

Universidad Nacional Autónoma de Nicaragua - León
Facultad de Ciencias
Departamento de Computación



Tesis para Optar al Título de Ingeniero en Sistemas de Información

Tema:

“Automatización del Registro Académico del Colegio La Salle - León”

Integrantes:

- **Narciso Antonio Bonilla Tórrez.**
- **Deybin Bayardo Chavarría Estrada.**
- **Wilton Alfredo Guardado Castillo.**
- **Douglas Guillermo Molina Castro.**

Tutor: Msc. Martín Ibarra Padilla.

Miércoles 20 de Agosto del 2008

CONTENIDO

1. INTRODUCCIÓN	1
2. ANTECEDENTES.....	2
3. JUSTIFICACIÓN	3
4. OBJETIVOS	4
5. MARCO TEÓRICO	5
Sistemas de Información.....	5
5.1.1 Actividades de un Sistema de Información.....	5
5.1.2 Tipos de Sistemas de información	7
Conceptos Básicos de Base de Datos	7
PHP	8
Origen de PHP.....	8
Características de PHP.....	9
Usos de PHP	10
Cómo funciona PHP	10
MYSQL	11
Usos de MYSQL	11
Características Principales de MYSQL	11
Macromedia Dreamweaver.....	12
JavaScript.....	15
Servidores Web.....	16
Internet Information Server (IIS).....	17

Servidor Apache	18
Arquitectura Cliente/Servidor	19
5.9 Protocolo HTTP.....	20
5.10 Protocolo TCP/IP	24
5.11 Gestión de Calidad de un Producto (Norma ISO - 9001)	25
6. METODOLOGÍA DEL TRABAJO	28
Materiales	28
Métodos	29
Análisis	31
6.3.1 Especificaciones de Requisitos de Software (ERS)	31
Diseños	54
6.4.1 Diagrama de Flujo de Datos (DFD)	54
6.4.2 Diccionario de Datos (DD).....	61
6.4.3 Diagrama Entidad – Relación (ER)	64
6.4.4 Diseño de Datos.....	65
6.4.5 Diseño Arquitectónico (DA)	75
6.4.6 Diseño de Interfaz (DI)	76
Codificación	84
Pruebas.....	116
Mantenimiento.....	117
7. CONCLUSIONES	118
8. RECOMENDACIONES.....	119

9. BIBLIOGRAFÍA	120
10. ANEXOS	121
10.1 Glosario	121
10.2 Constancia de satisfacción del Cliente (Colegio La Salle - León).....	128
10.3 Informes que brinda la Aplicación	129

AGRADECIMIENTO

En este trabajo monográfico de alguna u otra manera nos han ayudado muchas personas, pero en especial queremos agradecer a las que han estado siempre con nosotros y que nos han dado consejos, confianza y apoyo siempre que lo necesitamos, sin ellos no hubiésemos podido alcanzar una de nuestras metas que es culminar esta tesis.

Sin lugar a dudas nuestro primer agradecimiento, es para **Dios**, por darnos la vida, nunca nos ha dejado solos, siempre nos ha iluminado y gracias a él tuvimos la oportunidad de estudiar una carrera.

A lo más hermoso que Dios nos ha dado "**Nuestros Padres**", a ellos que siempre nos han apoyado en todas nuestras decisiones, que siempre nos han dado ánimo y fortaleza con sus ejemplos y palabras, dándonos a entender que con esfuerzo, trabajo y fe nada es imposible, damos gracias a nuestros Padres por darnos lo mejor de ellos y sobre todo por confiar en nosotros.

No queremos dejar de agradecer al **Msc. Martín Ibarra** que ha sido un pilar fuerte en el desarrollo de nuestro trabajo, que con sus consejos, recomendaciones y con el tiempo que nos ha brindado hemos podido presentar nuestra tesis con una excelente calidad.

DEDICATORIA

A DIOS: ser infinito que nos guió, iluminó y nos dió la sabiduría para que pudiésemos terminar con nuestra tesis y que nos permitió vencer los obstáculos puesto en nuestro camino.

A Nuestros Padres: que con su ayuda incondicional nos han dado la oportunidad de tener una educación integral para la formación cívica y moral de cada uno de nosotros.

A NUESTRO TUTOR: Msc. Martín Ibarra, por su paciencia, esfuerzo y voluntad para enseñarnos.



1. INTRODUCCIÓN

El desarrollo de la tecnología ha alcanzado un gran auge en nuestra sociedad, el uso de la computadora se ha convertido en una poderosa herramienta para instituciones, organizaciones, colegios y para nosotros como usuarios.

Haciendo uso de Internet, podemos encontrarnos con sistemas de información para el control de registro académico y que pueden ser descargados para ser usados. El problema con estos sistemas es que no cumplen con todas las expectativas de los Colegios.

Los cambios que ocurren en nuestro entorno, impulsan a las personas a hacer uso de las tecnologías. Vemos estos cambios en los colegios donde el aumento del estudiantado obliga el uso de técnicas informáticas para llevar un control automatizado que resultaría muy engorroso hacerlo de la forma tradicional. Esto se puede ver en el Colegio La Salle – León, quién realiza 1700 matrículas aproximadamente cada año, incluyendo los alumnos de primer ingreso y de los cuales se estima que son 100 cada año; esto es muy difícil para los que laboran en dicho Colegio, puesto que tratar con estos datos no es una tarea fácil.

Por tanto valiéndonos de las herramientas tecnológicas y el conocimiento adquirido durante nuestros 5 años de estudios superiores en Ingeniería de Sistemas informáticos hemos desarrollado una aplicación para la Automatización del Registro Académico del colegio La Salle.



2. ANTECEDENTES

El desarrollo de software de gestión se ha vuelto común en todas las áreas de trabajo, no obstante debido a la gran cantidad de datos que se manejan manualmente, conllevan a errores en su manipulación y a medida que estos volúmenes de datos crecen, presentar informes y consultar dichos datos se vuelve una tarea engorrosa.

La gestión que realizan los colegios para el control del registro académico es compleja, un trabajo que hay que considerar, más aún cuando son pocas las personas que acceden y a su vez manipulan estos datos.

En el año 2006 en el Colegio La Salle se implementó una pequeña aplicación para el control de notas, y que fue desarrollada en Microsoft Access, dicha aplicación contaba con algunas deficiencias como: el de ingresar notas por materia a través de un disquette que proporcionaba cada maestro al operador que manejaba la aplicación, a continuación éste abría la base de datos y copiaba las notas a la misma, no lo hacían por medio de una interfaz a como lo hacen actualmente con la implementación de la aplicación desarrollada por nosotros mismos, garantizando siempre la seguridad de los datos.

Nosotros hemos desarrollado un sistema de información que permite a demás de llevar el control de notas, gestionar matrículas, certificados, promedios, entre otras funciones.



3. JUSTIFICACIÓN

Hemos automatizado el registro académico del colegio La Salle con la implementación de la aplicación desarrollada, ya que ayuda al personal autorizado a manipular una gran cantidad de datos como: notas de alumnos, profesores, entre otros; de manera ordenada y eficiente.

Con esta aplicación, el Colegio La Salle se ha puesto a la vanguardia frente a otras instituciones del sector educación proveyendo servicios seguros y confiables a todos los posibles usuarios (Profesores, Estudiantes y la Administración).

Los profesores se han beneficiados por la aplicación ya que la automatización del ingreso de notas ha agilizado su trabajo, ahorrándoles tiempo y esfuerzo.

Los estudiantes tienen la oportunidad de visualizar sus calificaciones tanto del año activo como de años anteriores.

La administración tiene un mejor control de todo el registro académico del Colegio.

Nosotros como estudiantes de Ingeniería en Sistemas de Información nos hemos beneficiado de alguna manera, puesto que a través de este trabajo adquirimos nuevos conocimientos además de los que ya poseíamos de acuerdo a nuestro perfil de ISI, y sobre todo porque podremos adquirir el título de ISI que nos merece.



4. OBJETIVOS

OBJETIVO GENERAL:

- Desarrollar una aplicación para la Gestión del Registro Académico del Colegio La Salle - León.

OBJETIVOS ESPECÍFICOS:

- Aplicar el Ciclo de vida del Software para desarrollar la aplicación.
- Automatizar el proceso de matrícula.
- Generar boletines para los estudiantes del colegio.
- Llevar el control de notas de los estudiantes activos y no activos.
- Generar certificados de notas para quienes lo solicitan.
- Asegurar la calidad de nuestro desarrollo aplicando estándares internacionales establecidos para dichos fines.



5. MARCO TEÓRICO

5.1 Sistemas de Información.

5.1.1 Actividades de un Sistema de información.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- **Entrada de Información:**

Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de disquete, los códigos de barras, los escáner, la voz, los monitores sensibles al tacto, el teclado y el Mouse, entre otras.

- **Almacenamiento de Información:**

El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o disquetes y los discos compactos (CD-ROM).

- **Procesamiento de Información:**

Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de



decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

- **Salida de Información:**

La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, disquetes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un sistema de información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfaz automática de salida.

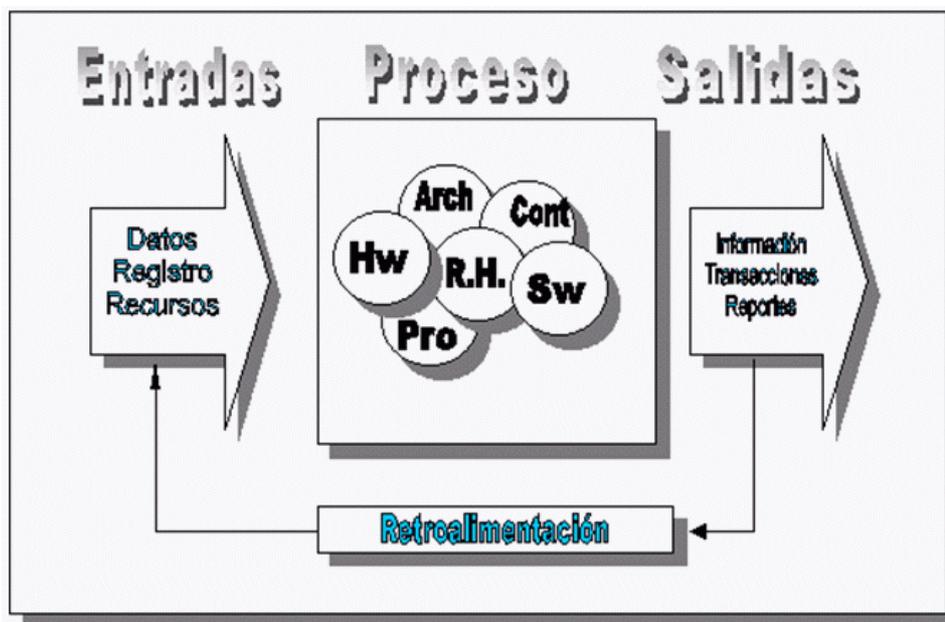


Fig. 1. Actividades de un sistema de información.



