportes = 99
LIMIT 0, 30
```

idrespuestasfijas	reportes_idreportes	reactivos_idreactivos	opcion_elegida	puntaje
154	99	19	3	0
155	99	20	3	0
156	99	24	4	0
157	99	25	4	0

Figura 5.11. Se verifica que en la tabla respuestasfijas se guardaron las opciones elegidas por el alumno, así como el puntaje correspondiente.

Vemos que en la tabla reportes se guardo la calificación del examen con identificador de reporte 99.

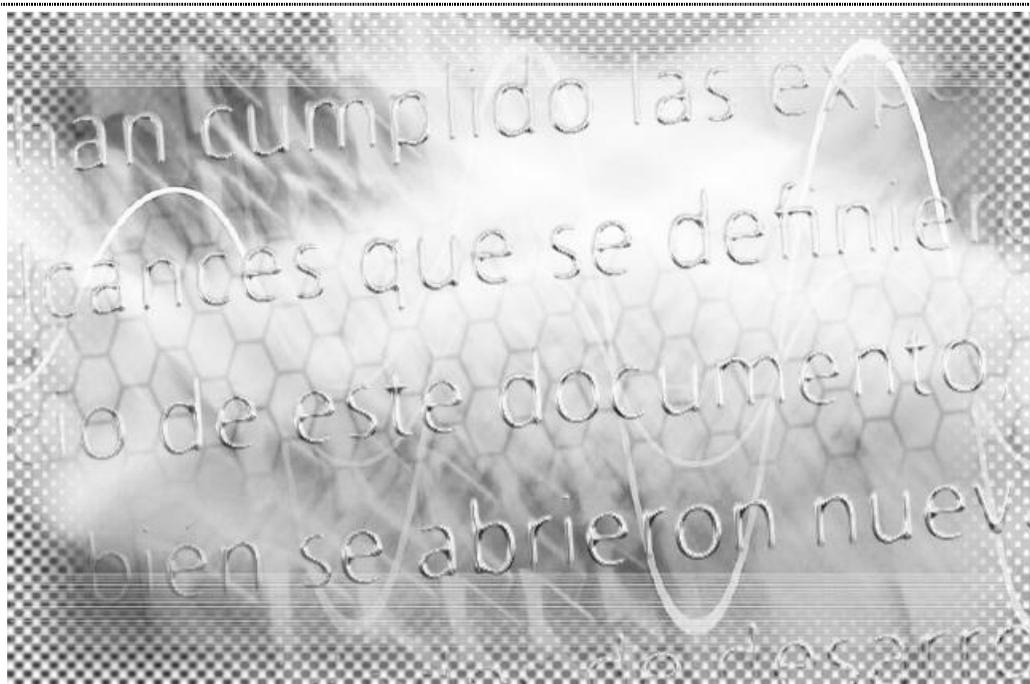
Organizar según la clave: ninguna

idreportes	alumnos_idalumnos	calificacion	tiempo_resolucion	fecha	tipo
78	6	7	NULL	2009-04-20 15:23:15	Básico
79	6	0	NULL	2009-04-20 18:45:22	Avanzado
80	6	0	NULL	2009-04-20 18:49:34	Avanzado
81	6	NULL	NULL	2009-04-20 18:53:41	Avanzado
82	6	0	NULL	2009-04-20 18:58:02	Avanzado
83	6	3	NULL	2009-04-20 19:13:12	Avanzado
84	6	0	NULL	2009-04-20 19:26:47	Avanzado
85	6	6.66667	NULL	2009-04-20 19:28:03	Avanzado
86	6	6.66667	NULL	2009-04-20 19:29:18	Avanzado
87	6	6.66667	NULL	2009-04-20 19:32:19	Avanzado
88	6	0	NULL	2009-04-20 19:34:02	Básico
89	6	NULL	NULL	2009-04-20 19:46:35	Básico
90	6	0	NULL	2009-04-21 08:28:47	Básico
91	6	0	NULL	2009-04-21 10:19:25	Básico
92	6	2.5	NULL	2009-04-21 19:27:49	Básico
93	6	5	NULL	2009-04-21 19:47:30	Básico
94	6	0	NULL	2009-04-21 20:25:41	Intermedio
95	6	6.66667	NULL	2009-04-21 21:17:33	Avanzado
96	6	2.5	NULL	2009-04-23 17:22:06	Intermedio
97	6	5	NULL	2009-04-23 17:33:21	Básico
98	6	0	NULL	2009-04-23 19:09:33	Básico
99	4	0	NULL	2009-05-10 21:20:26	Básico

Mostrar: 30 filas empezando de 0 Número de página: 2

Figura 5.12. Se verifica que en la tabla reportes se guarda la calificación total del examen de acuerdo al identificador del alumno que hizo el examen.

6. CONCLUSIONES



El SIEL representa una herramienta para que el alumno pueda valorar su grado de aprendizaje en el área de Circuitos Eléctricos, así mismo para que el docente cuente con un instrumento más, para evaluar el grado de aprovechamiento individual y grupal.

Es un sistema piloto, que permite evaluar en línea el avance de los alumnos en el área de Circuitos Eléctricos, mediante la aplicación de exámenes con diferente grado de dificultad, abarcando los temas incluidos en la asignatura.

Se ha cumplido el objetivo de elaborar esta herramienta que tienen la dinámica de generar exámenes aleatorios; con preguntas y respuestas únicas para cada alumno, no importando que el enunciado del problema sea el mismo y esto se convierte en un elemento idóneo que evita que los alumnos tengan el mismo problema y que sea resuelto en forma grupal, circulando para todos únicamente

las respuestas, ya que debido a esta característica los resultados variarían entre cada uno de ellos y conforme a la función mezcla que es la encargada de dar valores únicos, aún cuando sea el mismo alumno en dos sesiones diferentes le generara dos exámenes y en el peor de los escenarios que le asignara el mismo reactivo los valores de cada una de las variables a considerar serán diferentes.

Los reactivos que forman el banco de preguntas, están formados por:

Preguntas fijas: Al momento de dar de alta este tipo de reactivo se guarda en la base de datos el enunciado de la pregunta, el diagrama del circuito (sí existe) y las respuestas relacionadas a este reactivo. Para la construcción de las respuestas dentro de la base de datos, la primera opción siempre será la correcta, las soluciones erróneas se construyen tomando en cuenta los errores más comunes que tienen los alumnos al realizar un examen, tales como cambiar el sentido de las fuentes involucradas, cambiar el signo de una o más variables en una suma algebraica de una variable, etc. y se almacenan en forma consecutiva después de la opción correcta de la base de datos

Preguntas dinámicas: Como principal aportación del sistema se concibió la idea de este tipo de reactivos; además de contar con los campos comunes de los reactivos fijos del enunciado de problema y nombre del diagrama del reactivo, cuenta con información concerniente a las ecuaciones de malla o nodo del reactivo; las ecuaciones de la matriz de malla o nodo, se van a ingresar por medio de una cadena de texto con el siguiente formato: `matriz(fila(..., ...), fila(..., ...))` y para el vector de corriente voltaje el formato es el siguiente: `vector(..., ...)`. De tal forma que al realizar un examen con este tipo de reactivos, el sistema lo primero que hace, es obtener la información concerniente a las ecuaciones de malla tanto de la matriz, como del vector; por medio de las cadenas de texto que se tienen en la base de datos, después interpreta la información que contiene esta cadena para obtener las ecuaciones de cada elemento de la matriz y vector del problema.

Posteriormente se buscan dentro de cada ecuación las variables que intervienen para asignarles un valor y obtener el valor de matriz así como el vector del problema específico. Para formar los valores aleatorios, se cuenta con un algoritmo que a partir del número de cuenta y el nombre del alumno, permite obtener una secuencia de números que pueden ser usados en la asignación de valores para las variables del problema, dichos valores son guardados dentro de una cadena generadora, la cual es una cadena de texto, que se guarda en la base de datos.

El SIEL integra dos tipos de exámenes:

Examen Básico: Para crear este tipo de examen, se utiliza un algoritmo que permite seleccionar un grupo de preguntas tomando como base, el puntaje de cada reactivo y cuya suma individual de reactivos da la calificación final (10 puntos).

Examen Avanzado: En este tipo de examen se van a tomar los reactivos dinámicos, por no contar con más reactivos para hacer una selección, se eligen los mismos reactivos para todos los exámenes, el hecho de ser reactivos dinámicos va a permitir que aunque sean los mismo reactivos, no se tengan las mismas respuestas; ya que estas son dinámicas al momento de hacer el examen, que finalmente ésta es la gran herramienta que aporta nuestro sistema, aún cuando no exista un banco muy grande de preguntas dinámicas estas variarán entre los alumnos, con motivo de la función mezcla que toma en cuenta el nombre y número de cuenta de cada alumno para generar valores aleatorios que servirán para asignar a cada una de las variables del reactivo.

Cuando el alumno finaliza el examen el sistema graba las opciones elegidas en la base de datos, para en un primer momento generar el reporte del examen que se acaba de realizar y si posteriormente el alumno o el profesor desean consultar los

resultados de determinado examen, podrán hacerlo al entrar al módulo correspondiente y seleccionar la acción deseada (Consulta del Reporte).

Con el diseño e implementación del SIEL se logró caracterizar al Sistema de la siguiente forma:

- Instalar el sistema en el servidor del Departamento de Ingeniería de Control, con la siguiente dirección <http://132.248.59.112>, lo que permitirá que el Sistema se encuentre en línea para el profesor, quien está en posibilidades de programar un examen a sus alumnos.
- El profesor puede subir tanto reactivos fijos como dinámicos, estos últimos deben de cumplir con el formato previamente establecido.
- Respecto a los reactivos dinámicos, el profesor puede verificar los resultados del reactivo a elegir para comprobar su correcta ejecución.
- El profesor puede dar de alta a alumnos ya sea individualmente o en grupo por medio de un archivo con formato xml, el cual debe contener los siguientes columnas: nombre, número de cuenta y el número de grupo.
- El profesor puede consultar los reactivos existentes, para realizar modificaciones o darlos de baja.
- Puede consultar los grupos existentes, así como dar de baja a algún grupo en específico.

- Pueden consultar los reportes de los exámenes realizados por los alumnos, eligiendo primero el grupo y posteriormente al alumno del cual se quieren obtener los reportes que ha realizado.
- En relación con el alumno, este puede realizar los dos tipos de exámenes que ofrece el sistema, consultar los reportes de los mismos donde aparecerá únicamente la calificación obtenida y realizar acciones administrativas de su cuenta, al poder cambiar datos como usuario, *password*, correo electrónico.

Expectativas y Mejoras al Sistema:

- Para el despliegue en pantalla con una resolución aceptable del diagrama del circuito que corresponda a cada reactivo, se sugiere que el archivo de imagen se realice en SMART DRAW (Software con costo), sin embargo puede realizarse en cualquier otra aplicación pero que permita manejar una resolución mínima de 600x600 p, aunque convendría realizarlo en un formato único.
- Se desarrollaron funciones para resolver un sistema de ecuaciones con números complejos con la finalidad de resolver reactivos con fasores, sin embargo la función debe afinarse para tomar fuentes dependientes.
- El SIEL puede ser ampliado para obtener medidas estadísticas a nivel grupal y ver la comparación de avance entre grupos, así como en la incorporación de reactivos que abarquen otros temas más complejos del área.

Aún cuando actualmente existen soluciones que aparentemente pueden satisfacer el propósito de este proyecto, se encontró útil y enriquecedor el diseño y la elaboración de un sistema, hecho a la medida que cubriera diferentes expectativas

en materia de evaluación de una asignatura específica. Si bien la navegación en el sistema es a través de un método aparentemente lineal y estático, el corazón del sistema no es así, ya que el desarrollo del motor generador y solucionador de problemas en este caso para el área de circuitos eléctricos, manifestó ser una aportación importante en la educación de los alumnos, ya que se explota la generación de reactivos dinámicos; es decir, aunque sea el mismo alumno los parámetros del reactivo serán diferentes, debido a su uso auto evaluativo, demuestra que es una alternativa substancial, en el desarrollo profesional de los alumnos y una herramienta eficaz de valoración de la enseñanza impartida por los profesores.

El sistema queda completamente abierto a modificaciones, mejoras e incorporación de módulos que puedan ser útiles en otros ámbitos; no sólo en el área de circuitos eléctricos, para este propósito se entrega junto con este trabajo los programas fuentes debidamente comentados, para que aquellos que quieran continuar enriqueciendo esta aportación sean bienvenidas en aras de generar una herramienta poderosa que permita ser un elemento de evaluación a las futuras generaciones del área de Ingeniería y por qué no de otras disciplinas.

Se han cumplido las expectativas y alcances que se definieron al inicio de este documento, pero también se abrieron nuevas oportunidades de desarrollo de soluciones a problemas que inicialmente no se notaron claramente, dichos problemas pueden ser abordados en un futuro con una visión diferente a la que al momento de realizar esta investigación se contaba, dicha visión es lo que hemos adquirido y que en gran medida lo que nos hace mejores profesionistas y le otorga su razón de ser a la elaboración de este trabajo de tesis.

7. ANEXO A

7.1 INSTALACIÓN DE HERRAMIENTAS

Existen dos maneras de instalar Apache, PHP y MySQL. La primera de ellas consiste en instalar cada programa por separado y modificar los archivos de configuración de cada uno de ellos para ponerlos en funcionamiento.; la segunda es por medio de “instaladores” que automatizan el proceso de instalación conjunta de todas las aplicaciones que necesitamos, uno de los mejores y más conocidos es **XAMPP**¹ (acrónimos de Plataforma X, Apache, MySQL, PHP y Perl). XAMPP provee un ambiente completo de desarrollo web con PHP, Apache y MySQL y puede ser instalado al descargar, descomprimir y correr el instalador. XAMPP hace la instalación extremadamente sencilla y es completamente libre y está disponible para Windows, Linux, Mac Os y Solaris.

7.1.1 INSTALACIÓN DE XAMPP

Para empezar se debe descargar el software de la página <http://www.apachefriends.org/en/xampp.html>, y debido a que esta instalación se hará en Windows la versión que se descargará será la correspondiente a XAMPP for Windows (Ver figura A.2.).

¹ <http://www.apachefriends.org/en/xampp.html>

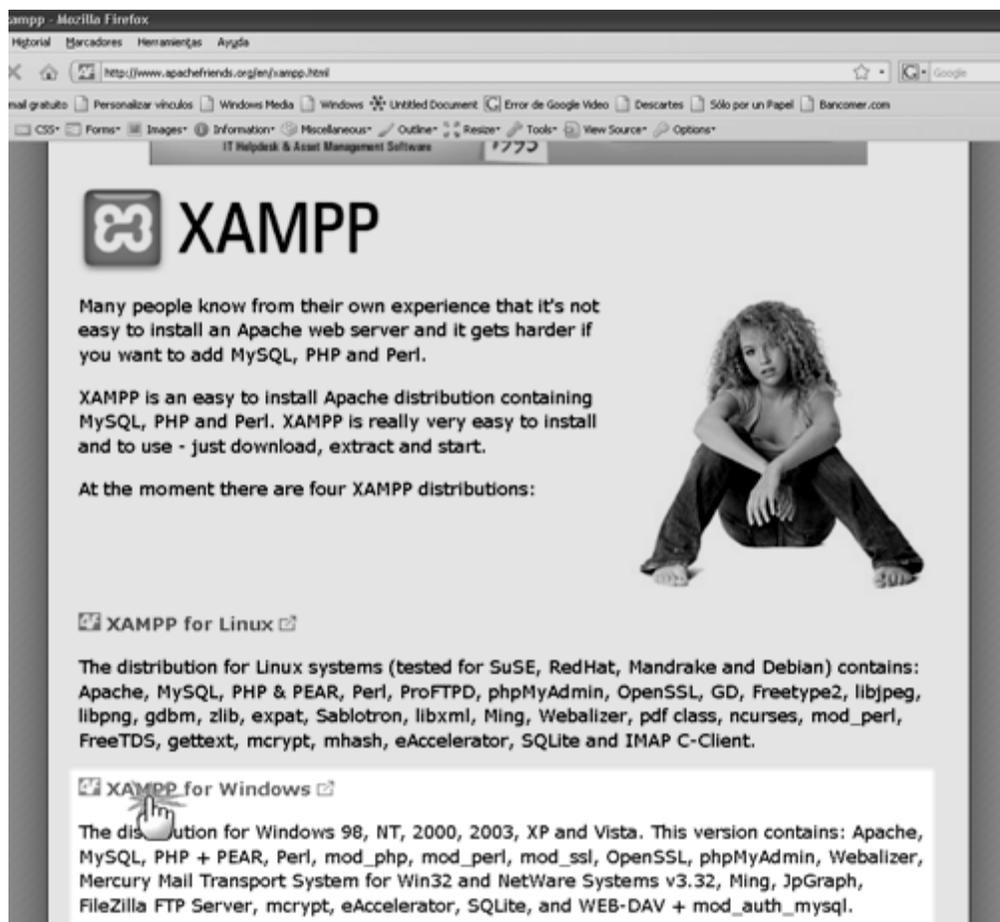


Figura A.2. Página de descarga de XAMPP

La última versión es la 1.7.0 e incluye las versiones de Apache, PHP y MySQL, Filezilla y otros como se observa en la Figura A.3. Se puede descargar XAMPP en tres variantes (Ver Figura A.4), por conveniencia en la instalación se debe descargar la variante **Installer**.

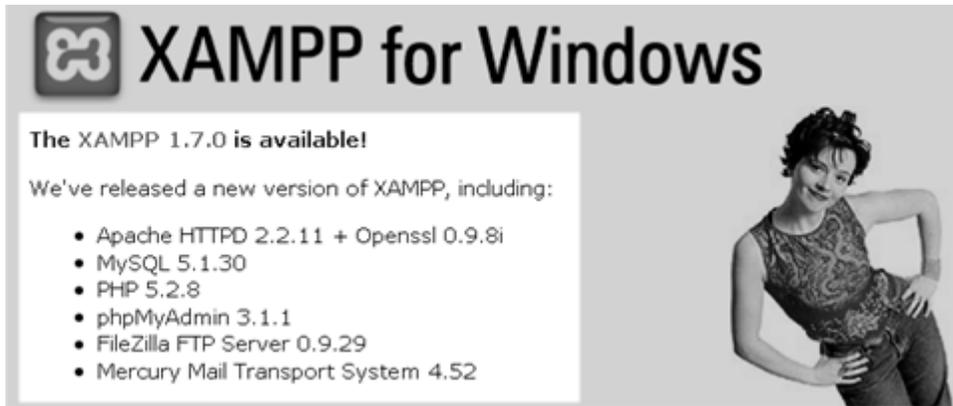


Figura A.3 Versiones incluidas

Download

XAMPP

You can download XAMPP for Windows as three different variations:

Installer
Easy and safe: XAMPP with a comfortable installer.

ZIP archive
For purists: XAMPP as ordinary ZIP archive.

Self-extracting ZIP archive
Economical: XAMPP as very small self-extracting 7-ZIP archive.

XAMPP for Windows 1.7.0, 2008/12/22

Version	Size	Content
XAMPP Windows 1.7.0 [Basic package]		Apache HTTPD 2.2.11, MySQL 5.1.30, PHP 5.2.8 + Switch, Openssl 0.9.8i, phpMyAdmin 3.1.1, XAMPP Control Panel 2.5, Webalizer 2.01-10, Mercury Mail Transport System v4.52, FileZilla FTP Server 0.9.29, SQLite 2.8.15, ADOdb 4.98, Zend Optimizer 3.3.0, XAMPP Security, Ming. For Windows 2000, 2003, XP, VISTA. See also README
Installer	39 MB	Installer MD5 checksum: cfee3ed914428c7ea11b8f280bd1793a
ZIP	82 MB	ZIP archive MD5 checksum: 6ca2be44086d67f9d52a44f834873773
EXE (7-zip)	33 MB	Selfextracting 7-ZIP archive MD5 checksum: 94bb94acbcfc56141e51a18fb67ebef7

Figura A.4 Variantes de descarga

Se procede entonces a ejecutar el archivo descargado (Ver Figura A.5) y de aquí en adelante el trabajo de instalación es casi completamente automático, sólo hace falta especificarle al instalador en que unidad y carpeta deseamos realizar la instalación (Ver Figura A.6), se recomienda utilizar una carpeta diferente a Archivos de programa, sobre todo si se es usuario de Windows Vista, ya que sus

restricciones en permisos de escritura pudiera causar algún tipo de inconveniente. Si se tiene alguna otra partición diferente a aquella en la que se encuentre el sistema operativo, lo mejor será instalar XAMPP en ella.



Figura A.5 Pantalla de bienvenida

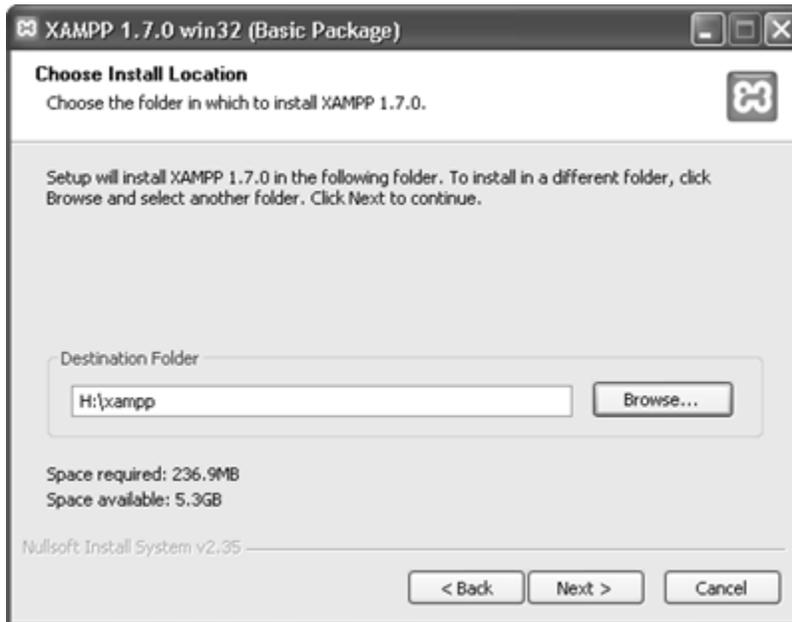


Figura A.6 Elección del directorio de instalación

Después de este paso, es necesario indicarle al instalador si se desea que haya un acceso directo a XAMPP en el escritorio y en la lista del menú de programas; además, si se requiere que Apache, MySQL y Filezilla sean programas que corran como servicios, se deja sin marcar las opciones de servicios y se continúa con la instalación (Ver Figura A.7).

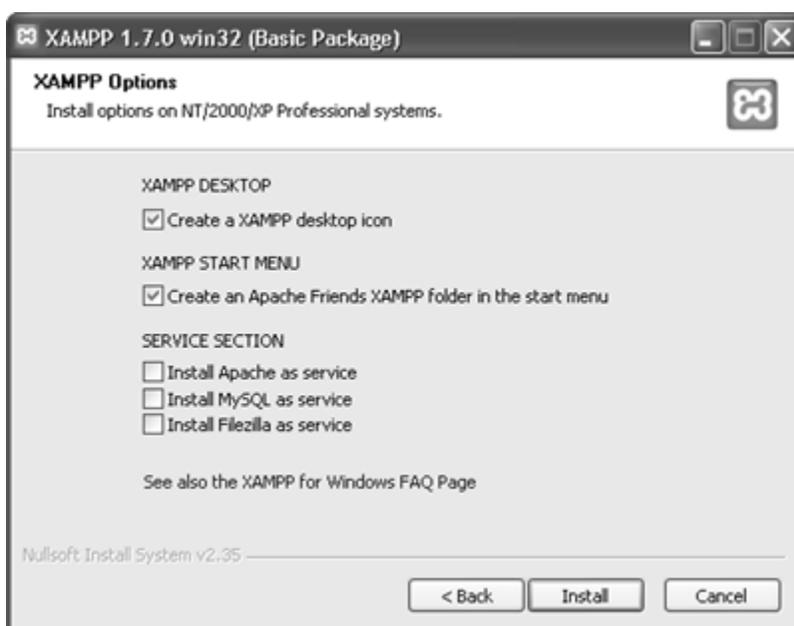


Figura A.7 Instalar MySQL y Apache como servicios

Al finalizar la instalación se pregunta al usuario si desea abrir el panel de control de XAMPP, al presionar **Si**, se desplegará el panel en la parte inferior derecha de la pantalla; para iniciar Apache, MySQL o Filezilla basta con presionar el botón correspondiente **Start** (iniciar) que cambiará a **Stop** (parar) si desea detener la ejecución de los mismos. También puede instalar o desinstalar los servicios de los mismos o verificar el estado en el que se encuentren estos. (Ver figura A.8)

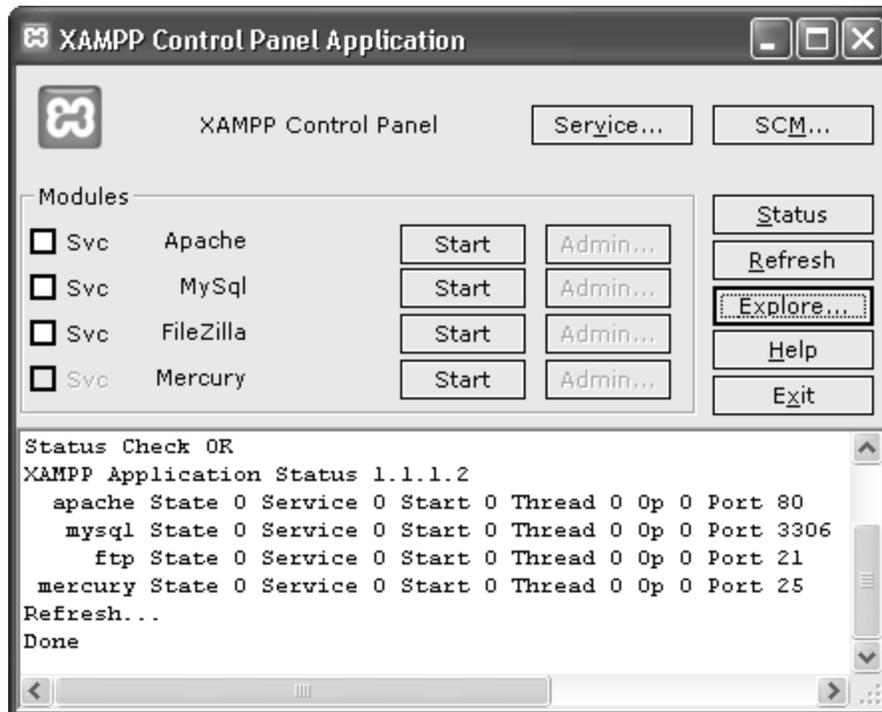


Figura A.8. Panel de control de XAMPP.

Se inicia entonces Apache y MySQL, aparecerá la leyenda **Running** (corriendo); para probar la instalación, es necesario abrir el navegador y escribir **http://localhost** o **http://127.1.1** en la barra de direcciones, se verá una página en donde se elige el idioma y una vez seleccionado será redireccionado a una página de bienvenida (Ver Figura A.9). Del lado izquierdo hay un menú, es necesario dar un clic en **Estado** y ver que todos los componentes necesarios están corriendo (Ver Figura A.10), si hasta este punto todo ha funcionado como ha sido descrito, entonces se ha instalado correctamente el servidor Apache.

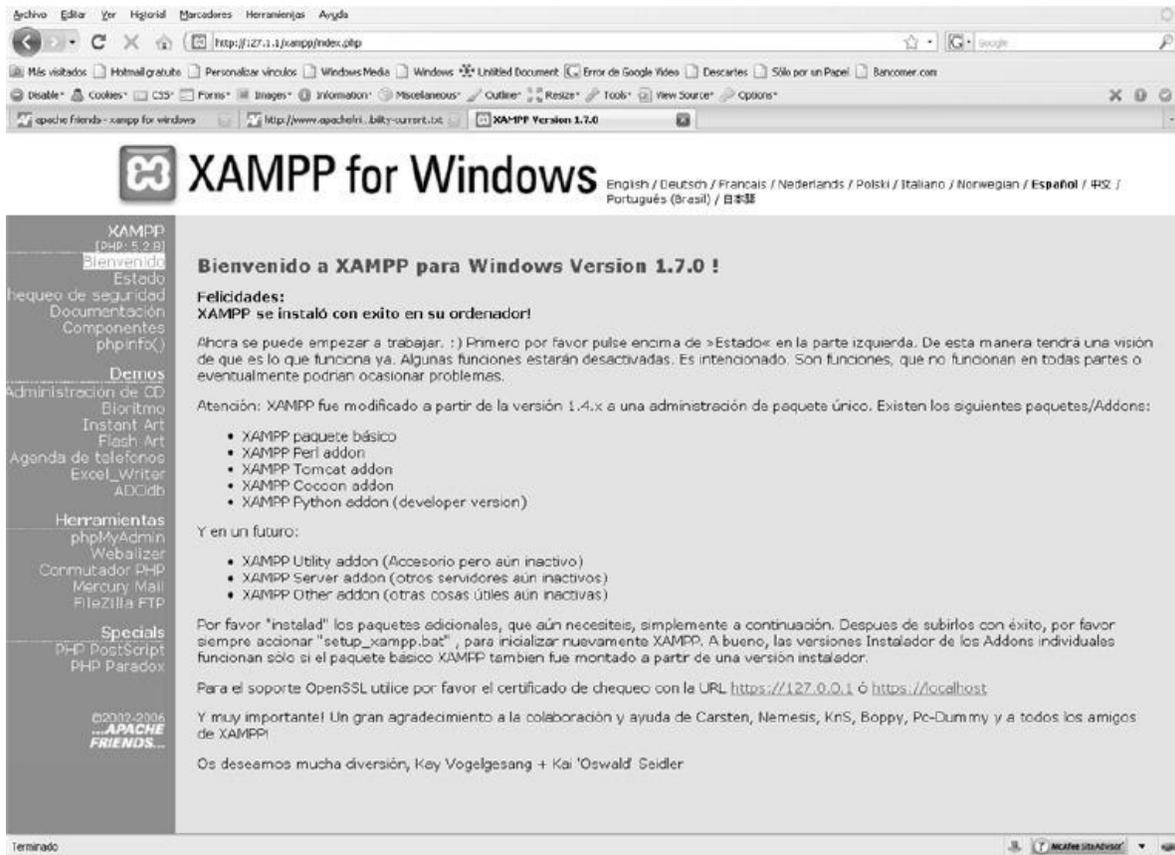


Figura A.9 Pantalla de bienvenida de XAMPP

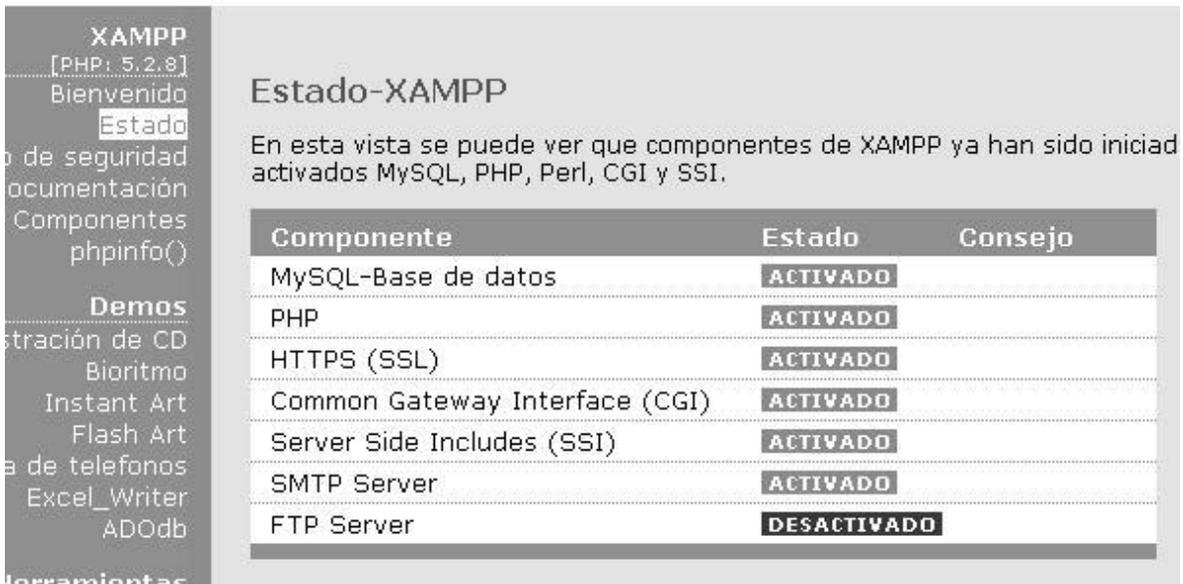


Figura A.10 Comprobación de estado de los componentes

Posteriormente se tiene que configurar la protección por contraseña de la carpeta XAMPP y de MySQL, para ello hay que escribir `http://localhost/security/index.php` en el navegador e ingresar el usuario y contraseña para ambas secciones (Ver Figura A.11) y después presionar **Password changing** (cambiar contraseña) para la sección de MySQL y posteriormente presionar **Make safe the XAMPP directory** (hacer seguro el directorio XAMPP), con esto se protege el acceso al servidor y a la base de datos.

The screenshot shows a web interface titled "Security console MySQL & XAMPP directory protection". It is divided into two main sections:

- MYSQL SECTION: "ROOT" PASSWORD**
 - MySQL SuperUser: root
 - Current password: [password field]
 - New password: [password field]
 - Repeat the new password: [password field]
 - PhpMyAdmin authentication: http cookie
 - Security risk! ----
 - Safe plain password in text file?
(File: C:\xampp\security\security\mysqlrootpasswd.txt)
 - Button: Password changing
- XAMPP DIRECTORY PROTECTION (.htaccess)**
 - User: root
 - Password: [password field]
 - Security risk! ----
 - Safe plain password in text file?
(File: C:\xampp\security\security\xamppdirpasswd.txt)
 - Button: Make safe the XAMPP directory

Figura A.11 Protección por contraseña de MySQL y la carpeta de XAMPP

8. ANEXO B

8.1 CÓDIGO FUENTE

Tabla 4.9.

```
6 |
7 function RProfesorBasico($reporteTipo,$sid){
8 //Se buscan los datos del alumno en la tabla alumnos
9 $consulta=mysql_query("SELECT * FROM alumnos WHERE IdAlumnos='$sid'");
10 $row=mysql_fetch_row($consulta);
11
12     $sid=$row[0];
13     $cuenta=$row[2];
14     $nombre=$row[3];
15     $usuario=$row[4];
16     $password=$row[5];
17     $email=$row[6];
18     $grupo=$row[1];
19
20 echo "<h2>Reporte Básico en Detalle del Alumno ".$nombre."<br> Reporte : "
21 ".$reporteTipo."</h2>";
22
23 ///////////////////////////////////////////////////CUERPO DEL REPORTE////////////////////////////////////
24
25 //Se buscan los datos del examen en las tablas reactivos y respuestasfijas
26
27 $query = "SELECT reactivo.imagen, reactivo.enunciado, reactivo.idreactivos,
28 respuesta.opcion_elegida, respuesta.puntaje
29 FROM reactivos AS reactivo
30 JOIN respuestasfijas AS respuesta
31 ON reactivo.idreactivos = respuesta.reactivos_idreactivos
32 WHERE respuesta.reportes_idreportes = '$reporteTipo'";
33
34 $consulta=mysql_query($query);
35 $indice=1;
36 while ($row=mysql_fetch_row($consulta))
37 {
38     $diagrama = $row[0];
39
```