5.1.2 Tipos de Sistemas de Información.

Los sistemas de información son desarrollados de acuerdo a diferentes propósitos, las necesidades de la organización y de los niveles organizacionales de la misma. Cada uno de estos sistemas difieren en sus características y cada uno tiene un objetivo fundamental para lograr satisfacer las necesidades de un sistema dentro de una organización.

Actualmente los sistemas de información pueden ser de diferentes tipos, entre los cuales están:

1. Sistemas Transaccionales o Sistemas de Procesamiento de Datos:

Son “sistemas de información computarizados desarrollados para procesar gran cantidad de datos para transacciones rutinarias”.

2. Sistemas de información Gerencial:

No reemplazan a los sistemas de información transaccional, sino que todos los sistemas de información gerenciales incluyen procesamiento de transacciones, también están en la capacidad de generar información para la toma de decisiones.

3. Sistemas de Apoyo a las Decisiones:

Enfatiza el apoyo a la toma de decisiones en todas sus fases, aunque la decisión actual todavía es del dominio del tomador de decisiones.

4. Sistemas Expertos:

Selecciona la mejor solución a un problema, es decir, no interviene un tomador de decisiones en el proceso de buscar la mejor solución.

5.2 Conceptos Básicos de Base de Datos.

- **Atributos:**

Propiedades o características de una entidad.

- **Base de Datos:**

Es una colección de datos interrelacionados, almacenados en conjunto sin redundancia perjudicial o innecesaria; su finalidad es la de servir a una aplicación o más, de la mejor manera posible.



- **Clave Principal:**

Se usa para hacer referencia a registros específicos de una tabla desde otra tabla. También se puede decir cuando uno o más campos se identifican de manera única en cada registro de una tabla.

- **Clave Ajena:**

Es un conjunto de campos de una tabla cuyos valores deben coincidir con los de la clave primaria de otra tabla o ser nulos.

- **Integridad Referencial:**

Garantiza que todas las claves ajenas tomen valores correctos.

- **Registro:**

Es el conjunto de información referida a una misma persona u objeto. Un registro vendría a ser algo así como una ficha.

- **Relación:**

Representa una asociación entre conceptos relevantes del dominio del problema sobre la que el sistema debe almacenar información.

- **Tabla:**

Unidad donde crearemos el conjunto de datos de nuestra base de datos. Estos datos estarán ordenados en columnas verticales.

5.3 PHP.

5.3.1 Origen de PHP.

PHP fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, Personal Home Page Tools.

Al principio, PHP sólo estaba compuesto por algunas macros que facilitaban el trabajo a la hora de crear una página Web. Hacia mediados de 1995 se creó el analizador sintáctico y se llamó PHP/F1 Versión 2, y sólo reconocía el texto HTML y algunas directivas de mSQL. A partir de este momento, la contribución al código fue pública.



El crecimiento de PHP desde entonces ha sido exponencial, y han surgido versiones nuevas como la actual PHP5.

Es una solución para la construcción de páginas Web con independencia de la Base de Datos y del servidor Web, válido para cualquier plataforma.

5.3.2 Características de PHP.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Las principales características de PHP son:

- Rapidez.
- Su facilidad de aprendizaje.
- Su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos.

Se distribuye de forma gratuita bajo una licencia abierta.

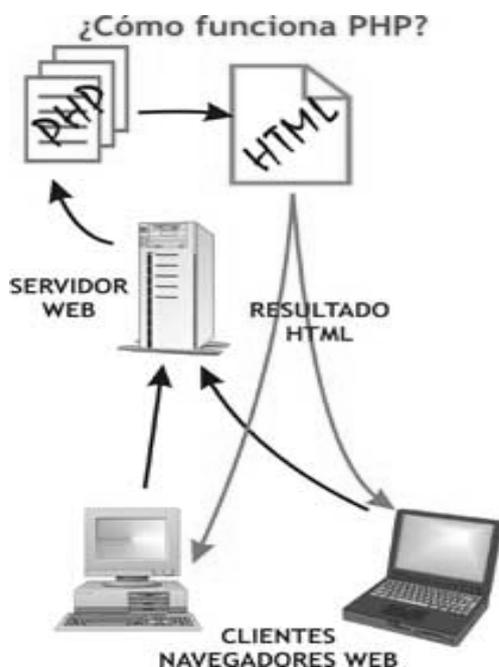


5.3.3 Usos de PHP.

Los principales usos del PHP son los siguientes:

- Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS X.

5.3.4 Cómo funciona PHP.



A diferencia de 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 es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML (Wap).

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP.

El modo de operación de PHP es el siguiente:

Fig. 2. Función de PHP.

- El Navegador realiza una petición al servidor (se escribe la URL).
- Después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador.

Por último el Navegador muestra la respuesta del servidor.



5.4 MYSQL.

5.4.1 Usos de MYSQL.

MySQL es muy rápido, seguro y fácil de usar. MySQL también ha desarrollado un conjunto de características muy prácticas, en estrecha cooperación con otros usuarios. MySQL fue desarrollado para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido usado exitosamente en ambientes de producción con altas demandas, por varios años.

Aunque está bajo un desarrollo constante, MySQL siempre ofrece conjunto de funciones muy poderoso y eficiente. La conectividad, velocidad y seguridad hace de MySQL una suite poderosa para acceder a bases de datos en Internet.

5.4.2 Características Principales de MYSQL.

A continuación se describen algunas de las características más importantes de MySQL:

- Escrito en C y C++, testado con GCC 2.7.2.1. Usa GNU autoconf para portabilidad.
- Clientes C, C++, Eiffel, PHP, Python, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexible y segura.
- Todas las palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.
- 16 índices por tabla, cada índice puede estar compuesto de 1 a 15 columnas o partes de ellas con una longitud máxima de 127 bytes.
- Todas las columnas pueden tener valores por defecto.
- Utilidad *Isamchk* para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.



- Todos los comandos tienen -help o -? Para las ayudas.
- Soporta diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows 95 (con fuentes), se puede utilizar ACCESS para conectar con el servidor.
- Muy rápida usando joins, optimizada para un barrido multi-joins.
- Todas las funciones y operadores soportan en el **SELECT** y **WHERE** como partes de consultas. Ejemplo:

```
mysql> SELECT CONCAT(nombre," ",apellido) FROM nombre_tabla  
WHERE ingreso >10000 AND edad >30
```
- Todas las cláusulas SQL soportan **GROUP BY** y **ORDER BY**.

5.5 Macromedia Dreamweaver.

Dreamweaver es un software fácil de usar que permite crear páginas web profesionales.

Las funciones de edición visual de Dreamweaver 8 permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML.

Se puede crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios web, actualizando el sitio web en el servidor sin salir del programa.

Novedades de Dreamweaver

En este punto comentaremos las características que aporta esta nueva versión sobre la anterior, Dreamweaver MX 2004.

- Integración de RSS: con Dreamweaver 8 podrás integrar entradas RSS provenientes de otras páginas con sólo introducir la fuente y arrastrar y colocar los campos. De esta forma podrás introducir datos en formato XML fácil y cómodamente.
- Mejoras CSS: esta última versión ha mejorado mucho respecto a la compatibilidad y manejo de estilos de cascada. De esta forma se ha



mejorado el panel de estilos CSS, donde ahora podrás acceder a la configuración de cada uno de los estilos desde una lista mucho mejor dotado de una cuadrícula editable desde donde podrás modificar sus propiedades. Además, Dreamweaver 8, añade una nueva barra de herramientas que proporciona la reproducción inmediata de los estilos para diferentes medios (pantalla, impresora, webTV, PDAs...).

- **Accesibilidad:** Dreamweaver 8 incorpora las normas de accesibilidad de prioridad 2 marcadas por la WCAG/W3C.
- **Transferencia de archivos:** Ahora con Dreamweaver 8 podrás seguir trabajando con tus archivos mientras el programa se comunica con tu servidor e incluye los archivos creados o modificados recientemente. Su sincronización ha mejorado notablemente siendo posible una mejor gestión de cambios, además de permitir en uso de bloqueo/desbloqueo de archivos para que estos no se sobre escriban.
- **Interfaz mejorada:** Los usuarios con problemas visuales podrán acceder a una opción de Aumento de la pantalla en vista de diseño para analizar o trabajar con difíciles anidamientos de tablas. Además de la inclusión de información visual gracias a las guías que permitirán la medición píxel a píxel de todos los elementos.
- **Nueva barra de herramientas:** Se ha añadido una barra de herramientas a Dreamweaver 8, podrás encontrarla en la parte lateral izquierda del modo de Código, esta barra hace mucho más accesible el código al permitirnos la navegación por etiquetas y su contracción. Una de las nuevas novedades es la posibilidad de añadir comentarios con un sólo clic.

Editar páginas web

Cualquier editor de texto permite crear páginas web. Para ello sólo es necesario crear los documentos con la extensión HTML o HTM, e incluir como contenido del documento el código HTML deseado. Puede utilizarse incluso el Bloc de notas para hacerlo.

Pero crear páginas web mediante el código HTML es más costoso que hacerlo utilizando un editor gráfico. Al no utilizar un editor gráfico cuesta mucho más insertar cada uno de los elementos de la página, al mismo tiempo que es más complicado crear una apariencia profesional para la página.

Hoy en día existe una amplia gama de editores de páginas web. Uno de los más utilizados, y que destaca por su sencillez y por las numerosas funciones que incluye, es Macromedia Dreamweaver.



Además de Dreamweaver, existen otra serie de buenos editores de páginas web, como pueden ser Microsoft FrontPage, Adobe Pagemill, Adobe GoLive, NetObjects Fusion, CutePage, HotDog Profesional, Netscape Composer y Arachnophilia, algunos de los cuales tienen la ventaja de ser gratuitos.

Páginas Dinámicas

Las páginas dinámicas se almacenan en el servidor y cuando son requeridas se ejecutan y devuelven una salida de código HTML al cliente que lo visualizará en su navegador.

Mientras el archivo PHP se ejecuta en el servidor, puede crear una conexión a una base de datos y extraer o guardar información en ella.



5.6 JavaScript.

JavaScript es una de las múltiples aplicaciones que han surgido para extender las capacidades del Lenguaje HTML. JavaScript es un lenguaje script orientado a documento. No sirve para crear programas, sino para agregarles características a nuestra Web.

Las normas para poder escribir cualquier código de JavaScript se basan en algunos puntos básicos y que debemos cumplir siempre. Estas normas son las siguientes:

- Todo el código (sentencias) esta dentro de funciones.
- Las funciones se desarrollan entre las etiquetas `<script>` y `</script>`.
- Las etiquetas "`<script>`" deben colocarse entre las etiquetas `<head>` y `</head>`.
- Las etiquetas "`<title>`" no pueden estar colocadas entre las de "`<script>`".
- La llamada a la función se hace a través de un evento de un elemento del documento.

Las funciones son un conjunto de sentencias (bloque de código) que especifica al programa las operaciones a realizar. Son útiles para evitar la repetición de líneas y modular el código. Para trabajar con ellas hay que desarrollarlas y llamarlas cuando lo necesitemos.