```
40
41 $enunciado = $row[1];
42 $reactivo= $row[2];
43 $opcion = $row[3]; //Número de la respuesta elegida por el alumno
44 $puntaje = $row[4];
45
46 //De acuerdo al número de la respuesta se buscan las 4 opciones
47 //para elegir la elegida por el alumno
48 $query1 = "SELECT opcion1, opcion2, opcion3, opcion4
49 FROM reativofijo
50 WHERE reactivos_idreactivos = '$reactivo'";
51 $consulta1 = mysql_query($query1);
52 $row1 = mysql_fetch_row($consulta1);
53 switch($opcion){
54     case '1':
55         $alumnorespuesta = $row1[0];
56         break;
57     case '2':
58         $alumnorespuesta = $row1[1];
59         break;
60     case '3':
61         $alumnorespuesta = $row1[2];
62         break;
63     case '4':
64         $alumnorespuesta = $row1[3];
65         break;
66     default:
67         break;
68 }
69
69
70 echo "<b>Pregunta ".$indice."</b>: </b>".$enunciado."<br>";
71 echo "<img src='upload/$diagrama' width='300' height='260'>". "<br>";
72 echo "<b>Respuesta Correcta :</b>".$row1[0]."<br>";
73
74
75 echo "<b>Respuesta Alumno :</b>".$alumnorespuesta."<b> Calificacion : "
76 . $puntaje."</b><br><br>";
77 $indice++;
78 }
79
80 ///////////////FIN CUERPO DEL REPORTE////////////////////
81
82 //Se busca la calificación del reporte de acuerdo al identificador del alumno
83 //y el identificador del reporte.
84
85 $consulta=mysql_query("SELECT a.Nombre, r.Calificacion
86 FROM alumnos AS a
87 INNER JOIN reportes AS r ON a.IdAlumnos = r.alumnos_IdAlumnos
88 WHERE a.IdAlumnos = '$id'
89 AND r.idReportes = '$reporteTipo'
90 ");
91
92 $row=mysql_fetch_row($consulta);
93 $nombre=$row[0];
94 $calificacion=$row[1];
95 echo "<h3>Calificaci&oacute;n Final :".$calificacion."</h3>";
96
97 }
98
```

Tabla 4.10.

```

101 |
102 function RProfesorAvanzado($idreporte,$id) {
103
104 //Se buscan lo datos generales del alumno en la tabla alumnos
105
106 $consulta=mysql_query("SELECT * FROM alumnos WHERE IdAlumnos='$id'");
107 $row=mysql_fetch_row($consulta);
108
109     $id=$row[0];
110     $cuenta=$row[2];
111     $nombre=$row[3];
112     $usuario=$row[4];
113     $password=$row[5];
114     $email=$row[6];
115     $grupo=$row[1];
116
117 echo "<h2>Reporte Avanzado en Detalle del Alumno ".$nombre.
118 "<br> Reporte :".$idreporte."</h2>";
119
120 //Se buscan en la tabla reactivos y respuestadinam los datos del
121 //examen realizado por el alumno de acuerdo al identificacor del
122 // reporte.
123
124 $sql="SELECT r.enunciado,r.imagen, d.opcion1,d.opcion2, d.opcion3,d.opcion4,
125 d.opcion_elegida, d.cadena_generadora, d.puntaje, d.opcion_correcta
126 FROM reactivos AS r
127 JOIN respuestadinam AS d
128 ON r.idreactivos=d.reactivos_idreactivos
129 WHERE d.reportes_idreportes = '$idreporte'";
130 $consulta=mysql_query($sql);
131 $index=1;
132 $scalifTotal=0;
133
134
135 //Se crea un arreglo $respuesta[], donde se guardan las respuestas
136 //generadas del sistema.
137 while($row=mysql_fetch_row($consulta)){
138     $respuesta[0]=$row[2];
139     $respuesta[1]=$row[3];
140     $respuesta[2]=$row[4];
141     $respuesta[3]=$row[5];
142     $elegida=$row[6]-1; //Indice de la opción elegida por el alumno
143     $correcta=$row[9]-1;//Indice de la opción correcta generada por el sistema
144     echo "<b>Pregunta $index:</b>";echo $row[0]."<br>";
145     echo "<br><img src='upload/$row[1]' height= '300'><br>";
146     echo $row[7]."<br>";
147     echo "<b>Opcion correcta:</b>$respuesta[$correcta].<br>";
148     echo "<b>Opcion elegida:</b>$respuesta[$elegida]. <b>Puntaje:</b>
149     $row[8].<br>"; //opcion elegida
150     echo "<hr>";
151     $scalifTotal=$scalifTotal+$row[8];
152     $index++;
153 }
154 echo "<b>Calificacion : </b>$scalifTotal";
155
156 }
157

```

Tabla 4.13.

```

170 function Combinacion ($tipo){
171 //Función encargada de regresar un arreglo
172 //con los identificadores de preguntas
173 //que se van a usar en el examen.
174
175 //Se seleccionan los reactivos de acuerdo
176 //al nivel
177
178
179 $sql="SELECT idreactivos FROM reactivos
180 WHERE nivel='$tipo'";
181 $consulta=mysql_query($sql);
182 $indice=0;
183 while ($row=mysql_fetch_row($consulta)){
184 $preguntas[$indice]=$row[0];
185 //echo "Indice $indice:$preguntas[$indice]<br>";
186 $indice++;
187 }
188
189 //Se crean las combinaciones de 4 que se pueden
190 //formar con los identificadores de preguntas.
191
192 $pivote[0]=0;$pivote[1]=1;$pivote[2]=2;$pivote[3]=3;
193 $total=$indice;
194 $indice=0;
195 for ($i=$pivote[0];$i<($total-3);$i++){
196     $pivote[0]=$i;$pivote[1]=$i+1;$pivote[2]=$i+2;$pivote[3]=$i+3;
197     for ($j=$pivote[1];$j<($total-2);$j++){
198         for ($k=$pivote[2];$k<($total-1) ;$k++){
199             for ($l=$pivote[3];$l<$total;$l++){
200
201                 $combinacion[$indice]="$preguntas[$i], $preguntas[$j],
202                 |$preguntas[$k], $preguntas[$l]";
203
204
205                 $indice++;
206             }
207             $pivote[1]++;$pivote[2]++;$pivote[3]++;
208         }
209     }
210 }
211 }
212 for ($i=0;$i<$indice;$i++){
213 }
214
215 //De todas las combinaciones se elige aleatoriamente una
216 //Como es una cadena de texto con los 4 identificadores
217 //Se usa la función explode para separar los 4 identificadores
218 //y ponerlos en un arreglo. Se guarda en el arreglo
219 //$PreguntaCombinacion
220
221 $aleatorio=mt_rand(0, ($indice-1));
222 $PreguntaCombinacion=explode(" ", $combinacion[$aleatorio]);
223 return $PreguntaCombinacion;
224 }
225

```

Tabla 4.14.