Sintaxis del desarrollo:

```
function nombre_funcion([var1,var2,varN])
{
    sentencia(s);
}
```

Sintaxis de la llamada:

```
<elemento evento=nombre_funcion([val1,val2,valN]);>
nombre_funcion(valor1,valor2,valorN);
```

En el primero de los casos la llamada se realiza desde un elemento del documento. En el segundo caso la llamada se realiza desde el interior de otra función que también es posible.



5.7 Servidores Webs.

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso.

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear www.wikipedia.org en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio Web clásico podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- **Aplicaciones en el lado del cliente:** el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse mas lenguajes mediante el uso de plugins.
- **Aplicaciones en el lado del servidor:** el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.



Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional.

Algunos conceptos relacionados con las aplicaciones Web son:

- * **PHP**
- * **ASP**
- * **Perl**
- * **CGI**
- * **.NET**
- * **JSP (Tecnología Java)**

Algunos servidores Web importantes son:

- * **Apache**
- * **IIS**
- * **Cherokee**

Otros servidores, más simples pero más rápidos, son:

- * **lighttpd**
- * **thttpd**

5.7.1 Internet Information Server (IIS).

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: **FTP, SMTP, NNTP y HTTP/HTTPS**. Servidor conectado a Internet que puede contener paginas web, servicio de correo electrónico, etc.

Este servicio convierte a un computador en un servidor de internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Existen otros servidores web que pueden usarse como alternativa a este que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software Libre y otros muchos.

Versiones:

- * IIS 1.0, Windows NT 3.51 Service Pack 3
- * IIS 2.0, Windows NT 4.0



- * IIS 3.0, Windows NT 4.0 Service Pack 3
- * IIS 4.0, Windows NT 4.0 Option Pack
- * IIS 5.0, Windows 2000
- * IIS 5.1, Windows XP Professional
- * IIS 6.0, Windows Server 2003 y Windows XP Professional x64 Edition
- * IIS 7.0, Windows Vista y Windows Server "Longhorn"

5.7.2 Servidor Apache.

El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor parcheado). El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

El servidor Apache (**<http://www.apache.org>**) es el servicio que se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de Internet http. En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios.

El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor Web, este término podría referirse a la máquina que almacena y maneja los sitios Web, y en este sentido es utilizada por las compañías que ofrecen hospedaje. Alternativamente, el servidor Web podría referirse al software, como el servidor de HTTP de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas Web como respuesta a peticiones de los navegadores de los clientes.



Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios. Como su nombre implica, un servidor sirve información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor. En la **WWW**, un servidor Web es un ordenador que usa el protocolo http para enviar páginas Web al ordenador de un usuario, cuando el usuario las solicita.

Los servidores Web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar el Internet. Algunos servidores manejan solamente correo o solamente archivos, mientras que otros hacen más de un trabajo, ya que un mismo ordenador puede tener diferentes programas de servidor funcionando al mismo tiempo. Los servidores se conectan a la red mediante una interfaz.

5.8 Arquitectura Cliente / Servidor.

En las redes basadas en arquitecturas Cliente/Servidor, los servidores ponen a disposición de sus clientes recursos, servicios y aplicaciones. Dependiendo de qué recursos ofrece el servidor y cuales se mantienen en los clientes se pueden hacer distinciones entre distintas arquitecturas cliente-servidor.

La arquitectura Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC, TCP/IP, OSI, NFS, DRDA corriendo sobre DOS, OS/2, Windows o PC UNIX, en TokenRing, Ethernet, FDDI o medio coaxial, sólo por mencionar algunas de las posibilidades.

Dependiendo de estas características y su combinación, resulta el grado de complejidad de una solución C/S.

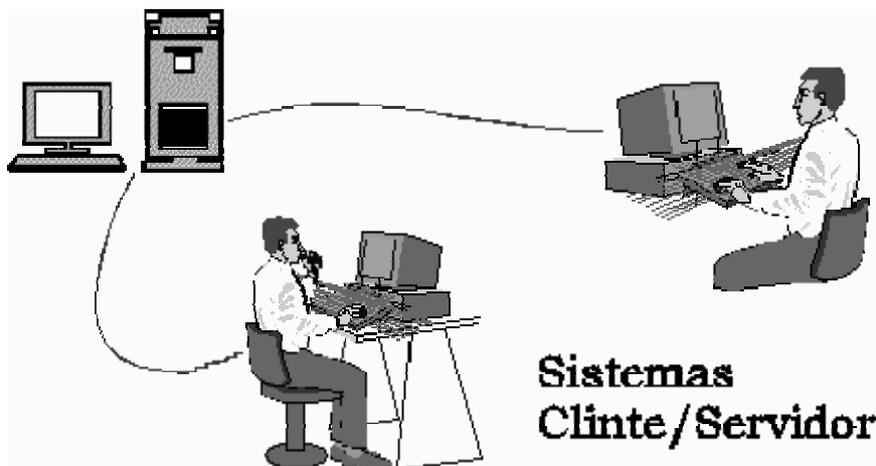


Fig. 3. Arquitectura Cliente/Servidor.

5.9 Protocolo HTTP.

El Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.



Etapas de una transacción HTTP.

Para profundizar más en el funcionamiento de HTTP, veremos primero un caso particular de una transacción HTTP; en los siguientes apartados se analizarán las diferentes partes de este proceso.

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.
Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,...), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,...
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP. Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.



Versiones.

HTTP ha pasado por múltiples versiones del protocolo, muchas de las cuales compatibles con las anteriores. El RFC 2145 describe el uso de los números de versión de HTTP. El cliente le dice al servidor al principio de la petición la versión que usa, y el servidor usa la misma o una anterior en su respuesta.

0.9

Obsoleta. Soporta sólo un comando, GET, y además no especifica el número de versión HTTP. No soporta cabeceras. Como esta versión no soporta POST, el cliente no puede enviarle mucha información al servidor.

HTTP/1.0 (mayo 1996)

Esta es la primera revisión del protocolo que especifica su versión en las comunicaciones, y todavía se usa ampliamente, sobre todo en servidores Proxy.

HTTP/1.1 (junio 1999)

Versión actual; las conexiones persistentes están activadas por defecto y funcionan bien con los proxies. También soporta peticiones en paralelo (pipelining), permitiendo enviar múltiples peticiones a la vez, lo cual hace que el servidor pueda prepararse para una sobrecarga de trabajo y servir las solicitudes más rápidamente al cliente.

HTTP/1.2

Los primeros borradores de 1995 del documento *PEP — an Extension Mechanism for HTTP* (el cuál propone el Protocolo de Extensión de Protocolo, abreviado PEP) los hizo el World Wide Web Consortium y se envió al Internet Engineering Task Force. El PEP inicialmente estaba destinado a convertirse en un rango distintivo de HTTP/1.2. En borradores posteriores, sin embargo, se eliminó la referencia a HTTP/1.2. El RFC 2774 (experimental), *HTTP Extension Framework*, incluye en gran medida a PEP. Se publicó en febrero de 2000.



Ejemplo de un diálogo HTTP.

Para obtener un recurso con el URL `http://www.tuhost.example/index.html`

1. Se abre una conexión al host `www.tuhost.example`, puerto 80 que es el puerto por defecto para HTTP.
2. Se envía un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: nombre-cliente
[Línea en blanco]
```

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
```

```
<html>
<body>
<h1>Página principal de tuHost</h1>
(Contentido)
.
.
.
</body>
</html>
```



5.10 Protocolo TCP/IP.

El Protocolo de Internet (IP) y el Protocolo de Transmisión (TCP), fueron desarrollados inicialmente en 1973 por el informático estadounidense Vinton Cerf como parte de un proyecto dirigido por el ingeniero norteamericano Robert Kahn y patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA, siglas en inglés) del Departamento Estadounidense de Defensa. Internet comenzó siendo una red informática de ARPA (llamada ARPAnet) que conectaba redes de ordenadores de varias universidades y laboratorios en investigación en Estados Unidos. World Wide Web se desarrolló en 1989 por el informático británico Timothy Berners-Lee para el Consejo Europeo de Investigación Nuclear (CERN, siglas en francés).

TCP/IP es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí. Hay que tener en cuenta que en Internet se encuentran conectados ordenadores de clases muy diferentes y con hardware y software incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que es en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto. La arquitectura del TCP/IP consta de cinco niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- **Aplicación:** Se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de ficheros (FTP), conexión remota (TELNET) y otros más recientes como el protocolo HTTP (*Hypertext Transfer Protocol*).
- **Transporte:** Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.
- **Internet:** Es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.



- **Físico:** Análogo al nivel físico del OSI.
- **Red:** Es la interfaz de la red real. TCP/IP no especifica ningún protocolo concreto, así es que corre por las interfaces conocidas, como por ejemplo: 802.2, CSMA/CD, X.25, etc.



Fig. 4. Arquitectura TCP/IP

5.11 Gestión de Calidad de un Producto (Norma ISO- 9001).

Calidad: es el grado de acercamiento a las necesidades y expectativas de los consumidores. Cumpliendo las necesidades y expectativas de los consumidores, se consigue satisfacción en el consumidor, que esta transmite a su entorno, generando más satisfacción.

Gestión de la calidad.

"Aspectos de la función de gestión que determinan y aplican la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, la garantía de calidad y la mejora de la calidad".

Dentro de la gestión de la calidad se observa:

- **Gestión de la calidad de software (ISO 9001):** Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad.
- **Política de calidad (ISO 9001):** Directrices y objetivos generales de una organización, relativos a la calidad, tal como se expresan formalmente por la alta dirección.



La gestión de la calidad se aplica normalmente a nivel de empresa. También puede haber una gestión de calidad dentro de la gestión de cada proyecto.

El aseguramiento de la calidad.

Ante todo se debe conocer:

- **Aseguramiento de la calidad:** "Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad".
- **Aseguramiento de la calidad de software:** Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad.

El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla. Hay quienes prefieren decir garantía de calidad en vez de aseguramiento.

La garantía, puede confundir con garantía de productos, mientras que el aseguramiento pretende dar confianza en que el producto tiene calidad.

El aseguramiento de calidad del software está presente en:

- Métodos y herramientas de análisis, diseño, programación y prueba.
- Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
- Estrategias de prueba multiescala.
- Control de la documentación del software y de los cambios realizados.
- Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
- Mecanismos de medida (métricas).
- Registro de auditorias y realización de informes.

Las actividades para el aseguramiento de calidad del software se detallan en:

- Métricas de software para el control del proyecto.
- Verificación y validación del software a lo largo del ciclo de vida (Incluye las pruebas y los procesos de revisión e inspección).
- La gestión de la configuración del software.



Algunos métodos del aseguramiento:

- Revisiones técnicas y de gestión (su objetivo es la evaluación).
- Inspección (su objetivo es la verificación). ¿Estamos construyendo el producto correcto?
- Pruebas (su objetivo es la validación). ¿Estamos construyendo el producto correctamente?
- Auditorias (su objetivo es la confirmación del cumplimiento).