```

220
229 function InsertBasico ($id,$num_intento,$preguntas,$tipo){
230 $num_intento++;
231 // SE ACTUALIZA EL NUMERO DE INTENTO DE LA TABLA ALUMNOS
232 switch($tipo){
233     case "Básico":
234         mysql_query("Update alumnos Set intento_b='$num_intento'
235             Where IdAlumnos='$id'");
236         break;
237     case "Intermedio":
238         mysql_query("Update alumnos Set intento_i='$num_intento'
239             Where IdAlumnos='$id'");
240         break;
241     default:
242         break;
243 } //SE CREA EL REPORTE DEL EXAMEN
244 mysql_query("Insert into reportes (alumnos_idalumnos, fecha, tipo)
245 values ('$id', now(), '$tipo')");
246 //SE GUARDAN LOS NUMEROS DE LAS PREGUNTAS EN TABLA RESPUESTASEXAMEN
247 //Se crea la variable de sesión 'reporte', la cual guarda el identificador
248 //de reporte.
249 $id_last_report=mysql_insert_id();
250 $_SESSION['reporte']=$id_last_report;
251 $numero=count($preguntas)+1;
252 for($i=0;$i<$numero;$i++) {
253     $question=$preguntas[$i];
254     mysql_query("Insert into respuestasfijas
255 ( reportes_idreportes, reactivos_idreactivos )
256 values
257 ('$id_last_report', '$question') ");
258 }
259 echo "<br>";
260 }
261

```

Tabla 4.15.

```

263
264 function PreguntaBasico($idReactivo, $indice){
265 //Se despliega la información del reactivo elegido
266 $numero = $indice+1; //Contador que indica el número de reactivo
267 //Se busca en la tabla reactivos y reativofijo los datos de la
268 //pregunta
269 $query="SELECT reactivo.enunciado, reactivo.imagen,
270 respuesta.opcion1, respuesta.opcion2, respuesta.opcion3,
271 respuesta.opcion4
272 From reactivos AS reactivo
273 JOIN reativofijo AS respuesta
274 ON reactivo.idreactivos = respuesta.reactivos_idreactivos
275 WHERE reactivo.idreactivos = '$idReactivo'";
276 $consulta=mysql_query($query);
277 $row=mysql_fetch_row($consulta);
278
279 $pregunta=$row[0];
280 $diagrama=$row[1];
281 $respuesta[0]=$row[2]; //arreglo con las 4 respuestas del reactivo
282 $respuesta[1]=$row[3];
283 $respuesta[2]=$row[4];
284 $respuesta[3]=$row[5];
285 echo "
286 <h3>$numero. $pregunta :</h3>
287 <td rowspan='2'><img src='upload/$diagrama' width='300' height='260'></td>";
288 $radio="respuesta".$numero;
289 $respuesta1=shuffle($respuesta); //se desordenan las preguntas
290 DespliegueRadiol ($radio,$respuesta); //se despliegan las respuestas
291 //con un radio button.
292 }
293

```

Tabla 4.16.

```

296
297 function DespliegueRadiol ($radio, $opcion){
298 $index=1;
299 echo "<br>";
300 for ($i=0;$i<4;$i++){
301     if($i==0){
302         echo "<INPUT TYPE='RADIO' NAME='$radio' VALUE='$opcion[$i]' >
303         | $opcion[$i]<br> ";
304     }
305     else
306     {
307         echo "<INPUT TYPE='RADIO' NAME='$radio' VALUE='$opcion[$i]' >
308         $opcion[$i]<br> ";
309     }
310
311     $index++;
312 }
313
314 }
315
316

```

Tabla 4.17.

```

318
319 function PreguntaBasicoPost($idreporte){
320 //Despliegue de una pregunta despues de hacer una recarga de la página
321 //del examen básico. Se buscan los datos del reactivo
322 $numero = 1;
323 $sql="SELECT reactivos_idreactivos FROM respuestasfijas
324 WHERE reportes_idreportes='$idreporte'";
325 $consulta=mysql_query($sql);$index=1; // numero de pregunta
326 while($row=mysql_fetch_row($consulta)){
327     $pregunta=$row[0];
328     $sql1="SELECT r.enunciado, r.imagen, d.opcion1, d.opcion2,
329     d.opcion3, d.opcion4
330     FROM reativofijo AS d
331     JOIN reactivos AS r
332     ON d.reactivos_idreactivos=r.idreactivos
333     WHERE r.idreactivos = '$pregunta'";
334     $consulta1=mysql_query($sql1);
335     while($row1=mysql_fetch_row($consulta1)){
336         $respuesta[0]=$row1[2];
337         $respuesta[1]=$row1[3];
338         $respuesta[2]=$row1[4];
339         $respuesta[3]=$row1[5];
340         echo "<b>Pregunta $index:</b>";echo $row1[0]."<br>";
341         echo "<br><img src='upload/$row1[1]' height= '270'><br>";
342         echo $row1[7]."<br>";
343         $radio="respuesta".$numero;
344         //se desordenan las respuestas del reactivo
345         $respuesta1=shuffle($respuesta);
346         //Se despliegan las respuestas en radio buttons.
347         DespliegueRadiol($radio,$respuesta);    }
348         $index++;
349         $numero++;    }
350 }
351

```

Tabla 4.18.

```

353
354 function RespuestaBasico($idreporte,$alumno){
355 $sql="SELECT f.reactivos_idreactivos,f.opcion1,f.opcion2,f.opcion3,
356 f.opcion4 FROM reativofijo AS f
357 JOIN respuestasfijas AS r
358 ON f.reactivos_idreactivos=r.reactivos_idreactivos
359 WHERE r.reportes_idreportes ='$idreporte'";
360 $consulta=mysql_query($sql);
361 $indice=0;
362 $CalifTotal=0;
363 while ($row=mysql_fetch_row($consulta)){
364 $pregunta=$row[0];
365 $correcta=$row[1];
366 $opcion[0]=$row[1];$opcion[1]=$row[2];$opcion[2]=$row[3];$opcion[3]=$row[4];
367 $index=0;
368 //Se busca el indice que corresponde a la respuesta elegida por el alumno
369 //entre las respuestas que hay de ese reactivo
370 for($i=0;$i<4;$i++){
371 if($opcion[$i]==$alumno[$indice]){
372 $elegida=$index+1;
373 break;
374 }
375 $index++;
376 }//Si la respuesta del alumno corresponde a la respuesta correcta en el
377 //sistema se da calificación aprobatoria
378 if ($correcta==$alumno[$indice]) {
379 $calif=10/4;
380 }
381 else
382 {
383 $calif=0;
384 }
385 $CalifTotal=$CalifTotal+$calif;
386 }
387 $indice++;
388 //Se guardan el indice de la respuesta del alumno y la calificación
389 //del reactivo
390 mysql_query("Update respuestasfijas Set
391 opcion_elegida= '$elegida', puntaje='$calif'
392 Where reactivos_idreactivos='$pregunta' and reportes_idreportes='$idreporte'");
393 //Se guarda la calificación total del examen
394 mysql_query("UPDATE reportes SET calificacion='$CalifTotal'
395 WHERE idreportes='$idreporte'");
396 }
397 }
398 }
399 }

```

Tabla 4.19.

```
403
404 function ReporteBasico($reporte, $respuesta){
405
406 //Se buscan los datos de la pregunta para ser desplegados en el reporte del examen
407 $query = "SELECT reactivo.imagen, reactivo.enunciado, reactivo.idreactivos,
408 respuesta.opcion_elegida, respuesta.puntaje
409 FROM reactivos AS reactivo
410 JOIN respuestasfijas AS respuesta
411 ON reactivo.idreactivos = respuesta.reactivos_idreactivos
412 WHERE respuesta.reportes_idreportes = '$reporte'";
413
414 $consulta=mysql_query($query);
415 $indice=1;
416 $CalifTotal=0;
417 while ($row=mysql_fetch_row($consulta))
418 {
419 $enunciado = $row[1];
420 $reactivoId= $row[2];
421 $opcion = $row[3];
422 $puntaje = $row[4];
423 $index=$indice-1;
424 $CalifTotal=$CalifTotal+$puntaje;
425 echo "<b>Pregunta ".$indice."<b> Calificación :</b>".$puntaje."<hr>";
426 $indice++;
427
428 }
429 echo "<b>Calificación Total :</b> $CalifTotal <br>";
430
431 }
432
```

Tabla 4.21.

```
435
436 function SetIntentos($intento,$id){
437 //Actualiza el número de intentos en la tabla alumnos
438 //del examen avanzado
439 $intento++;
440 $sql="UPDATE alumnos SET intento_a='$intento' WHERE
441 idalumnos='$id'";
442 mysql_query($sql);
443 }
444
445
```

Tabla 4.22.

```
448
449 function ReporteAvanzado($idalumnos, $preguntas){
450 //Se crea el reporte con los datos del identificador del alumno
451 //fecha de realización y tipo de examen
452
453 $sql = "INSERT into reportes
454 (alumnos_idalumnos, fecha,tipo)
455 VALUES ($idalumnos, now(), 'Avanzado)";
456 mysql_query($sql);
457 $id_reporte = mysql_insert_id();
458
459 $numero = count($preguntas);
460 //Se guardan los identificadores de reactivos elegidos para
461 // el examen
462     for($i=0;$i<$numero;$i++){
463         $reactivo = $preguntas[$i];
464         $sql = "INSERT into respuestasdinam
465 (reportes_idreportes, reactivos_idreactivos)
466 VALUES ($id_reporte, $reactivo)";
467         mysql_query($sql);
468     }
469
470 return $id_reporte;
471 }
472
```

Tabla 4.23.

```
475
476 function DatosPregunta($id_reactivos){
477 $sql = "SELECT reactivo.imagen, reactivo.enunciado, matriz.valor ,
478 matriz.nombre
479 FROM matrices AS matriz
480 JOIN reactivos AS reactivo
481 ON reactivo.idreactivos=matriz.reactivos_idreactivos
482 WHERE reactivos_idreactivos='$id_reactivos'";
483 $consulta = mysql_query($sql);
484 $row = mysql_fetch_row($consulta);
485 return $row;
486 }
487
```

Tabla 4.24.

```
490 function CadenaMatriz ($cadena){//////////
491 //Función que se encarga de separar los elementos de la matriz
492 $separacion = explode("fila",$cadena); //Se obtienen los renglones de la matriz
493 //Se obtienen los elementos del primer renglón
494 $MatrizValor = explode(",",$separacion[1]);
495 array_pop($MatrizValor);
496 $filas = count($separacion);
497 $columnas = $filas-1;
498
499 for ($i=2;$i<$filas;$i++){
500     //Se obtienen los elementos de los demás renglones
501     $pivote = explode(",",$separacion[$i]);
502     for ($j=0;$j<$columnas;$j++){
503         $agregar = $pivote[$j];
504         array_push($MatrizValor,$agregar);
505     }
506 }
507
508 return $MatrizValor;
509 }
510
```

Tabla 4.25.

```
513 function SelectVector($id_reactivos){
514 //Función que regresa la cadena del vector
515 $sql = "SELECT nombre, valor from matrices WHERE reactivos_idreactivos='$id_reactivos'";
516 $consulta = mysql_query($sql);
517 //Se cuentan con dos renglones el primer es el renglón que contiene
518 //los datos de la cadena de la matriz y el segundo la cadena del
519 //vector
520 while($row = mysql_fetch_row($consulta))
521 {
522     $cadena[0]=$row[0];
523     $cadena[1] = $row[1];
524 }
525 return $cadena;
526 }
527
```

Tabla 4.26.

```
529
530 function CadenaVector ($cadena){
531 //Se separa el nombre del vector de los elementos
532 $vector=explode("(", $cadena);
533 $VectorValor= explode(",",$vector[1]); //Se obtienen los elementos del vector
534 return $VectorValor;
535 }
536
```

Tabla 4.27.

```

538
539 function GetNumResistencias($id_reactivos){
540 $ssql = "SELECT count(literales) FROM literales
541 WHERE tipo = '0' AND
542 reactivos_idreactivos ='$id_reactivos'";
543 $consulta = mysql_query($ssql);
544 $row = mysql_fetch_row($consulta);
545 return $row[0];
546 }
547

```

Tabla 4.28.

```

549
550 function GetNumFem($id_reactivos){
551 $ssql = "SELECT count(literales) FROM literales
552 WHERE tipo = '1' AND
553 reactivos_idreactivos ='$id_reactivos'";
554 $consulta = mysql_query($ssql);
555 $row = mysql_fetch_row($consulta);
556 return $row[0];
557 }
558

```

Tabla 4.29.

```

561 function VectorOpciones ($renglon, $columna, $resistencia,
562 $MatrizValor, $TipoMatriz,$peticion, $VectorValor, $fem,$reporte,$reactivos){
563 $correcta=mt_rand(0,3); //Se elige aleatoriamente cual va a ser
564 //el conjunto de valores de las variables se generan las 4 respuestas
565 $CadenaGeneradora = "";
566 for($k=0;$k<4;$k++){
567 $VResistencias = ValorResistencias($resistencia);
568 $VPeticiones = ValorPeticiones($fem);
569 if($k==$correcta){
570 //La respuesta que corresponda a la respuesta correcta elegida con anterioridad
571 //por el sistema es la que se despliega con sus valores para que el alumno pueda
572 //resolverla y es la que se guarda como respuesta correcta
573 $show=0;
574 $CadenaGeneradora= ShowVariables($resistencia,$VResistencias,$fem,$VPeticiones,
575 $peticion);
576 }else
577 {
578 $show=0;
579 }
580 //Se obtienen los valores de la matriz.
581 $M=MatrizExpresion1($renglon, $columna, $resistencia, $MatrizValor,
582 $TipoMatriz,$VResistencias,$show);
583 //Se obtienen los valores del vector
584 $V=VectorUno($columna, $renglon, $peticion, $VectorValor, $VPeticiones,$show);
585 //Se resuelve el sistema de ecuaciones
586 $Xfinal=Jacobi($renglon,$columna,$V,$M);
587
588 $opcion[$k]="";
589 //Se guardan los resultados del sistema de n ecuaciones
590 for($i=0;$i<$renglon;$i++){
591 $opcion[$k] .= "V".($i+1). " = ".round($Xfinal[$i],2)." [V] :";
592 }
593

```

```
593
594 }
595 //Se guarda la opción correcta, las 4 respuestas del reactivo
596 //La cadena generadora que contiene los datos de las variables
597 //los identificadores de reactivos
598 RespuestaDinamica($correcta,$opcion, $CadenaGeneradora,$reactivos, $reporte);
599 return $opcion;
600 }
601
```

Tabla 4.30.