El control de la calidad.

Se debe conocer:

- **Control de calidad:** "Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio".
- **Control de la calidad del software:** Técnicas y actividades de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad, centradas en mantener bajo control el proceso de desarrollo y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

El control de la calidad del software está centrado en dos objetivos fundamentales:

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

En general, se puede decir que el control de de la calidad del software son las actividades para evaluar la calidad de los productos desarrollados.



6. METODOLOGÍA DEL TRABAJO

La ingeniería del software se centra en el hecho, establecimiento y uso de los principios robustos de la ingeniería. Orientados a obtener software económicos y fiables para que funcionen de manera eficiente en máquinas reales. Este proyecto consta de herramientas de alta calidad permiten mayor rapidez en el procesamiento de los datos.

La ingeniería del software está formada por 3 elementos claves:

- Métodos.
- Herramientas.
- Procedimientos.

6.1 Materiales.

a. **Hardware:** En nuestro proyecto a nivel de hardware empleamos las siguientes herramientas:

- 2 PC con las siguientes características:
 - ✓ Memoria RAM de 256 MB
 - ✓ 40 GB en Disco Duro
 - ✓ Procesador Intel® Athlon™ CPU
 - ✓ Velocidad de procesamiento 1.6 Ghz
 - ✓ Red de Área Local o Intranet

b. **Software:** Las herramientas software que empleamos durante el desarrollo de nuestra aplicación son:

- ✓ Microsoft Word 2003
- ✓ Microsoft Visio 2003
- ✓ Easy Case Profesional Versión 4.21.016
- ✓ Windows Server 2003 (Sistema Operativo)
- ✓ Windows XP Pro. Versión 2002 (Sistema Operativo)
- ✓ MySQL Server 5.0 (Sistema Gestor de Base de Datos)
- ✓ Internet Information Server 5.1 y 6 (Servidores Web)
- ✓ Service Pack 2
- ✓ FireWall de Windows
- ✓ Macromedia Dreamweaver 8 (Editor)
- ✓ PHP 5 (Lenguaje de Programación)



6.2 Métodos

Indican cómo construir técnicamente el software con la ayuda de diferentes tipos de modelos (estructurado, cascada, espiral, etc.) los cuales proporcionan una serie de pasos que se deben seguir durante el proceso de desarrollo.

Los modelos del ciclo de vida realizan actividades comunes como:

- Gestión del proyecto por medio de la descomposición del mismo en etapas.
- Uso de alguna metodología de trabajo en cada etapa.
- Verificación y validación de cada fase de desarrollo.

Descripción del ciclo de vida

El ciclo de vida exige un enfoque sistemático y secuencial acerca del desarrollo del software que indica con el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento.

Los modelos de desarrollo abordan el estudio de las actividades del sistema en las siguientes etapas:

- Planificación.
- Análisis.
- Diseño del sistema.
- Construcción y elaboración del sistema.

El modelo de desarrollo de sistemas que será implementado en nuestro sistema es el **MODELO EN CASCADA**.

Modelo en Cascada

Descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

Actividades del Ciclo de Vida en Cascada

- 1) **Recopilación de información:** estudio del problema para obtener una convicción del sistema que se desea desarrollar. Abarca el cómo trabaja un sistema existente y preparar modificaciones al sistema para cambiar el comportamiento del mismo.



- 2) **Análisis:** Análisis del ERS, especificación de tareas.
- 3) **Diseño:** Definición de las estructuras de datos, arquitectura e interfaces.
- 4) **Codificación:** Conversión del diseño a un lenguaje de programación.
- 5) **Prueba:** El software debe ser probado para descubrir los defectos que puedan existir en la función, en la lógica y en la implementación.
- 6) **Mantenimiento:** La fase de mantenimiento se centra en el cambio. Esta fase aplica los pasos de las fases de definición y de desarrollo, pero en el contexto del software ya existe. Durante la fase d mantenimiento se centran tres tipos de cambios: corrección, adaptación y mejora.

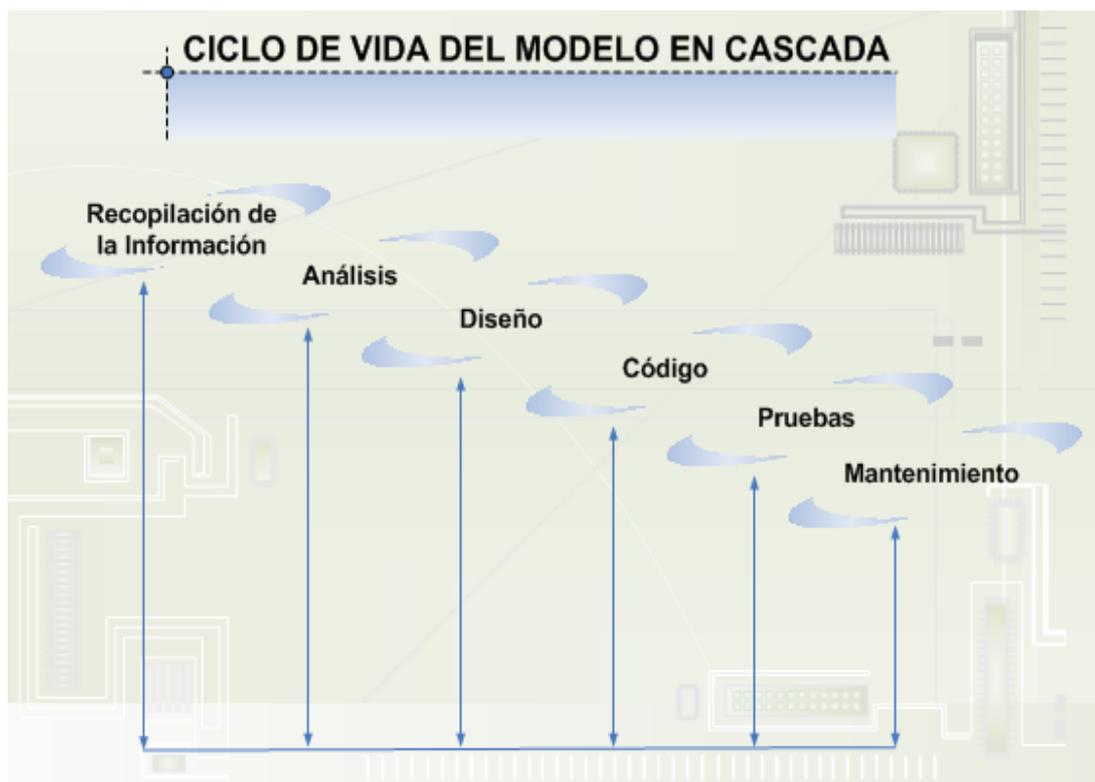


Fig. 5. Fases del Modelo en Cascada.



6.3 Análisis.

6.3.1 Especificaciones de Requisitos Software (ERS)

1.- Introducción.

1.1.- Propósito.

Definición del conjunto de especificaciones de requisitos software que debe cumplir la aplicación de “**Registro Académico del Colegio La Salle- León**”, que consiste en automatizar los procesos de Matrícula y Control de Calificaciones.

Este documento se dirige a la Dirección del Centro y a los Usuarios Finales que deberán estudiarlo para comprender su funcionamiento y hacer un uso correcto y eficiente de la aplicación.

1.2.- Alcance.

El nombre con el que se conocerá a la aplicación será: “**Registro Académico del Colegio La Salle – León**”.

El producto realizará las siguientes funciones:

1. Captura de los datos necesarios para llenar la ficha de ingreso del estudiante.
2. Ordenar por sección los listados de estudiantes matriculados.
3. Agregar una nueva asignatura a la lista de asignaturas existentes.
4. Modificar una asignatura de la lista de asignaturas existentes.
5. Eliminar una asignatura de la lista de asignaturas existentes.
6. Asignar asignaturas a cada profesor existente.
7. Ordenar asignaturas para la impresión de boletines.
8. Ordenar asignaturas para la impresión de certificados de calificaciones.
9. Agregar un nuevo profesor a la lista de profesores existentes.
10. Modificar los datos de un profesor de la lista de profesores existentes.
11. Eliminar un profesor de la lista de profesores existentes.
12. Asignar profesor guía a cada sección.
13. Agregar las calificaciones de los estudiantes por sección.
14. Modificar las calificaciones de un estudiante.
15. Visualizar las calificaciones de un estudiante.
16. Activar el parcial al cual se le agregará las calificaciones de los estudiantes.
17. Mostrar los datos personales de un estudiante.
18. Generar certificados de calificaciones.
19. Registrar el LAF de la calificación final o de reparación por sección.
20. Modificar el LAF de la calificación final o de reparación de un estudiante.



21. Generar boletines por sección.
22. Registrar un nuevo usuario de la aplicación.
23. Modificar la clave de un usuario de la aplicación.
24. Borrar un usuario de la aplicación.
25. Validar entrada de usuarios a la aplicación.
26. Cambiar clave de acceso a herramientas de la aplicación.
27. Cambiar clave de acceso a matrículas de estudiantes.
28. Respalidar las calificaciones de los estudiantes egresados.
29. Se podrán emitir los siguientes listados:
 - Estudiantes Matriculados por grado.
 - Estudiantes ya ordenados por sección.
 - Calificaciones de los estudiantes por sección.
 - Estudiantes egresados del colegio correspondientes al año lectivo.
 - Asignaturas.
 - Profesores.
 - Estudiantes aprobados y reprobados por sección.
 - Los tres mejores promedios por grado o a nivel del colegio.

1.3.- Definiciones, acrónimos y abreviaturas.

- **Estudiante:** Entidad que contendrá información de todos los estudiantes del colegio. De aquí en adelante se llamará ESTUDIANTE.
- **Asignatura:** Entidad que mantendrá información de las asignaturas impartidas a los estudiantes. De aquí en adelante se llamará ASIGNATURA.
- **Profesor:** Entidad que guardará información de los profesores existentes en el colegio. De aquí en adelante se llamará PROFESOR.
- **Calificación:** Entidad que mantendrá información de las calificaciones de los estudiantes. De aquí en adelante se llamará CALIFICACION.
- **Imparte:** Entidad que almacenará información descriptiva referente a las asignaturas impartidas por cada profesor. De aquí en adelante se llamará IMPARTE.
- **Responsable:** Entidad que contendrá información referente a los padres del estudiante, indicándose cuál de los dos es el que responderá por el estudiante. De aquí en adelante se llamará RESPONSABLE.
- **Tutor:** Entidad que guardará información sobre quién responderá por el estudiante en caso de que el responsable no sea uno de los padres. De aquí en adelante se llamará TUTOR.



- **Hermanos del Estudiante:** Entidad que guardará información sobre los hermanos del estudiante que actualmente estudian en el colegio. De aquí en adelante se llamará HERMANOS_ESTUDIANTE.
- **Datos Familiares:** Entidad que almacenará información limitada de los padres del estudiante. De aquí en adelante se llamará DATOS_FAMILIA.
- **Salud del Estudiante:** Entidad que contendrá información descriptiva referente a la salud del estudiante. De aquí en adelante se llamará SALUD_ESTUDIANTE.
- **Usuario:** Entidad que mantendrá información descriptiva de todos los usuarios de la aplicación. De aquí en adelante se llamará USR.
- **Ingreso de Notas:** Entidad que contendrá información sobre el parcial al cual los profesores pueden ingresar las notas de los estudiantes. De aquí en adelante se llamará INGRESONOTAS.
- **Asignaturas para boletines:** Entidad que guardará información referente al orden de las asignaturas para imprimir los boletines. De aquí en adelante se llamará ORDENARASIGBOLETIN.
- **Respaldo de Historial:** Entidad que contendrá información únicamente cuando se cargue un respaldo de los estudiantes egresados para luego poder generar algún certificado de notas de dicho respaldo. De aquí en adelante se llamará RESPALDOHISTORIAL.
- **Asignaturas para certificados de primaria:** Entidad que mantendrá información descriptiva referente al orden de las asignaturas para imprimir un certificado de notas de primaria. De aquí en adelante se llamará ASIG_CERTIFICADOS_P.
- **Asignaturas para certificados de secundaria:** Entidad que guardará información sobre el orden de las asignaturas para imprimir un certificado de notas de secundaria. De aquí en adelante se llamará ASIG_CERTIFICADOS_S.
- **Historial:** Entidad que almacenará información descriptiva referente a las calificaciones obtenidas por los estudiantes desde que ingresaron al colegio hasta la salida del mismo. De aquí en adelante se llamará HISTORIAL.

1.4.- Referencias.

- ❖ Folleto Especificación de Requisitos Software. Suministrado por Msc. Martín Ibarra Padilla.



- ❖ Información obtenida como resultado de las entrevistas del 09/08/07 al 01/10/07 al director, secretaria y profesor de informática del centro. Realizadas por los desarrolladores del proyecto.

1.5.- Visión General.

Primera mente se realizará una visión general del producto a desarrollar para pasar posteriormente a estudiar cada uno de los requisitos específicos individualmente.

2.- Descripción General.

2.1.- Relaciones del producto.

La aplicación se desarrollará en un equipo con las siguientes características:

- ❖ HP 750N.
- ❖ 1.6 GHz.
- ❖ Procesador AMD Athlon.
- ❖ 256 MB de RAM.

El equipo en el que se implementará el producto final es:

- ❖ HP 750N.
- ❖ 2 GHz.
- ❖ Procesador Pentium 4.
- ❖ 160 GB de disco duro.
- ❖ 512 MB de RAM.
- ❖ Sistema Operativo Windows 2003 Server.

2.2.- Funciones del Producto.

El producto software debe realizar las siguientes **funciones**:

- ✓ Cuando se necesite matricular un nuevo estudiante, el usuario debe llenar en el ordenador la ficha de ingreso del mismo para posteriormente imprimirla y agregarla a su expediente.
- ✓ Cuando se desee ordenar por sección los listados de estudiantes matriculados, el usuario deberá introducir el año lectivo de la matrícula y seleccionar de una lista el grado que va a ordenar, luego indicar la sección en la que estará ubicado cada estudiante para posteriormente almacenar el orden indicado en la base de datos.



- ✓ Cuando se desee agregar una nueva asignatura, el usuario debe introducir en el ordenador los datos de la misma, para posteriormente almacenarla en la base de datos.
- ✓ Cuando se desee modificar una asignatura, el usuario debe introducir el nombre de la asignatura para buscar los datos correspondientes de la misma y así modificarlos, posteriormente se actualizarán los datos en la base de datos.
- ✓ Cuando se desee eliminar una asignatura, el usuario deberá de introducir el nombre de la asignatura para su correspondiente búsqueda, a continuación se eliminará de la base de datos.
- ✓ Cuando se desee asignar una asignatura a un profesor, el usuario deberá seleccionar de una lista al profesor, la asignatura y a qué sección le impartirá dicha asignatura para posteriormente almacenarlos en la base de datos.
- ✓ Cuando se desee ordenar las asignaturas para la impresión de boletines, el usuario seleccionará de una lista el orden de las asignaturas que quiera que aparezca en los boletines, para posteriormente almacenar este orden en la base de datos.
- ✓ Cuando se desee ordenar las asignaturas para la impresión de certificados de notas, el usuario seleccionará de una lista el orden de las asignaturas que quiera que aparezca en los certificados, para posteriormente almacenar este orden en la base de datos.
- ✓ Cuando se desee agregar un nuevo profesor, el usuario debe introducir en el ordenador los datos del mismo, para posteriormente almacenarlo en la base de datos.
- ✓ Cuando se desee modificar los datos de un profesor, el usuario debe introducir el nombre del profesor para su correspondiente búsqueda y así modificarlos, posteriormente se actualizarán los datos en la base de datos.
- ✓ Cuando se desee eliminar un profesor, el usuario deberá introducir el nombre del profesor para su correspondiente búsqueda, a continuación se eliminará de la base de datos.
- ✓ Cuando se desee asignar un profesor guía a una sección, el usuario deberá seleccionar de una lista el grado, la sección y el profesor que será el guía de la misma para posteriormente almacenar los datos correspondientes en la base de datos.



- ✓ Cuando se desee registrar las calificaciones de los estudiantes, el usuario deberá seleccionar de una lista la sección y la asignatura a la cual se le ingresará las notas de acuerdo al parcial que tenga permitido ingresar, luego introducirá a cada estudiante su calificación correspondiente para posteriormente almacenarlas en la base de datos.
- ✓ Cuando se desee modificar las calificaciones de un estudiante, el usuario deberá de introducir el número de carnet o los apellidos del estudiante para la búsqueda de las calificaciones correspondientes del mismo y así modificarlas, posteriormente se actualizarán en la base de datos.
- ✓ Cuando se desee visualizar las calificaciones de un estudiante, el usuario debe introducir el número de carnet del estudiante o los apellidos para buscar todas las calificaciones del mismo, y así visualizarlas por pantalla.
- ✓ Cuando se desee activar el parcial al cual los profesores tendrán permitido ingresar notas, el usuario deberá de seleccionar de una lista el parcial que quiere activar para posteriormente almacenar en la base de datos dicho parcial de manera activa.
- ✓ Cuando se desee mostrar los datos personales de un estudiante, el usuario deberá introducir el número de carnet o los apellidos del estudiante para buscar los datos correspondiente del mismo, y posteriormente mostrarlos por pantalla.
- ✓ Cuando se desee generar certificado de calificaciones, el usuario deberá introducir el nombre o número de carnet del estudiante así como indicar los grados que aparecerán en el certificado, y a continuación se imprimirá.
- ✓ Cuando se desee registra el LAF de la nota final o de reparación, el usuario deberá indicar si el LAF será para la nota final o para la de reparación, la sección y la asignatura correspondiente, para posteriormente almacenarlos en la base de datos.
- ✓ Cuando se desee modificar el LAF de la nota final o de reparación de un estudiante, el usuario deberá introducir el número de carnet del mismo, una vez encontrado se deberá introducir nuevamente el LAF para actualizarlo en la base de datos.
- ✓ Cuando se desee generar boletines por sección, el usuario debe seleccionar de una lista la sección y el tipo de boletín, y a continuación se imprimirán; si desea que el boletín de un estudiante en particular no se imprima deberá de indicarlo.



- ✓ Cuando se desee registrar un nuevo usuario de la aplicación, el usuario deberá de ingresar los datos necesarios para el nuevo usuario, y posteriormente se almacenarán en la base de datos.
- ✓ Cuando se desee modificar la clave de un usuario de la aplicación, el mismo usuario se encargará de modificar la clave para posteriormente actualizarla en la base de datos.
- ✓ Cuando se desee borrar un usuario de la aplicación, el usuario deberá ingresar el nombre de usuario para su correspondiente búsqueda, y si es encontrado será eliminado de la base de datos.
- ✓ Cuando un usuario desee entrar a la aplicación, los datos correspondientes a este usuario serán validados con el fin de averiguar de quién se trata y así presentar las opciones que tendrá disponible el usuario.
- ✓ Cuando se desee cambiar la clave de acceso a las herramientas de la aplicación, el mismo usuario se encargará de modificar la clave para posteriormente actualizarla en la base de datos.
- ✓ Cuando se desee cambiar la clave de acceso a matrículas de estudiantes, el mismo usuario se encargará de modificar la clave para posteriormente actualizarla en la base de datos.
- ✓ El usuario podrá obtener los siguientes informes:
 - Estudiantes Matriculados por grado.
 - Estudiantes ya ordenados por sección.
 - Calificaciones de los estudiantes por sección.
 - Estudiantes egresados del colegio correspondientes al año lectivo.
 - Asignaturas.
 - Profesores.
 - Estudiantes aprobados y reprobados por sección.
 - Los tres mejores promedios por grado o a nivel del colegio.

2.3.- Características del Usuario.

El usuario final de la aplicación debe tener conocimientos informáticos mínimos sobre un sistema computarizado.

2.4.- Restricciones Generales.

- El lenguaje de programación utilizado será PHP 5.
- Sistema Gestor de Base de Datos MySQL 5.0 Server Console.
- Servidor de Aplicaciones Internet Information Server 6.0.



- Sistema Operativo Windows XP Pro. Versión 2002 Service Pack 2.
- Editor Macromedia Dreamweaver 2008 Español.

Se deberán seguir los estándares de la programación estructurada.

El Colegio La Salle usuario de la aplicación deberá tener como requisitos mínimos:

- Computadores con: 20GB de disco duro, 128MB de memoria RAM, procesadores con 1200Mhz.
- Sistemas Operativos Windows XP Pro.
- Red de Área Local.

3.- Requisitos Específicos.

3.1- Requisitos Funcionales.

3.1.1- Matrícula de un nuevo estudiante.

3.1.1.1.- Entradas.

Por teclado. Datos del Estudiante:

- Grado /Año.
- Nombres y Apellidos.
- Lugar de Nacimiento.
- Fecha de Nacimiento.
- Sexo.
- Religión.
- Sacramento.
- Centro Educativo Anterior.
- Hermanos Estudiando en el Centro (Si / No).

Datos de los Padres:

- Nombre del Padre.
- Ocupación del Padre.
- Teléfono de la Oficina del Padre.
- Nombre de la Madre.
- Ocupación de la Madre.
- Teléfono de la Oficina de la Madre.
- Estado Civil de los Padres.
- Dirección de la Casa.
- Teléfono de la Casa.

Datos del Responsable: (En caso que no sea uno de los Padres)



- Nombres y Apellidos.
- Parentesco con el Estudiante.
- Ocupación.
- Teléfono.

Datos Familiares:

- Padre vive en el hogar.
- Madre vive en el hogar.

Datos de salud del Estudiante:

- Problema De.
- Hospitalizado.
- Consumo de Medicamentos.
- Tratamiento Médico.
- Seguro Médico.