```
604 function ValorResistencias($resistencia){
605     for ($i=0;$i<$resistencia;$i++) {
606         $r[$i]=mt_rand(1,18);
607     }
608     return $r;
609 }
610 }
611
```

Tabla 4.31.

```
613
614 function ValorPeticiones($fem){
615     for ($i=0;$i<$fem;$i++) {
616         $p[$i]=mt_rand(1,3);
617     }
618     return $p;
619 }
620 }
621
```

Tabla 4.32.

```
624 function ShowVariables($resistencia,$VResistencia,$fem,$VPeticiones,$datos){
625 //Se genera una cadena de texto con las variables involucradas con sus
626 //respetivos valores
627 $ok ="";
628 for ($i=0;$i<$resistencia;$i++) {
629     $ok .= "R".($i+1).".".$VResistencia[$i]. " ohms ; ";
630 }
631 for ($i=0;$i<$fem;$i++) {
632     switch($datos){
633     case "Corriente":
634         $ok .= "Is".($i+1).".".$VPeticiones[$i]. " [A]; ";
635         break;
636     case "Voltaje":
637         $ok .= "V".($i+1).".".$VPeticiones[$i]. " ; ";
638         break;
639     default:
640         break;
641     }
642 }
643 echo $ok."<br>";
644 return $ok;
645 }
646
```

Tabla 4.33.

```

649 function MatrizExpresion1($renglon, $columna,$resistencia,$valores,
650 $TipoMatriz,$VResistencias, $show){
651 ///////////////////////////////////////////////////////////////////
652 //Obtener valores de resistencias/////////
653 ///////////////////////////////////////////////////////////////////
654 //echo "<br>";
655 for ($i=0;$i<$resistencia;$i++) {
656     $r[$i]=$VResistencias[$i];
657 }
658 ///////////////////////////////////////////////////////////////////
659 $cont1=0;
660 for ($i=0;$i<$renglon;$i++){
661     for ($j=0;$j<$columna;$j++){
662         $matriz[$i][$j]= $valores[$cont1];
663
664         $separacion=explode("|", $valores[$cont1]);
665         $numero=count($separacion);
666         ///////////////////////////////////////////////////////////////////
667         //Numero de Parametros en la expresion////
668         ///////////////////////////////////////////////////////////////////
669         $Rt=0;
670         for ($k=0;$k<$numero;$k++){ //+++++++
671             $resis=$separacion[$k];
672             //asignar el signo*****
673             if (strstr($resis,"-")) {
674                 $signo="menos";
675             }
676             else{
677                 $signo="mas";
678             }
679             $posicion=strpos($resis,"R");
680             $len=strlen($resis);
681             |
682             $diff=$len-$posicion;
683             $pivote=substr($resis, ($posicion+1), 1);
684
685             for ($l=0;$l<$resistencia;$l++){ //*****
686                 $indice="";
687                 if ($pivote==1){
688                     if ($diff>=3){
689                         $buscar1=substr($resis, $posicion, 3);
690                         $indice=substr($resis, ($posicion+1), 2);
691                     }
692                     else{
693                         $buscar1=substr($resis, $posicion, 2);
694                         $indice=substr($resis, ($posicion+1), 1);
695                     }
696                 }
697             }
698             if ($indice==""){
699                 $buscar1=substr($resis, $posicion, 2);
700                 $indice=substr($resis, ($posicion+1), 1);
701             }
702             $indice--;
703             $buscar="R".(string)($l+1);
704             if (strstr($resis,$buscar)){
705                 ///////////////////////////////////////////////////////////////////
706                 switch ($TipoMatriz){//switch=====
707                     case "Resistencia":
708                         if ($signo=="mas"){
709                             if ($indice>=0 AND is_numeric($indice)){
710                                 $Rt=$Rt+$r[$indice];
711                             }
712                         }
713                         else{
714                             if ($indice>=0 AND is_numeric($indice)){
715

```

```

715
716     $Rt=$Rt-$r[$indice];
717     }
718     }
719     break;
720     case "Admitancia":
721     if($signo=="mas"){
722     if($indice>=0 AND is_numeric($indice)){
723     $Rt=$Rt+1/$r[$indice];
724     }
725     }
726     else{
727     if($indice>=0 AND is_numeric($indice)){
728     $Rt=$Rt-1/$r[$indice];
729     }
730     }
731     break;
732     default:
733     break;
734     } //switch=====
735     break;
736     ////////////////
737     }
738     } //*****
739     } //+++++++
740     $cont1++;
741     $A[$i][$j]=$Rt;
742     }
743 }
744 // VER VALORES DE MATRIZ
745 if ($show){
746 echo "Matriz de ".$TipoMatriz."<br>";
747 echo "<table width='400' border='1'>";
748
749 //-----
750 for ($i=0;$i<$renglon;$i++){
751     echo "<tr>";
752     for ($j=0;$j<$columna;$j++){
753     echo "<td align='center'>".$A[$i][$j]."</td>";
754     }
755     echo "</tr>";
756 }
757 //-----
758 echo "</table>";
759 }
760
761 return $A;
762 }
763

```

Tabla 4.34.

```

765
766 function VectorUno ($columna, $renglon, $datos, $valores,
767     $VPeticiones, $show) {
768     //////////////////////////////////////
769     //Obtener valor del Vector////////
770     //////////////////////////////////////
771     for ($i=0;$i<$columna;$i++) {
772         switch ($datos) {
773             case "Corriente":
774                 $current[$i]=$VPeticiones[$i];
775                 break;
776             case "Voltaje":
777                 $V[$i]=$VPeticiones[$i];
778                 break;
779             default:
780                 break;
781         }
782     }
783     //////////////////////////////////////
784     switch ($datos) {
785         case "Corriente":
786             $buscar="IS";
787             break;
788         case "Voltaje":
789             $buscar="VS";
790             break;
791         default:
792             break;
793     }
794     $cont1=0;
795     for ($i=0;$i<$renglon;$i++) { //for 1*****
796         $separacion=explode("|", $valores[$cont1]);
797         $numero=count ($separacion);
798         //////////////////////////////////////
799         //Numero de Parametros en la expresion////
800         //////////////////////////////////////
801         $Vt=0;
802         $Ct=0;
803         for ($k=0;$k<$numero;$k++) { //+++++
804             $vect=$separacion[$k];
805             //asignar el signo*****
806             if (strstr($vect,"-")) {
807                 $signo="menos";
808             }
809             else{
810                 $signo="mas";
811             }
812             for ($l=0;$l<$renglon;$l++) { //*****
813                 $buscar1=$buscar.(string) ($l+1);
814                 if (strstr ($vect, $buscar1)) {
815                     //////////////////////////////////////
816                     switch ($datos) { //switch=====
817                         case "Corriente":
818                             if ($signo=="mas") {
819                                 $Ct=$Ct+$current[$l];
820                             }
821                             else{
822                                 $Ct=$Ct-$current[$l];
823                             }
824                             break;
825                         case "Voltaje":
826                             if ($signo=="mas") {
827                                 $Vt=$Vt+$V[$l];
828                             }
829                             else{
830

```

```
830
831
832     $Vt=$Vt-$V[$1];
833     }
834     break;
835     default:
836     break;
837     } //switch=====
838     break;
839     ////////////////
840     }
841     } //*****
842     } //+++++++
843     $cont1++;
844     $Vs[$i]=$Vt;
845     $Is[$i]=$Ct;
846 } //for 1*****
847 // Desplegar Vector////////////////////////////////////
848 if ($show){
849 echo "Vector de ".$datos."<br>";
850 echo "<table width='200' border='1'>";
851 for ($i=0;$i<$columna;$i++) {
852     switch($datos){
853     case "Corriente":
854         echo "<tr><td align='center'>".$Is[$i]."</td></tr>";
855         break;
856     case "Voltaje":
857         echo "<tr><td align='center'>".$Vs[$i]."</td></tr>";
858         break;
859     default:
860         break;
861     }
862 }
863 }
864 echo "</table>";
865 }
866 // Desplegar Vector////////////////////////////////////
867
868 switch($datos){
869     case "Corriente":
870         return $Is;
871         break;
872     case "Voltaje":
873         return $Vs;
874         break;
875     default:
876         break;
877 }
878 }
879 }
```

Tabla 4.35.

```

882 function Jacobi($indicei, $indicej, $Vector, $Matriz) { //*****
883 //Se asignan valores al vector inicial
884 for ($i=0; $i<$indicei; i++) {
885 $Xo[$i]=1;
886 }
887 //Numero de iteraciones////////
888 for ($k=0; $k<110; $k++) {
889
890 for ($indice=0; $indice<$indicei; $indice++) { //////////
891 for ($i=0; $i<$indicei; i++) {
892 $Xtot[$i]=0;
893 }
894 for ($i=0; $i<$indicej; $i++) {
895 if ($indice!=$i) {
896 $Xtot[$indice]=$Xtot[$indice]+$Xo[$i]*$Matriz[$indice][$i];
897 }
898 }
899 $X[$indice]=($Vector[$indice]-$Xtot[$indice])/$Matriz[$indice][$indice];
900 } //////////
901 //Actualizacion del Vector Inicial
902 for ($i=0; $i<$indicei; $i++) {
903 $Xo[$i]=$X[$i];
904 }
905 //echo "<br>";
906 //Actualizacion del Vector Inicial
907 }
908 //Numero de iteraciones////////
909 return $Xo;
910 }
911

```

Tabla 4.36.

```

913
914 function RespuestaDinamica($correcta, $opcion, $cadena, $reactivo,
915 $reporte) {
916 $opcion1=$opcion[0]; $opcion2=$opcion[1]; $opcion3=$opcion[2];
917 $opcion4=$opcion[3];
918 $correcta++;
919 $sql = "UPDATE respuestasdinam SET
920 opcion1='$opcion1', opcion2='$opcion2',
921 opcion3='$opcion3', opcion4='$opcion4',
922 opcion_correcta='$correcta',
923 cadena_generadora='$cadena'
924 WHERE reportes_idreportes='$reporte'
925 AND reactivos_idreactivos='$reactivo'";
926 mysql_query($sql);
927 }
928

```

Tabla 4.37.

```

1110
1111 function div1($primero,$segundo) //<=====
1112 {
1113 if (($segundo['re']==0 && $segundo['im']==0))
1114 {
1115     return (-1);
1116 }
1117 $a=$primero['re'];
1118 $b=$primero['im'];
1119 $c=$segundo['re'];
1120 $d=$segundo['im'];
1121 $e=($a*$c)+($b*$d);
1122 $f=($c*$c)+($d*$d);
1123 $g=($b*$c)-($a*$d);
1124 $result['re']=$e/$f;
1125 $result['im']=$g/$f;
1126 return $result;
1127 }
1128 }
1129
930
931 function DespliegueRadio ($radio, $opcion){
932 $index=1;
933 echo "<br>";
934 for ($i=0;$i<4;$i++){
935     if ($i==0){
936         echo "<INPUT TYPE='RADIO' NAME='$radio' VALUE='$index' >
937 |         $opcion[$i]<br> ";
938     }
939     else
940     {
941         echo "<INPUT TYPE='RADIO' NAME='$radio' VALUE='$index' >
942 $opcion[$i]<br> ";
943     }
944     //echo $opcion[$i]."<br>";
945     $index++;
946 }
947 }
948

```

Tabla 4.38.

```

950
951 function PreguntaAvanzado($idreporte){
952 $sql="SELECT r.enunciado,r.imagen, d.opcion1,d.opcion2, d.opcion3,
953 d.opcion4,d.opcion_elegida, d.cadena_generadora, d.puntaje
954 FROM reactivos AS r
955 JOIN respuestasdinam AS d
956 ON r.idreactivos=d.reactivos_idreactivos
957 WHERE d.reportes_idreportes = '$idreporte'";
958 $consulta=mysql_query($sql);
959 $index=1;
960 $califTotal=0;
961 while($row=mysql_fetch_row($consulta)){
962     $numero = $index;
963 //Arreglo que contiene las 4 respuestas del reactivo
964 $respuesta[0]=$row[2];
965 $respuesta[1]=$row[3];
966 $respuesta[2]=$row[4];
967 $respuesta[3]=$row[5];
968 $elegida=$row[6]-1;
969 echo "<b>Pregunta $index:</b>";echo $row[0]."<br>";
970 echo "<br><img src='upload/$row[1]' height= '300'><br>";
971 echo $row[7]."<br>";
972 $radio="respuesta".$numero;
973 DespliegueRadio ($radio,$respuesta);
974
975     echo "<hr>";
976     $index++;
977 }
978 }
979

```

Tabla 4.39.

```

981
982 function AlumnoRespuestas($respuestas, $idreporte){
983 $ssql="SELECT reactivos_idreactivos,opcion_correcta
984 FROM respuestasdinam
985 WHERE reportes_idreportes='$idreporte'";
986 $consulta=mysql_query($ssql);
987 $indice=0;
988 $scalif=10/3;
989 $scalifTotal=0;
990 while($row=mysql_fetch_row($consulta)){
991 $reactivo=$row[0];
992 $opcion=$row[1];
993 $respuesta=$respuestas[$indice];
994 if($opcion==$respuesta){
995 $scalifTotal=$scalifTotal+$scalif;
996 $puntaje=$scalif;
997 }else
998 {
999 $puntaje=0;
1000 }
1001 $ssql1="UPDATE respuestasdinam
1002 SET opcion_elegida='$respuesta',
1003 puntaje='$puntaje'
1004 WHERE reportes_idreportes='$idreporte'
1005 AND reactivos_idreactivos='$reactivo'";
1006 $consultal=mysql_query($ssql1);
1007 $indice++;
1008 }
1009 $ssql1="UPDATE reportes
1010 SET calificacion='$scalifTotal'
1011 WHERE idreportes='$idreporte'";
1012 mysql_query($ssql1);
1013 }

```

Tabla 4.40.

```

1018
1019 function ResultadoReporte($idreporte){
1020 $ssql="SELECT r.enunciado,r.imagen, d.opcion1,d.opcion2, d.opcion3,d.opcion4,
1021 d.opcion_elegida, d.cadena_generadora, d.puntaje
1022 FROM reactivos AS r
1023 JOIN respuestasdinam AS d
1024 ON r.idreactivos=d.reactivos_idreactivos
1025 WHERE d.reportes_idreportes = '$idreporte'";
1026 $consulta=mysql_query($ssql);
1027 $index=1;
1028 $scalifTotal=0;
1029
1030 echo "<b>Reporte $idreporte Examen Avanzado <br><br></b>";
1031 while($row=mysql_fetch_row($consulta){
1032 $elegida=$row[6]-1;
1033 echo "<b>Pregunta $index:</b><b>Puntaje:</b>".(round($row[8],2)).".<br>";
1034 echo "<hr>";
1035 $scalifTotal=$scalifTotal+$row[8];
1036 $index++;
1037 }
1038 echo "<b>Calificacion : </b>".(round($scalifTotal,2));
1039 }
1040

```

Tabla 4.44

```

1043 function JacobiC($indicei, $indicej, $Vector, $Matriz) { //*****
1044
1045 for ($i=0; $i<$indicei; $i++) {
1046     $Xo[$i]=array('re'=>1, 'im'=>1); //<=====asignacion
1047 }
1048 //Numero de iteraciones//////////
1049 for ($k=0; $k<110; $k++) {
1050     for ($indice=0; $indice<$indicei; $indice++) { ////////////
1051         for ($m=0; $m<$indicei; $m++) {
1052             $Xtot[$m]=array('re'=>0, 'im'=>0); //<=====asignacion
1053         }
1054         for ($i=0; $i<$indicej; $i++) {
1055             if ($indice!=$i) {
1056                 //Multiplicacion ----- Suma
1057                 $producto=mult1($Xo[$i], $Matriz[$indice][$i]); //<====operacion
1058                 $Xtot[$indice]=sumal($Xtot[$indice], $producto); //<====operacion
1059             }
1060         }
1061         //Resta ----- Division
1062         $subtraccion=restal($Vector[$indice], $Xtot[$indice]); //<====operacion
1063         $X[$indice]=div1($subtraccion, $Matriz[$indice][$indice]);
1064         //<=====operacion
1065     } ////////////
1066     //Actualizacion del Vector Inicial
1067     for ($i=0; $i<$indicei; $i++) {
1068         $Xo[$i]=$X[$i];
1069     }
1070     //Actualizacion del Vector Inicial
1071 }
1072 //Numero de iteraciones//////////
1073 return $Xo;
1074 }
1075

```

Tabla 4.45.

```

1078
1079 function sumal($primero, $segundo) //<=====
1080 {
1081     $result['re']=$primero['re']+$segundo['re'];
1082     $result['im']=$primero['im']+$segundo['im'];
1083 }
1084 return $result;
1085 }
1086

```

Tabla 4.46

```

1088
1089 function restal($primero, $segundo) //<=====
1090 {
1091     $result['re']=$primero['re']-$segundo['re'];
1092     $result['im']=$primero['im']-$segundo['im'];
1093     return $result;
1094 }
1095

```

Tabla 4.47