Datos de desenvolvimiento:

- Comportamiento.
- Con quienes prefiere jugar.
- Actividades preferidas.
- Carácter.
- Se relaciona mejor con.
- Dificultad de expresión.
- Atrae la atención.

3.1.1.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. La fecha de matrícula se tendrá que especificar, de igual manera el número de carnet. El resto de datos deberán ser introducidos por teclado. Si se trata de una matrícula de reingreso los datos del estudiante aparecerán al buscar su número de carnet y podrán ser actualizados. Una vez introducidos los datos se podrán almacenar en las entidades ESTUDIANTE, RESPONSABLE, SALUD_ESTUDIANTE, DESENVOLVIMIENTO_ESTUDIANTE, HERMANOS_ESTUDIANTE, DATOS_FAMILIA y TUTOR para posterior uso.

3.1.1.3.-Salidas.

Con los datos mencionados se podrá imprimir la hoja de matrícula del estudiante para agregarla a su expediente.

3.1.2.- Ordenar por sección los listados de estudiantes matriculados.

3.1.2.1.- Entradas.



Por teclado:

- Año Lectivo.
- Grado.
- Sección.

3.1.2.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo Año Lectivo deberá ser introducido; el grado y la sección deberán ser seleccionados de una lista. Una vez introducidos los datos se podrán almacenar en las entidades CALIFICACION e HISTORIAL para posterior uso.

3.1.2.3.- Salidas.

Se tendrán a los estudiantes ya ubicados en sus respectivas secciones.

3.1.3.- Agregar una nueva Asignatura.

3.1.3.1.- Entradas.

Por teclado:

- Nombre de la Asignatura.
- Grados en los que se impartirá la asignatura.

3.1.3.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombre asignatura se debe introducir por teclado, los grados en los que se impartirá deberán ser indicados en una serie de casillas de verificación. Una vez introducidos los datos se podrán almacenar en la entidad ASIGNATURA para posterior uso.

3.1.3.3.- Salidas.

Se añadirá la nueva asignatura a la lista de asignaturas existentes.

3.1.4.- Modificar una Asignatura.

3.1.4.1.- Entradas.

Por teclado:

- Nombre de la Asignatura.
- Nuevo Nombre.



3.1.4.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombre de la asignatura se deberá seleccionar de una lista, a continuación se debe introducir el nuevo nombre. Posteriormente será actualizada en la entidad ASIGNATURA.

3.1.4.3.- Salidas.

Se modificarán los datos de la asignatura.

3.1.5.- Eliminar una Asignatura.

3.1.5.1.- Entradas.

Por teclado:

- Nombre de la Asignatura.

3.1.5.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombre asignatura se deberá seleccionar de una lista. Posteriormente se eliminará de la entidad ASIGNATURA.

3.1.5.3.- Salidas.

Se eliminará la asignatura correspondiente.

3.1.6.- Asignar una asignatura a un profesor.

3.1.6.1.- Entradas.

Por teclado:

- Nombre Completo del Profesor.
- Asignatura que impartirá.
- Sección a la que le impartirá la asignatura.

3.1.6.2.- Proceso.

El nombre del profesor y la asignatura podrán ser seleccionados de una lista, se deberá indicar la sección a la que el profesor le impartirá la asignatura. Una vez introducidos los datos se podrán almacenar en la entidad IMPARTE para posterior uso.



3.1.6.3.- Salidas.

La asignatura será asignada al profesor correspondiente.

3.1.7.- Ordenar las asignaturas para la impresión de boletines.

3.1.7.1.- Entradas.

Por teclado:

- Nombre de la asignatura.

3.1.7.2- Proceso.

El usuario seleccionará de una lista el orden de las asignaturas que quiera que aparezca en los boletines, para posteriormente almacenar este orden en la entidad ORDENARASIGBOLETIN para posterior uso.

3.1.7.3.- Salidas.

Se almacenará el orden indicado de las asignaturas para los boletines.

3.1.8.- Ordenar las asignaturas para la impresión de certificados de notas.

3.1.8.1.- Entradas.

Por teclado:

- Nombre de la asignatura.
- Nivel.

3.1.8.2- Proceso.

El usuario seleccionará el nivel y la asignatura de una lista, indicando el orden de las asignaturas que quiera que aparezca en los certificados de notas, para posteriormente almacenar este orden en las entidades ASIG_CERTIFICADOS_P y ASIG_CERTIFICADOS_S para posterior uso.

3.1.8.3.- Salidas.

Se almacenará el orden indicado de las asignaturas tanto para certificados de primaria como de secundaria.

3.1.9.- Agregar un nuevo Profesor.

3.1.9.1.- Entradas.



Por teclado:

- Número de Cédula.
- Nombres y Apellidos.
- Dirección.
- Teléfono.
- Especialidad.
- Profesor De.

3.1.9.2- Proceso.

El campo ProfesorDe podrá ser seleccionado de una lista, los campos restantes deberán ser introducidos por teclado. Una vez introducidos los datos se podrán almacenar en la entidad PROFESOR para posterior uso.

3.1.9.3.- Salidas.

Se añadirá un nuevo profesor a la lista de profesores existentes.

3.1.10.- Modificar los datos de un profesor.

3.1.10.1.- Entradas.

Por teclado:

- Nombres y Apellidos del profesor.

Datos proporcionados por el sistema:

- Número de Cedula.
- Dirección.
- Teléfono.
- Especialidad.
- Profesor De.

3.1.10.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombres y apellidos del profesor se debe introducir por teclado para buscar los datos correspondiente al mismo y así modificarlos, a continuación se actualizarán en la entidad PROFESOR.

3.1.10.3.- Salidas.

Se modificarán los datos del profesor correspondiente.



3.1.11.- Eliminar un profesor.

3.1.11.1.- Entradas.

Por teclado:

- Nombres y Apellidos del profesor.

3.1.11.2- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombres y apellidos del profesor se debe introducir por teclado para buscarlo en la base de datos y posteriormente eliminarlo de la entidad PROFESOR.

3.1.11.3.-Salidas.

Se eliminará al profesor correspondiente.

3.1.12.- Asignar un profesor guía a una sección.

3.1.12.1.- Entradas.

Por teclado:

- Grado.
- Sección.
- Profesor Guía.

3.1.12.2- Proceso.

Los campos grado, sección y el profesor guía podrán ser seleccionados de una lista, una vez especificado los datos se almacenará en la entidad PROFESOR la sección de la que es guía dicho profesor.

3.1.12.3.-Salidas.

Se asignará el profesor guía a la sección correspondiente.

3.1.13.- Agregar las calificaciones de los estudiantes.

3.1.13.1.- Entradas.

Por teclado:

- Sección.
- Asignatura.
- Nota.



Datos proporcionados por el sistema:

- Nombres y Apellido del Estudiante.
- Número de carnet.
- Parciales.
- Año Lectivo.
- Profesor Guía.

3.1.13.2.- Proceso.

La asignatura podrá ser seleccionada de una lista, de igual manera la sección; el campo nota será introducido por teclado. Una vez introducidos los datos se podrán almacenar en las entidades CALIFICACION e HISTORIAL para posterior uso.

3.1.13.3.- Salidas.

Se almacenarán las calificaciones de los estudiantes.

3.1.14.- Modificar las calificaciones de un estudiante.

3.1.14.1.- Entradas.

Por teclado:

- Número de carnet o apellidos.
- Parcial.

Datos proporcionados por el sistema:

- Asignaturas.
- Notas.

3.1.14.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El parcial podrá ser seleccionado de una lista, el número de carnet o apellidos del estudiante deberá ser introducido en el campo correspondiente para buscar las calificaciones del mismo y así modificarlas y actualizarlas en las entidades CALIFICACION e HISTORIAL.

3.1.14.3.- Salidas.

Se actualizarán las calificaciones del estudiante correspondiente.



3.1.15.- Visualizar las calificaciones de un estudiante.

3.1.15.1.- Entradas.

Por teclado:

- Número de carnet o apellidos.

Datos proporcionados por el sistema:

- Parciales.

3.1.15.2.- Proceso.

Se mostrará la pantalla de introducción de dato al usuario. El número de carnet o apellidos del estudiante deberá ser introducido en el campo correspondiente para buscar todas las calificaciones del mismo en la entidad CALIFICACION o HISTORIAL y así visualizarlas por pantalla.

3.1.15.3.- Salidas.

Se podrá ver las calificaciones del estudiante.

3.1.16.- Activar el parcial al cual los profesores podrán ingresar notas.

3.1.16.1.- Entradas.

Por teclado:

- Parcial.

3.1.16.2.- Proceso.

El campo parcial deberá ser seleccionado de una lista. Una vez indicado el parcial a activar, se actualizará dicho parcial poniéndolo a estado activo en la entidad INGRESNOTAS.

3.1.16.3.- Salidas.

Se pondrá en estado activo el parcial correspondiente.

3.1.17.- Mostrar los datos personales de un estudiante.

3.1.17.1.- Entradas.

Por teclado:



- Número de carnet o apellidos.

Datos proporcionados por el sistema:

Datos del Estudiante:

- Grado /Año.
- Nombre y Apellidos.
- Lugar de Nacimiento.
- Fecha de Nacimiento.
- Sexo.
- Religión.
- Sacramento.
- Centro Educativo Anterior.
- Hermano Estudiando en el Centro (Si / No).

Datos de los Padres:

- Nombre del Padre.
- Ocupación del Padre.
- Teléfono de la Oficina del Padre.
- Nombre de la Madre.
- Ocupación de la Madre.
- Teléfono de la Oficina de la Madre.
- Estado Civil de los Padres.
- Dirección de la Casa.
- Teléfono de la Casa.

Datos del Responsable: (En caso que no sea uno de los Padres)

- Nombre y Apellidos.
- Parentesco con el Estudiante.
- Ocupación.
- Teléfono.

Datos Familiares:

- Padre vive en el hogar.
- Madre vive en el hogar.

Datos de salud del estudiante:

- Problema De.
- Hospitalizado.
- Consumo de Medicamentos.
- Tratamiento Médico.
- Seguro Médico.



Datos de desenvolvimiento:

- Comportamiento.
- Con quienes prefiere jugar.
- Actividades preferidas.
- Carácter.
- Se relaciona mejor con.
- Dificultad de expresión.
- Atrae la atención.

3.1.17.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El número de carnet o apellidos del estudiante deberá ser introducido en el campo correspondiente para buscar todos los datos personales del mismo en la base de datos y así visualizarlos por pantalla.

3.1.17.3.- Salidas.

Se podrá ver o imprimir los datos personales del estudiante.

3.1.18.- Certificado de calificaciones de un estudiante.

3.1.18.1.- Entradas.

Por teclado:

- No. Carnet o Nombres y Apellidos.
- Grados que aparecerán en el certificado.

Datos proporcionados por el sistema:

- Calificaciones.