```
1097
1098 function mult1($primero,$segundo) //<=====
1099 {
1100 $result['re']=($primero['re']*$segundo['re'])
1101 -($primero['im']*$segundo['im']);
1102 $result['im']=($primero['re']*$segundo['im'])
1103 +($segundo['re']*$primero['im']);
1104 return $result;
1105
1106 }
1107
1108
```


9. ANEXO C

9.1 MÉTODO DE JACOBI

Método Iterativo para la solución de sistemas de ecuaciones lineales de la forma $Ax = b$, se tiene un sistema de ecuaciones de la siguiente forma, como se muestra en la Figura C.1.:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1i}x_i + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2i}x_i + \dots + a_{2n}x_n &= b_2 \\ \cdot & \\ \cdot & \\ \cdot & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{ni}x_i + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Figura C.1.

se resuelve la X correspondiente a cada renglón.

$$\begin{aligned} x_1 &= \frac{1}{a_{11}} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1i}x_i - \dots - a_{1n}x_n) \\ x_2 &= \frac{1}{a_{22}} (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2i}x_i - \dots - a_{2n}x_n) \\ \cdot & \\ \cdot & \\ \cdot & \\ x_n &= \frac{1}{a_{nn}} (b_n - a_{n1}x_1 - a_{n2}x_2 - a_{n3}x_3 - \dots - a_{ni}x_i - \dots - a_{nn}x_n) \end{aligned}$$

Figura C.2.

se cuenta con un vector inicial, el cual propondrá una solución inicial, que va a iniciar el proceso iterativo:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Figura C.3.

Con este vector se obtienen los valores iniciales, los cuales se sustituyen en la misma ecuación de solución, para ir iterando hasta que el algoritmo converge a la solución única.

Para que el algoritmo converja se necesita una matriz diagonalmente dominante, la cual debe cumplir lo siguiente:

$$|a_{ij}| = \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|$$

Figura C.4.

La siguiente es la función que se implementó para resolver el sistema de ecuaciones por el método iterativo.

Los parámetros de entrada son:

- \$índicei, \$índicej, representan el tamaño de la matriz nxn.
- \$Vector: valores del vector corriente/voltaje.
- \$Matriz: valores de la matriz Admitancia/Voltaje.

Pasos:

1. Se inicializa un vector inicial de tamaño n, \$X₀, el cual va a depender del tamaño de la matriz. A cada uno de los elementos de este vector se le asigna un valor arbitrario, para este caso 1.
2. Se cuenta con un primer ciclo iterativo for, el cual representa el número de iteraciones que se van a realizar para obtener la solución.

3. Para obtener el primer vector de soluciones, se toman los valores de este vector inicial, y se resuelve el sistema según la Figura C.2.
4. Se cuentan con dos ciclos iterativos para recorrer los elementos de la matriz, el primero recorre los renglones de la matriz y el segundo las columnas.
5. Se cuenta con un vector $\$X_{tot}$, el cual va a contener las sumatorias de la multiplicación del elemento de la matriz por el elemento del vector (Ver Figura C.5). El ciclo iterativo que recorre las columnas es donde se realiza esta sumatoria, en la matriz recorre las columnas y en el vector recorre los elementos, realizando la multiplicación de ambos elementos, siempre y cuando el elemento de la matriz no pertenezca a la diagonal principal, para posteriormente hacer una sumatoria con el valor que se cuenta en la variable $\$X_{tot}$.
6. Al terminar de hacer el recorrido, de las columnas se realiza una división del contenido de la variable $\$X_{tot}[n]$ entre el elemento de la diagonal principal de ese renglón, para posteriormente restárselo al elemento del vector corriente/voltaje que corresponda.
7. Se continua con el siguiente renglón, inicializando en cero todos los elementos del vector $\$X_{tot}$, para volver a hacer el recorrido por las columnas. Se repiten pasos 5 y 6 hasta llegar al último renglón.
8. Al obtener la solución del sistema contenida en el vector $\$X$, se sustituyen estos valores en el vector inicial $\$X_0$. Se repiten pasos del 3 al 7 hasta hacer todas las iteraciones.

10. ANEXO D

10.1 HOJAS DE ESTILO EN CASCADA

CSS (Cascading Style Sheets - Hojas de Estilo en Cascada), es un lenguaje desarrollado por la *WWW Corporation*, con el fin de separar la parte gráfica de la estructura del HTML, funciona aplicando reglas de estilo a los elementos HTML, entre las que se pueden incluir tamaño, color de fondo, color del texto, posición de los elementos, márgenes, tipos y tamaños de letra, entre otros. HTML se usa para estructurar el contenido; CSS se usa para dar formato al contenido previamente estructurado, la separación entre la presentación de los documentos y del contenido de los mismos, hace que el mantenimiento del sitio sea más sencillo.

Utilizar CSS proporciona un mejor flujo de trabajo, mayor organización del código, menos peso en las páginas, y más flexibilidad a los cambios.

Para utilizar CSS debemos tomar en cuenta tres principales elementos (ver figura D.1):

Atributos o Propiedades: Son las palabras que se utilizan para indicar el estilo a modificar.

Valores: Son para definir cómo se modifica el atributo o la propiedad.

Selectores: Se usan para definir sobre cuales elementos HTML se aplican los estilos, si se quiere definir un estilo para toda la pagina, se usa el selector "*body*"

que se refiere a la etiqueta `<body>` del documento HTML; hay tres tipos de selectores:

Los selectores de etiquetas HTML: se utilizan escribiendo el nombre de la etiqueta a la que se le aplica el estilo.

Los selectores de identificador: se usan para aplicar estilos solo a las etiquetas identificadas con un nombre.

El selector de clase: se escribe en el documento CSS comenzando con un punto "." seguido del nombre que se le quiera poner a la clase.

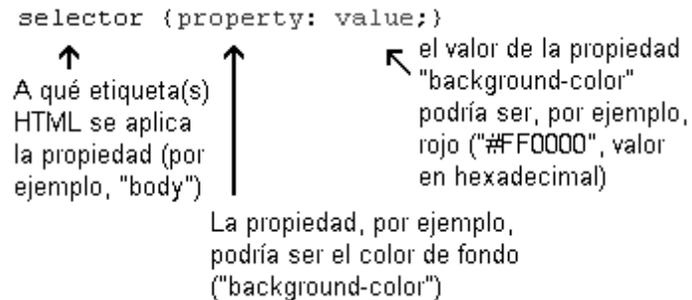


Figura D.1. Modelo CSS fundamental.

El modelo de caja en CSS describe las cajas o espacios que se generan a partir de los elementos HTML.

El modelo de caja también contiene opciones detalladas en lo referente al ajuste de márgenes, bordes, relleno (*padding*) y contenido de cada elemento.

La siguiente imagen muestra cómo se construye el modelo de caja. (ver figura D.2):

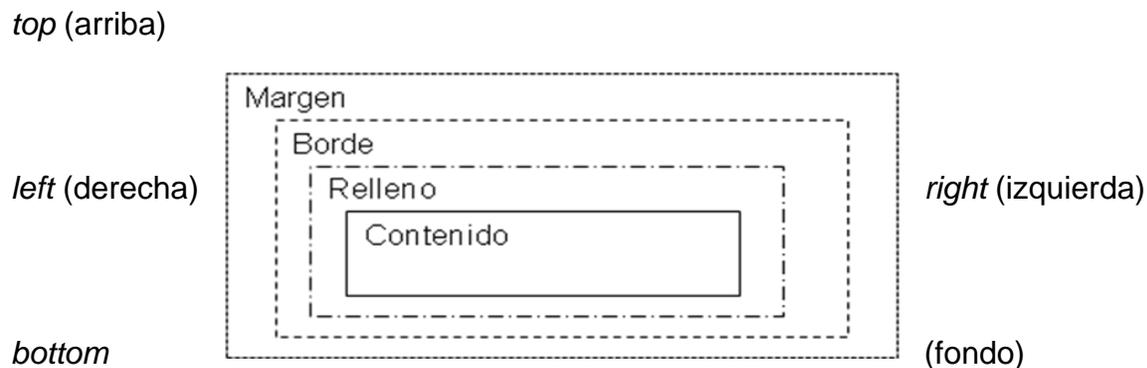


Figura D.2. Modelo de caja de CSS.

El modelo de caja del SIEL se muestra en la siguiente figura, en donde se pueden ver los diferentes elementos que componen el CSS, como son:

Caja: Espacio en donde se colocarán los demás elementos.

Encabezado: Que contiene la imagen del SIEL y la caja Encabezado-links.

Encabezado-links: Tiene los vínculos de Inicio, Admin y Contacto, los cuales llevan a las páginas correspondientes y son dinámicos de acuerdo al usuario.

Menú: Contiene los vínculos Profesor, Alumno y Contacto, también tiene la caja Fecha.

Fecha: Proporciona la fecha actual.

Contenido-caja: Tiene las cajas Barra-lateral y Principal.

Barra-lateral: Tiene vínculos hacia los diferentes usuarios y otros sitios de interés.

Principal: En esta caja se observarán las diferentes pantallas, de acuerdo a cada usuario, así como el examen.

Pie: Muestra el pie de página del SIEL. (ver figura D.3).



Figura D.3. Modelo de caja de CSS del SIEL.

La sintaxis de CSS es sencilla, primero se coloca el selector, luego se abre una llave "{" y se empiezan a colocar los atributos, seguidos de dos puntos ":" y luego el valor seguido de punto y coma ";", al final se cierra el estilo para el selector con el cierre de llave "}".

Se pueden definir tantos atributos con sus respectivos valores como se desee, separándolos con un espacio o un salto de línea.

En CSS se escriben los atributos y valores con minúsculas y los comentarios se encierran con `"/*"` para abrir y `"*/"` para cerrar.

Dentro del código de CSS se enuncian todos los elementos que se utilizarán en el código HTML, como *body* (cuerpo), títulos, vínculos, imágenes, tablas, botones, después se dan los atributos o propiedades a cada elemento, también se especifica la posición de las cajas y sus respectivas características.

Se define el estilo de los botones para realizar alguna acción, en este caso, buscar. En el código se inicia con el selector `"input.button"` seguido de la llave que abre `"{"`, después se enumeran los diferentes atributos correspondientes a la fuente: su tipo, tamaño y color `"font: bold 12px Arial, Sans-serif;"`, `"color: #FFF;"`, el alto del botón `"height: 24px;"`, su posición `"margin: 0;"`, relleno `"padding: 2px 3px;"`, imagen de fondo `"background: #d37e20 url(fondoBoton.png) repeat-x 0 0;"` y borde `"border: none;"`. (Ver figura D.4).



Figura D.4. Estilo de los botones del SIEL

11. ANEXO E

11.1 INSTALACIÓN DEL SERVIDOR

Para fines de este documento se utiliza un servidor dedicado sólo al propósito del SIEL, a continuación se describe de manera resumida la puesta en marcha del servidor con un sistema Linux, distribución Ubuntu para ser más precisos, utilizando la versión server 8.04.

Antes de iniciar con la instalación de un servidor web hay varios rubros que se deben tomar en cuenta: hardware, sistema operativo, configuración de la red, servicios que se van a ofrecer, seguridad, etc.

1. HARDWARE.

Se requiere un equipo que sea capaz de satisfacer nuestras necesidades y las de los clientes, en el caso de la elaboración de este sistema, el servidor cuenta con las siguientes características:

- Procesador Intel(R) Xeon(TM) CPU 3.00GHz
- Memoria RAM de 1 Gigabyte
- Disco duro SCSI 70 Gigabytes
- Unidad CD-ROM 40X
- Tarjeta Red Ethernet 10Base-T/100Base-TX RJ45

2. SERVICIOS

Los servicios con los que debe contar el servidor son los siguientes:

- Web
- Soporte PHP
- Soporte MySQL
- Soporte SSH
- Servidor de correo*

*Opcional

3. SISTEMA OPERATIVO.

Ubuntu es una de las más importantes distribuciones de GNU/Linux a nivel mundial; se basa en Debian GNU/Linux y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares, durante su instalación se pueden activar varios servicios para que el servidor tenga todo lo que se requiera para montar de un PC normal, un servidor que ofrezca servicios a internet.

Para iniciar la instalación es necesario descargar el disco de instalación de la página oficial de Ubuntu, como anteriormente se mencionó, se requiere descargar la versión servidor (*server*). Puede obtener la versión descargable desde el siguiente vínculo: <http://www.ubuntu.com/getubuntu>.

(Ver figura E.1 y figura E. 2).



Figura E.1. Página de Ubuntu.



Figura E.2. Descarga de Ubuntu server.

El archivo a descargar tiene formato *.iso, archivo imagen del disco de instalación, es necesario grabar dicho archivo a un disco compacto para su posterior uso. Cualquier programa de grabación de CDs con soporte ISO sirve para lograrlo. Con el disco de instalación grabado, deberá arrancarse con él la computadora que será el servidor, es necesario asegurarse en el BIOS del equipo que puede arrancarse desde una unidad de CD.

Al arrancar la computadora con el disco de instalación de Ubuntu, se muestra un menú, donde se elegirá el idioma de la instalación y posteriormente se elige la opción Instalar Ubuntu Server (ver figura E.3).

Hay que elegir el país en donde se encuentra el servidor, con lo que queda implícita la elección del idioma del sistema; se debe escribir el nombre que tendrá la computadora y se elige el huso horario, para configurar el reloj del sistema.



Figura E.3. Pantalla de instalación.

Es necesario particionar el disco duro de la computadora para cumplir con la estructura que requiere Ubuntu Linux, si la computadora sólo se utilizará como servidor, entonces se elige la opción Guiado (ver figura E.4) – Utilizar todo el disco duro, que es la opción más sencilla, si no es así entonces es necesario que se revise la documentación en la página de ayuda de Ubuntu.

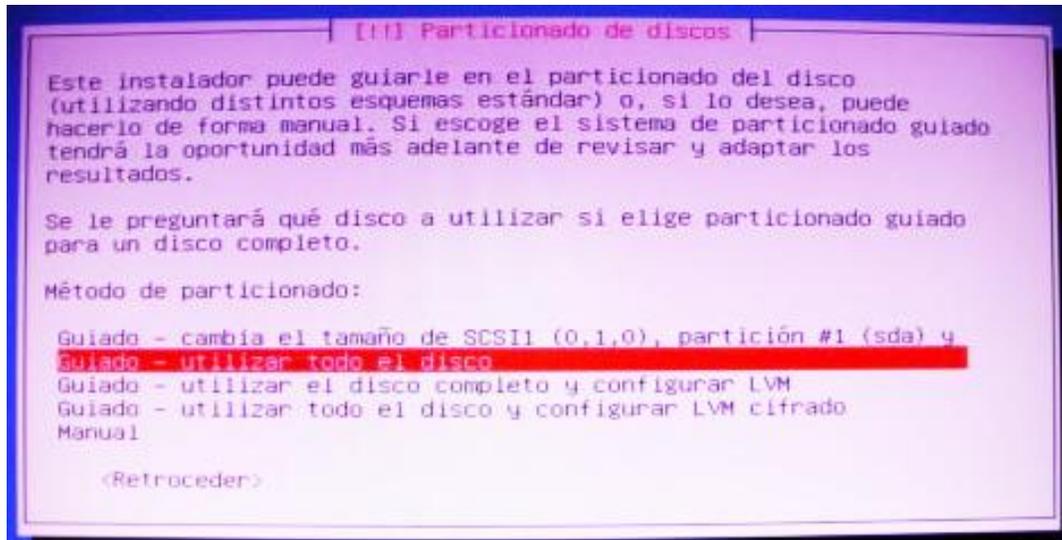


Figura E.4. Particionado

Después de elegir el método de particionado, se debe elegir el disco a particionar y si todo es correcto entonces se deben confirmar que los cambios al disco duro son los deseados y por lo tanto se llevarán a cabo.

Después se debe introducir el nombre del usuario del sistema y el nombre de usuario de la cuenta con su contraseña.

En la siguiente pantalla se deben elegir los servicios con los que debe contar el servidor (ver figura E.5), se selecciona las opciones **LAMP server** y **OpenSSH server**.

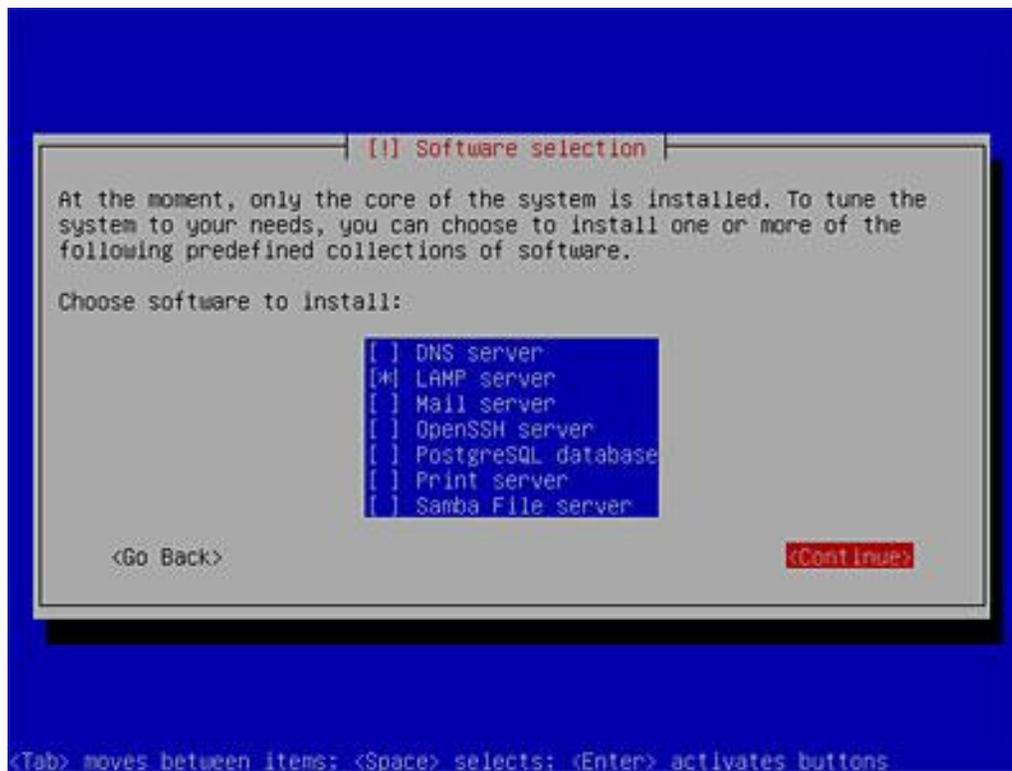


Figura E.5. Servicios

Después se debe ingresar la contraseña del usuario **root** para MySQL, que es nuestro administrador de Base de Datos (ver figura E.6).

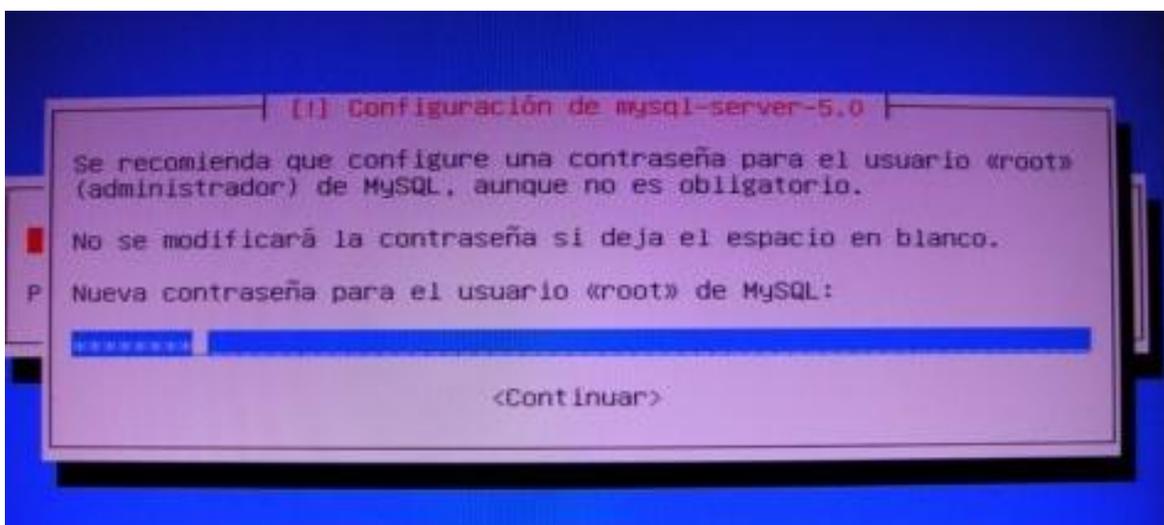


Figura E.6. Contraseña root

Una vez que termine la instalación; el sistema expulsa el disco de instalación, se presiona el botón continuar y el sistema se reiniciará.

Para ingresar al sistema es necesario introducir el nombre de usuario y contraseña.

4. CONFIGURACIÓN DE LA RED

Es necesario que nuestra tarjeta de red maneje una IP estática, en la instalación del sistema se configuró la IP de manera dinámica, se requiere cambiar dicha opción. Al hacerlo se permite que los usuarios externos a la red local puedan visualizar el servidor.

Para cambiar la configuración de la IP, ingresamos al sistema el nombre de usuario y contraseña por medio de una terminal se introduce el siguiente comando:

```
sudo nano /etc/network/interfaces
```

Lo anterior abre un editor de texto con el archivo **interfaces** (ver figura E.7).



```
GNU nano 2.0.7 File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

[ Read 10 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Figura E.7. Interfaces

En la imagen se muestra que la tarjeta de red primaria, está configurada para que se asigne una IP automáticamente vía DHCP, para empezar se debe cambiar esa palabra por la palabra **static** y a continuación se especifica la IP fija, máscara de subred, puerta de enlace o gateway y resolvidor de nombres DNS.

Para el fin de este documento se ha configurado la IP **132.248.59.112**, se pueden ver las otras opciones en la figura E.8.

```

GNU nano 2.0.7      File: /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 132.248.59.112
    netmask 255.255.255.0
    network 132.248.59.0
    #broadcast 132.248.59.255
    gateway 132.248.59.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 132.248.10.2 132.248.1.3
    dns-search fi-b.unam.mx

^G Get Help   ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
    
```

Figura E.8 Configuración de red

Una vez hechas todas las modificaciones al archivo (**interfaces**), se guarda presionando la tecla **F2**, después la tecla **Y** y después **Enter**, a continuación se debe reiniciar la red para que los cambios surtan efecto.

Para reiniciar la red se introduce el siguiente comando:

sudo /etc/init.d/networking restart

```

root@allen:/# /etc/init.d/networking restart
 * Reconfiguring network interfaces... [ OK ]
root@allen:/# █
    
```

Para comprobar que se han efectuado los cambios en la configuración, se introduce el comando **ifconfig**,

```
root@allen:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:14:22:75:69:c0
          inet addr:132.248.59.112  Bcast:132.248.59.255  Mask:255.255.255.0
          inet6 addr: fe80::214:22ff:fe75:69c0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:838032 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4794 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:60819558 (58.0 MB)  TX bytes:1942921 (1.8 MB)
          Base address:0xdcc0 Memory:fe7e0000-fe800000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:229 errors:0 dropped:0 overruns:0 frame:0
          TX packets:229 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15370 (15.0 KB)  TX bytes:15370 (15.0 KB)

root@allen:/# █
```

con lo que se muestra que la configuración de la IP ha quedado guardada.

5. SERVIDOR LAMP

Si hay algo que caracterice a **Linux Ubuntu** de otras distribuciones **Linux** es por su facilidad de manejo, al seleccionar (**LAMP Server**) durante la instalación se ha instalado en el equipo un conjunto de aplicaciones para poder usarla como un servidor listo para poder ofrecer contenido Web en nuestra red de área local o hacia internet.

La opción **LAMP Server** instala automáticamente **Apache HTTP Server**, **MySQL** y soporte para **PHP**, necesario para los fines de este documento (albergar y ejecutar los scripts del SIEL).

Una vez efectuada la instalación de **LAMP Server** se debe configurar algunas cosas para que todo funcione como se requiere y para tener control sobre ello.