3.1.18.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El número de carnet o nombre del estudiante deberá ser introducido, de igual manera los grados a mostrar en el certificado. Una vez introducidos los datos se buscará automáticamente al estudiante correspondiente en la entidad HISTORIAL, al no ser encontrado en dicha entidad, entonces se buscará en la entidad RESPALDOHISTORIAL para posteriormente mostrar o imprimir el certificado.

3.1.18.3.- Salidas.

Se podrá imprimir el certificado del estudiante.



3.1.19.- Registrar el LAF de la nota final o de reparación.

3.1.19.1.- Entradas.

Por teclado:

- Tipo de LAF (Final o Reparación).
- Sección.
- Asignatura.
- LAF.

Datos proporcionados por el sistema:

- No.Carnet de cada estudiante.
- Nombre Completo de cada estudiante.

3.1.19.2.- Proceso.

La sección y la asignatura podrán ser seleccionadas de una lista, el tipo de LAF deberá de indicarse y el LAF debe introducirse. Una vez introducidos los datos se almacenarán en las entidades CALIFICACION e HISTORIAL para posterior uso.

3.1.19.3.- Salidas.

Se almacenará el LAF de la asignatura correspondiente a cada estudiante.

3.1.20.- Modificar el LAF de la nota final o de reparación de un estudiante.

3.1.20.1.- Entradas.

Por teclado:

- No. Carnet.
- Nuevo LAF.

Datos proporcionados por el sistema:

- Nombre Completo del estudiante.

3.1.20.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El No.Carnet y nuevo LAF deberán ingresarse. Una vez introducidos los datos se actualizarán en las entidades CALIFICACION e HISTORIAL para posterior uso.



3.1.20.3.- Salidas.

Se actualizará el LAF de la asignatura correspondiente del estudiante.

3.1.21.- Generar boletines por sección.

3.1.21.1.- Entradas.

Por teclado:

- Tipo de boletín.
- Sección.

Datos proporcionados por el sistema:

- Lista de estudiantes.

3.1.21.2.- Proceso.

Tanto el tipo de boletín como la sección se podrán seleccionar de una lista. Si solo se desea generar el boletín de un estudiante en particular se deberá especificar. Una vez hecha la selección se buscarán automáticamente a los estudiantes correspondientes en las entidades ESTUDIANTE, CALIFICACION y ASIGNATURA, posteriormente se podrán imprimir el boletín del o los estudiantes.

3.1.21.3.- Salidas.

Con los datos mencionados se podrán imprimir los boletines de cada estudiante.

3.1.22.- Registrar un nuevo usuario de la aplicación.

3.1.22.1.- Entradas.

Por teclado:

- Nombre de Usuario.
- Clave.
- Confirmar Clave.
- Tipo de Usuario.

3.1.22.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. Tanto el nombre de usuario como la clave del mismo deberán ingresarse para ser validados, el tipo de usuario deberá seleccionarse de una lista. Una vez hecha la validación, el usuario se almacenará en la entidad USR y así podrá utilizar la aplicación.



3.1.22.3.- Salidas.

Se almacenará al nuevo usuario de la aplicación.

3.1.23.- Modificar la clave de un usuario de la aplicación.

3.1.23.1.- Entradas.

Por teclado:

- Nombre de Usuario.
- Clave Actual.
- Nueva Clave.
- Confirmar Nueva Clave.

3.1.23.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. Todos los datos de entrada serán validados, si la clave actual del usuario no coincide con la almacenada en la entidad USR, no se podrá cambiar la clave del usuario. En cambio si los datos son correctos, la clave del usuario se cambiará y se actualizará en la entidad USR de la base de datos.

3.1.23.3.- Salidas.

Con todos los datos mencionados se podrá modificar la clave del usuario en la base de datos.

3.1.24.- Borrar un usuario de la aplicación.

3.1.24.1.- Entradas.

Por teclado:

- Nombre de Usuario.

3.1.24.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El campo nombre de usuario deberá de ser introducido para posteriormente buscarlo en la entidad USR, y si es encontrado eliminarlo de dicha entidad.

3.1.24.3.- Salidas.

Se borrará al usuario correspondiente de la base de datos.



3.1.25.- Validar entrada de usuarios a la aplicación.

3.1.25.1.- Entradas.

Por teclado:

- Nombre de Usuario.
- Clave.

3.1.25.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. El nombre de usuario y la clave deberán introducirse para ser buscados en la entidad USR de la base de datos. Dependiendo de los datos del usuario, el sistema mostrará una lista de opciones correspondientes al tipo de usuario.

3.1.25.3.- Salidas.

Con los datos antes mencionados el usuario tendrá acceso a algunas funciones de la aplicación.

3.1.26.- Cambiar la clave de acceso a las herramientas de la aplicación.

3.1.26.1.- Entradas.

Por teclado:

- Clave actual.
- Nueva Clave.
- Confirmar Nueva Clave.

3.1.26.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. Se deberán de introducir todos los datos de entrada. Se verificará en la entidad USR que la clave actual pertenezca al usuario aplicación, si es así la clave actual se cambiará por la nueva actualizando de esta manera el registro correspondiente en la entidad USR.

3.1.26.3.- Salidas.

Se cambiará la clave de acceso a las herramientas de la aplicación.

3.1.27.- Cambiar la clave de acceso a matrículas de estudiantes.

3.1.27.1.- Entradas.

Por teclado:



- Clave actual.
- Nueva Clave.
- Confirmar Nueva Clave.

3.1.27.2.- Proceso.

Se mostrará la pantalla de introducción de datos al usuario. Se deberán de introducir todos los datos de entrada. Se verificará en la entidad USR que la clave actual pertenezca al usuario matricula, si es así la clave actual se cambiará por la nueva actualizando de esta manera el registro correspondiente en la entidad USR.

3.1.27.3.- Salidas.

Se modificará la clave de acceso a matriculas.

3.2.- Requisitos de Funcionamiento.

- Requisitos Estáticos: el sistema trabajará de forma remota.
- Requisitos Dinámicos: el tiempo de respuesta del sistema no debe aumentar exponencialmente conforme aumente la cantidad de registros de la base de datos.

3.3.- Requisitos de Diseño.

El formato de pantallas y listados de la aplicación deberá contener información acerca del nombre del Colegio.

3.4.- Atributos.

3.4.1.- Seguridad.

Todos los programas de la aplicación deberán estar protegidos mediante autorizaciones de uso.

3.4.2.- Mantenimiento.

Cualquier modificación que afecte a los requisitos mencionados en este documento, deberá ser reflejada en el mismo, así como la documentación obtenida en las fases de análisis, diseño y codificación.

3.5.- Otros Requisitos.

3.5.1.- Base de Datos.

El almacenamiento de la información se realizará por medio de una base de datos relacional.



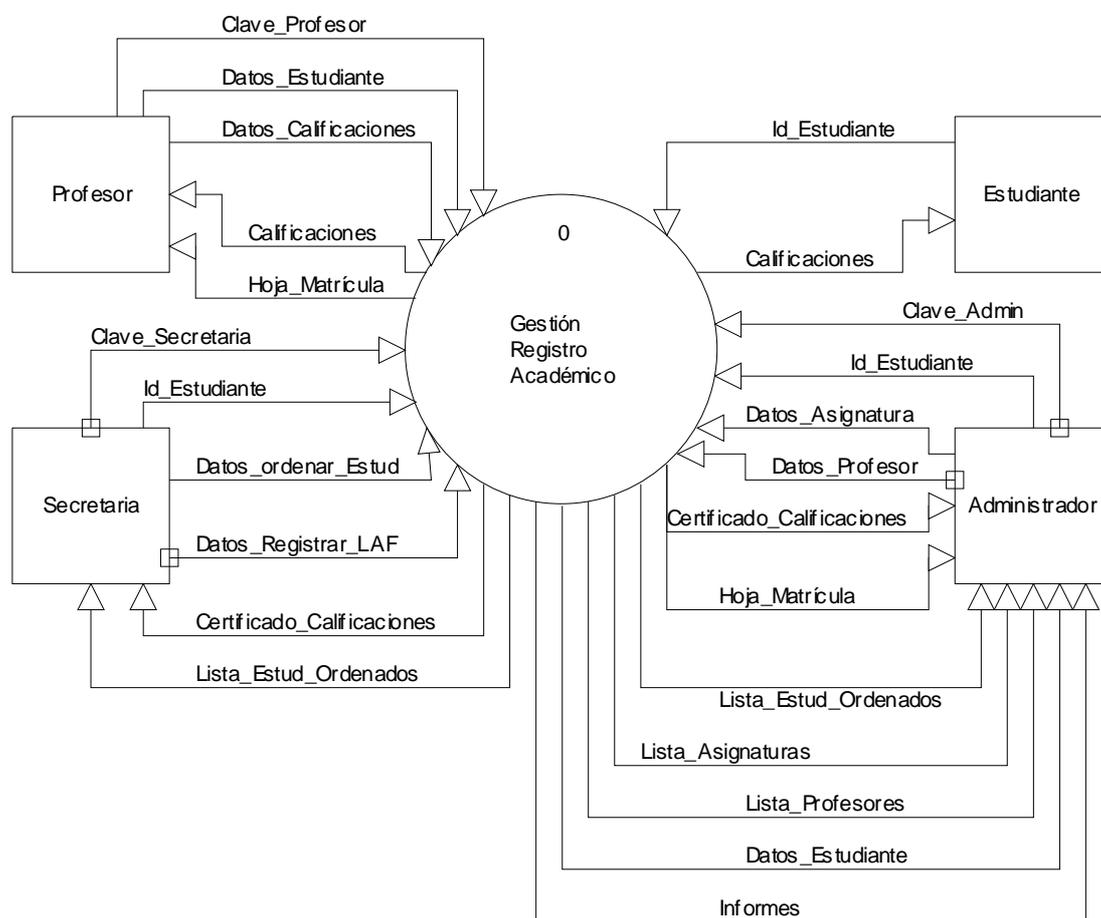
3.5.2.- Operaciones.

Todas las operaciones sobre la base de datos se relazarán según las mencionadas en el subapartado de seguridad.

6.4 Diseños.

6.4.1 Diagrama de Flujo de Datos (DFD).

➤ Diagrama de Contexto.



Nivel 0

Fig. 6. Nivel 0. Diagrama de Contexto.