Antes de nada se debe comprobar que el **Servidor Web Apache** y el soporte para **PHP** funcionan bien, para lo cual basta con abrir un navegador e introducir la IP que tiene el servidor. (<http://132.248.59.112>).

Si se muestra la siguiente pantalla:



It works!

quiere decir que el **Servidor Web Apache** funciona correctamente, para comprobar que hay soporte **PHP**, sólo es necesario cambiar el archivo index.html que muestra la leyenda **It works!** por un archivo PHP.

6. SEGURIDAD

Para evitar ataques al servidor, es importante instalar alguna aplicación que provea de la funcionalidad de restringir los accesos que resulten sospechosos, bloqueando a todo aquello que tenga un número de intentos fallidos de *login* a alguno de los servicios del servidor.

La aplicación que se utiliza para proteger al servidor se llama **Fail2Ban** y se va a encargar de cuidar las 24 horas del día.

Para instalarlo se ejecuta en una terminal el comando:

```
apt-get install fail2ban
```

Una vez instalado, se debe configurar, para ello usamos el siguiente comando:

```
nano /etc/fail2ban/jail.conf
```

Las opciones que se deben cambiar se especifican en la siguiente tabla.

Opción	Descripción
ignoreip :	IPs del área local que ignorará en caso de equivocación.
bantime :	Tiempo que el usuario que ha fallado en el <i>login</i> se quedará sin poder acceder al servicio. Especificado en segundos.
maxretry :	Numero de intentos de <i>login</i> .
backend :	Permite de un modo u otro la escritura de ficheros <i>log</i> .
destemail :	Dirección de correo donde se envían las alertas.
action :	Forma en que iptables aplica sus reglas.

12.MANTENIMIENTO Y ACTUALIZACIÓN

1. ARCHIVOS PRIMORDIALES

En esta sección se mencionan los archivos importantes y su ubicación, que cuenta con información que es guardada en forma de constantes que el sistema utiliza y que en lugar de estar buscando en las páginas que requieren, esta información sólo se busca en los archivos que a continuación se van a mencionar. De manera que si se quieren hacer cambios se deben ubicar estos archivos.

- **configuracion.php, miTesis.php:** archivos que contiene información de las rutas absolutas que usa el sistema para desplegar información así como variables de autenticación con la Base de Datos MySQL.
- **Funciones.php, matriz1.php, Jacobi.php:** archivos que contienen las funciones que se usan para llevar a cabo todas las funcionalidades del sistema.
- **Directorio images:** directorio donde se localiza la hoja de estilo que fue diseñada para el sistema así las diferentes imágenes que se usan.
- **MySQL:** se mencionan la dirección donde se puede acceder a la Base de Datos así como el *password* del usuario *root*, y los diferentes usuarios creados para tener acceso a esta.

La información relacionada con las rutas de directorios raíz y donde se guardan los archivos que se suben al servidor, así como la autenticación para acceder a la

Base de Datos; se guardan en constantes, las cuales se encuentran en dos archivos principales, `configuracion.php` y `miTesis.php`.

En el archivo `configuracion.php` se tienen constantes las cuales contienen información referente al directorio base, la ruta del directorio donde se guardan los archivos que se suben al servidor, el cual se encuentra en el directorio raíz.

- `$config_directorioBase`: ruta de acceso del directorio que contiene los archivos del sistema.
- `$config_upload`: ruta de acceso para guardar los archivos que se suben al servidor.

La información que sirve como autenticación, para acceder a la Base de Datos de MySQL, se encuentra en el archivo `miTesis.php`, se encuentra en las constantes:

- `$hostname_miTesis`: nombre del servidor donde se encuentra almacenado la Base de Datos.
- `$database_miTesis`: nombre de la Base de Datos, que contiene la información del trabajo de tesis.
- `$username_miTesis`: nombre de usuario válido para entrar en la Base de Datos.
- `$password_miTesis` : *password* del usuario.
- `$miTesis` : variable que contiene la función con los parámetros necesarios para tener acceso a la Base de Datos.

La ubicación del archivo se encuentra en el directorio `Connections`. Las funciones que realizan las diferentes tareas de los módulos del sistema se encuentran en el archivo `Funciones.php`. El *parser* (interpretador) que realiza la conversión de la cadena de texto; que contiene la información de las ecuaciones del circuito, a la matriz con los valores se encuentra en el archivo `Matriz1.php`. La solución del sistema de ecuaciones por el método de Jacobi se encuentra en el archivo `Jacobi.php`.

2. ARCHIVOS DE EXÁMENES Y REPORTES DE EXÁMENES:

- **Pregunta1_SIEL.php** : página de examen básico.
- **ExamAvanzado.php**: página de examen avanzado.
- **Reporte.php**: página de reporte de examen básico.
- **ReporteDinam.php**: página de reporte de examen avanzado.

3. HOJA DE ESTILO

La ubicación de la hoja de estilo se encuentra en el directorio images, así como todas las imágenes que se usaron para el diseño de la página.

4. BASE DE DATOS MySQL

El administrador de la Base de Datos, se localiza en la siguiente dirección:

<http://132.248.59.112/phpmyadmin/>

La Base de Dato usada es *bdsiel*.

Usuario: xxxxxx

Password: xxxxxx

13. GLOSARIO DE TERMINOS

A

ACTORES: Entidades dentro del sistema, las cuales permiten que éste funcione llevando a cabo tareas específicas.

ALGORITMO: Conjunto de operaciones ordenadas que se deben realizar para resolver un problema.

ANSI: (*American National Standards Institute*). Instituto Nacional Americano de Estándares.

APACHE: Servidor de páginas web de código abierto.

ASCII: (*American Standard Code for Information Interchange*). Código Estadounidense Estándar para el Intercambio de Información.

B

BASE DE DATOS: Base de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite, es un sistema de archivos electrónico.

BUGS: Error de Software.

C

CBT: (*Computed Based Training*) Capacitación Basada en Computadoras.

CIRCUITO ELÉCTRICO: Serie de elementos o componentes eléctricos o electrónicos, conectados eléctricamente entre sí con el propósito de generar, transportar o modificar señales electrónicas o eléctricas.

CSS: (*Cascading Style Sheets*) Hojas de Estilo en Cascada.

D

DISEÑO: En el ciclo de vida de un programa, una vez que los requisitos de un programa han sido establecidos en la fase de análisis, ya se puede iniciar la fase

de diseño. En esta etapa se tiene que encontrar una solución informática al problema planteado. Dicha solución determinará cómo se va a resolver el problema.

E

E-LEARNING: Sistemas de Enseñanza y Evaluación en Línea.

E-MAIL: Correo Electrónico.

ENTIDAD: Está compuesta de atributos que son los datos que definen el objeto.

ERROR: Es una tarea humana que provoca que un software contenga una falla.

F

FALLA: (*Failure*) Error de Software. Sucede cuando un programa no se comporta de manera apropiada.

FALTA: (*Fault*) Se lleva a cabo en el código del programa. La existencia de una falta en el programa puede ocasionar una falla (*failure*) en el sistema.

FILEZILLA: Es un cliente FTP, gratuito, libre y de código abierto.

FTP: (*File Transfer Protocol*). Protocolo de Transferencia de Archivos.

H

HTML: (*HyperText Markup Language*). Lenguaje de Marcas de Hipertexto.

HTTP: (*Hypertext Transfer Protocol*). Protocolo de Transferencia de Hipertexto.

HTTPd: (*HTTP daemon*). Programa que corre de fondo en un servidor Web y espera peticiones de entrada para responderles.

I

IBM: (*International Business Machines*). Maquinas de Negocio Internacional.

IMPLEMENTACIÓN: Formas y métodos para llevar a cabo algo, etapa donde efectivamente se programa el sistema.

INTERFAZ: Sistema que permite a los usuarios comunicarse e interactuar con la computadora y el software que ésta tiene instalado. A su vez, este software

(programas) se comunica mediante un sistema de interfaz con el hardware de la computadora.

INTERNET: Conjunto descentralizado de redes de comunicación interconectadas de alcance Mundial.

L

LINUX: Es el núcleo del sistema operativo libre denominado GNU/Linux

LMS: (*Learning Management System*). Sistemas de Administración de Aprendizaje.

M

MANUALES DE OPERACIÓN: Define la organización, los mecanismos y los procedimientos operativos a utilizar en las actividades del sistema que se refiere.

MacOS: Sistema Operativo de Macintosh.

MODELO DE PROCESO UNIFICADO: Marco de desarrollo de *software* que se caracteriza por estar dirigido por Casos de Uso, centrado en la arquitectura y por ser iterativo e incremental.

MODELO ENTIDAD – RELACIÓN: Herramienta de modelado que permite representar de manera grafica la estructura lógica de una base de datos.

MULTIPLATAFORMA: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación u otra clase de software, que puedan funcionar en diversos sistemas operativos o plataformas.

MYSQL: (*My Structured Query lenguaje*). Lenguaje de Consulta Estructurado

N

NCSA: (*National Center for Supercomputing Applications*). Centro Nacional para Aplicaciones de Supercomputación.

NULL: En SQL, NULL no es un valor; es un estado que indica que el valor es desconocido o no existente.

P

PASSWORD: Contraseña; serie secreta de caracteres que permite a un usuario tener acceso a un archivo, a una computadora o a un programa.

PHP: (*Hypertext Pre-Processor*). Pre-procesador de Hipertexto.

PHP/FI: (*Hypertext Pre-Processor / Form Interpreter*). Pre-procesador de Hipertexto / Formulario Interprete.

PERFORMANCE: Rendimiento.

PRUEBA: Fase donde se ensaya y experimenta en el desarrollo de un software; a veces se corre algunos riesgos en el uso de estos programas no suficientemente perfeccionados.

R

RDD: (*Responsability-Driven-Design*). Diseño por Responsabilidades.

RELACIÓN: Asociación entre entidades, sin existencia propia en el mundo real, pero necesaria para reflejar las interacciones existentes entre entidades

S

SERVIDOR WEB: Interfaz entre el cliente web que pide una URL y el sistema donde se encuentra ese servicio.

SEL: Sistema de Evaluación en Línea.

SITIO WEB: Conjunto de páginas web que están relacionadas entre sí, por lo general porque se ingresan desde un mismo dominio (o porque mantienen constante la raíz de la dirección URL); un sitio web puede estar constituido de una o más páginas web. La página web principal de un sitio web suele llamarse *index*, que puede tener la extensión .htm, .php, .asp, entre otras.

SOFTWARE: Conjunto de programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

SOFTWARE LIBRE: Software para cuya distribución no se necesita pagar una licencia.

SOLARIS: Es un sistema operativo de tipo Unix desarrollado por Sun Microsystems.

SQL: (*Structured Query Language*). Lenguaje de Consulta Estructurado.

U

URL: (*Uniform Request Locator*). Localizador Uniforme de Recursos.

UPDATE: Actualizar; método y/o utilidad para actualizar una aplicación informática.

USUARIOS: Personas reales que utilizan el sistema.

V

VALIDACIÓN: Verificar, controlar o filtrar, cada una de las entradas de datos que provienen desde el exterior del sistema.

W

WEB: Red Informática.

WINDOWS: Familia de sistemas operativos desarrollados y comercializados por Microsoft.

WWW: (*World Wide Web*). Red Global Mundial.

X

XAMPP: Plataforma X, contiene las herramientas Apache, MySQL, PHP y Perl.

14. BIBLIOGRAFÍA

Beginning PHP5, Apache, and
MySQL® Web Development
Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec,
Jeremy Stolz, Michael K. Glass
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com
Copyright © 2005 by Wiley Publishing, Inc., Indianapolis, Indiana

PHP 5 Recipes
A Problem-Solution Approach
Lee Babin, Nathan A. Good,
Frank M. Kromann, Jon Stephens
PHP 5 Recipes: A Problem-Solution Approach
Copyright © 2005 by Lee Babin, Nathan A. Good, Frank M. Kromann, Jon
Stephens
Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233
Spring Street, 6th Floor,
New York, NY 10013. Phone 1-800-SPRINGER, fax
201-348-4505, e-mail orders-ny@springer-sbm.com, or
visit <http://www.springeronline.com>.

PHP5 a través de ejemplos
Abraham Gutiérrez – Gines Bravo
Alfaomega Ra-Ma
México D.F.
2005

Sams Teach Yourself PHP, MYSQL and APACHE
Julie C. Meloni.

Bibliografía

Creación de un portal con PHP y MySQL

Jacobo Pavón Puertas

3ª Edición

Alfaomega Ra-Ma

Ingeniería de Software Orientada a Objetos con Uml, Java e Internet

Alfredo Weitzenfeld

Thomson

Circuitos Eléctricos

Dorf * Svoboda

6ª Edición

Alfaomega

Referencias Web

<http://www.webestilo.com/php/php00.phtml>

<http://www.maestrosdelweb.com/editorial/phpmysqlap/#a12>

<http://www.alegsa.com.ar/Dic/httpd.php>

http://es.wikipedia.org/wiki/Codigo_HTML

<http://en.wikipedia.org/wiki/Internet>

<http://es.wikipedia.org/wiki/MySQL>

<http://es.wikipedia.org/wiki/ASCII>

<http://www.firebirdsql.org/manual/es/nullguide-es-queesnull.html>

[http://es.wikipedia.org/wiki/Solaris_\(sistema_operativo\)](http://es.wikipedia.org/wiki/Solaris_(sistema_operativo))

<http://www.microsoft.com/Spain/windows/>

<http://es.wikipedia.org/wiki/FileZilla>

<http://www.filezilla-project.org/>

<http://www.sidar.org/recur/desdi/traduc/es/css/box.html>

<http://www.cristalab.com/tutoriales/tutorial-basico-de-css-c94/>

<http://www.biblioweb.dgsca.unam.mx/libros/repositorios/contenido.htm>

Información de PHP consultada en las siguientes direcciones de la RED (www)

<http://www.mysql.com/>

Página oficial del MySQL

<http://dev.mysql.com/doc/>

Documentación Disponible

<http://dev.mysql.com/doc/refman/5.0/es/introduction.html>

Manual de Referencia