➤ **Diagrama de Sistema.**

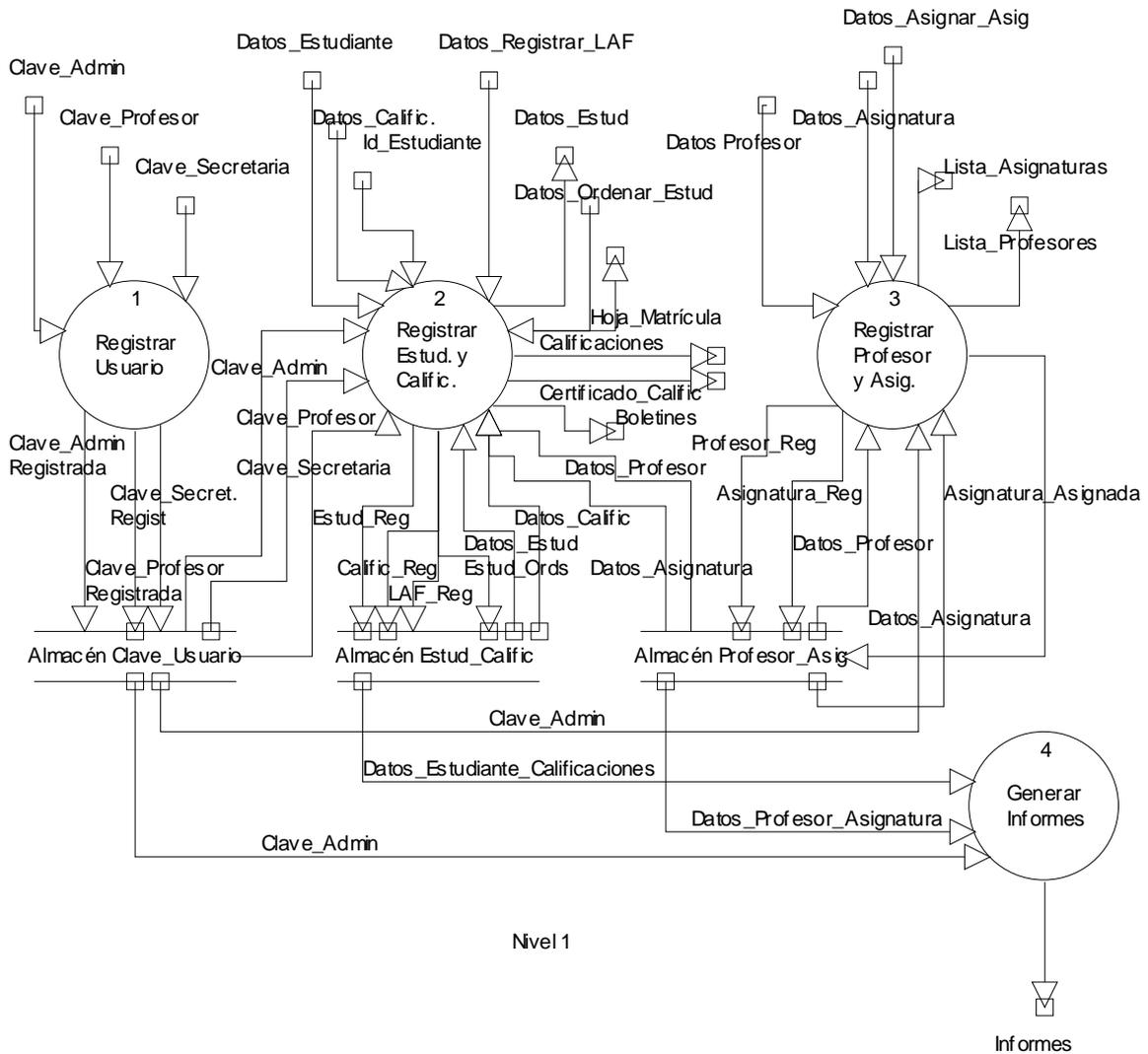
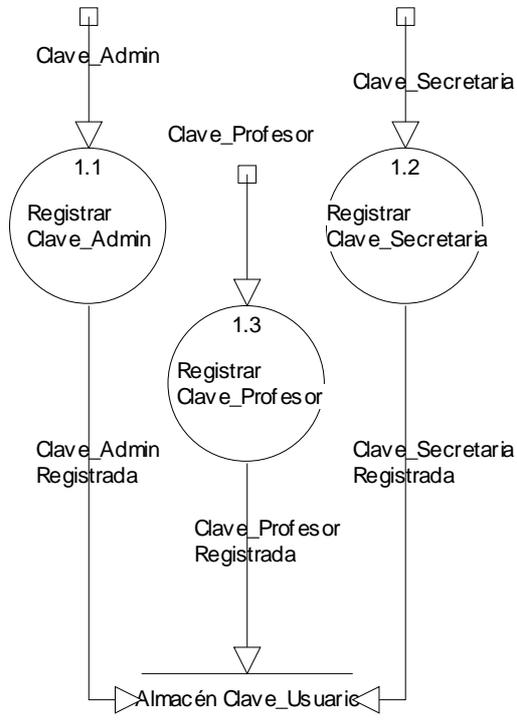
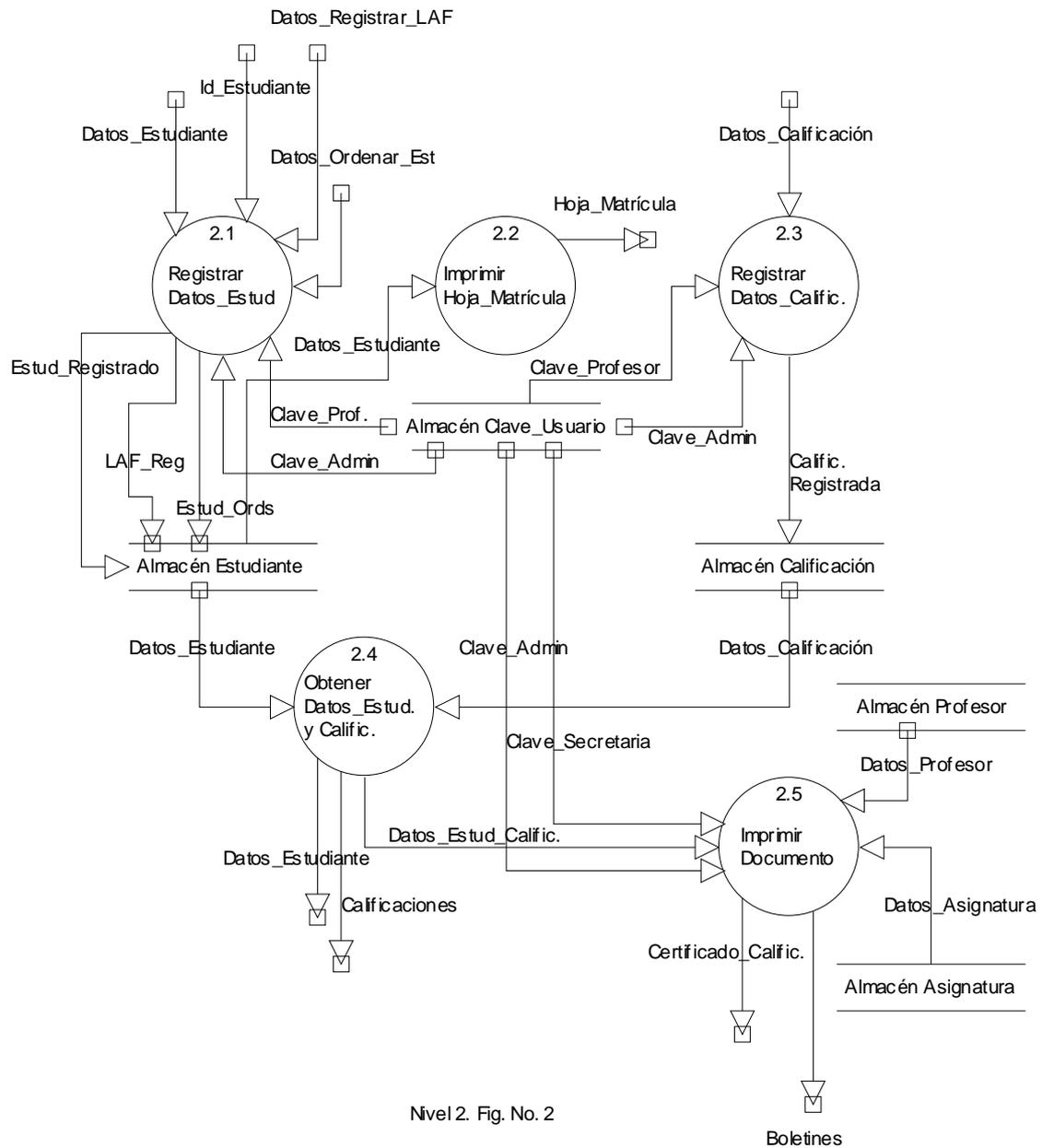


Fig. 7. Nivel 1. Diagrama de Sistema.



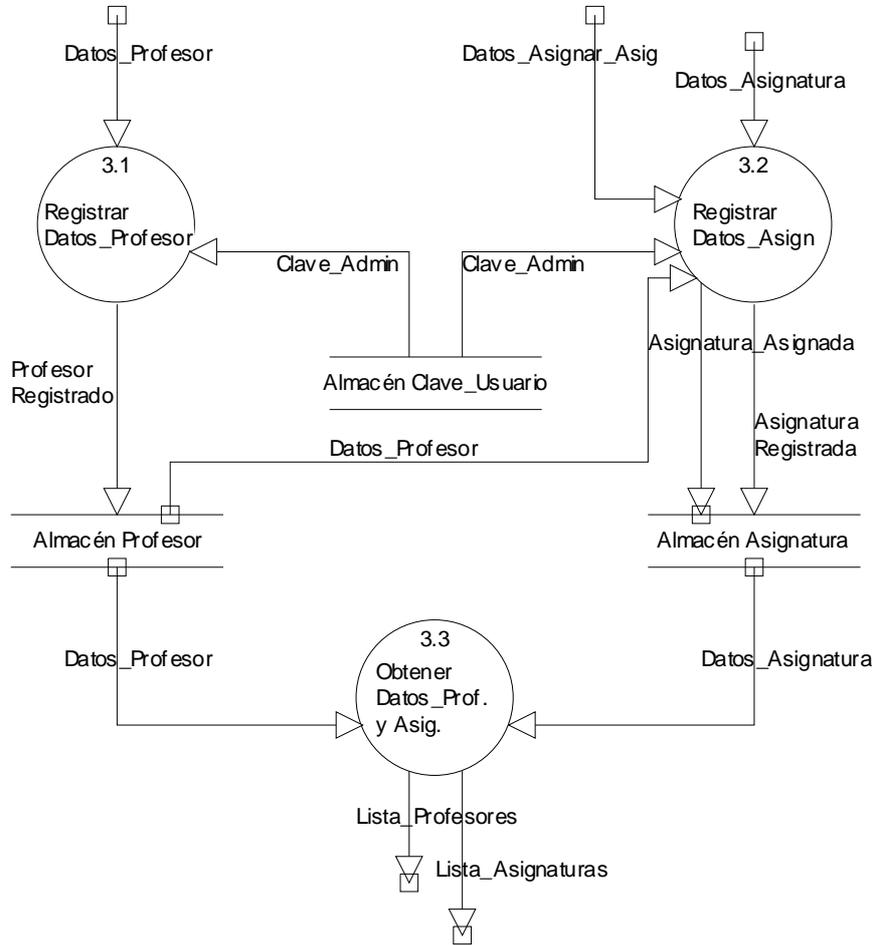
Nivel 2. Fig. No.1

Fig. 8. Nivel 2. Explosión del Proceso No. 1



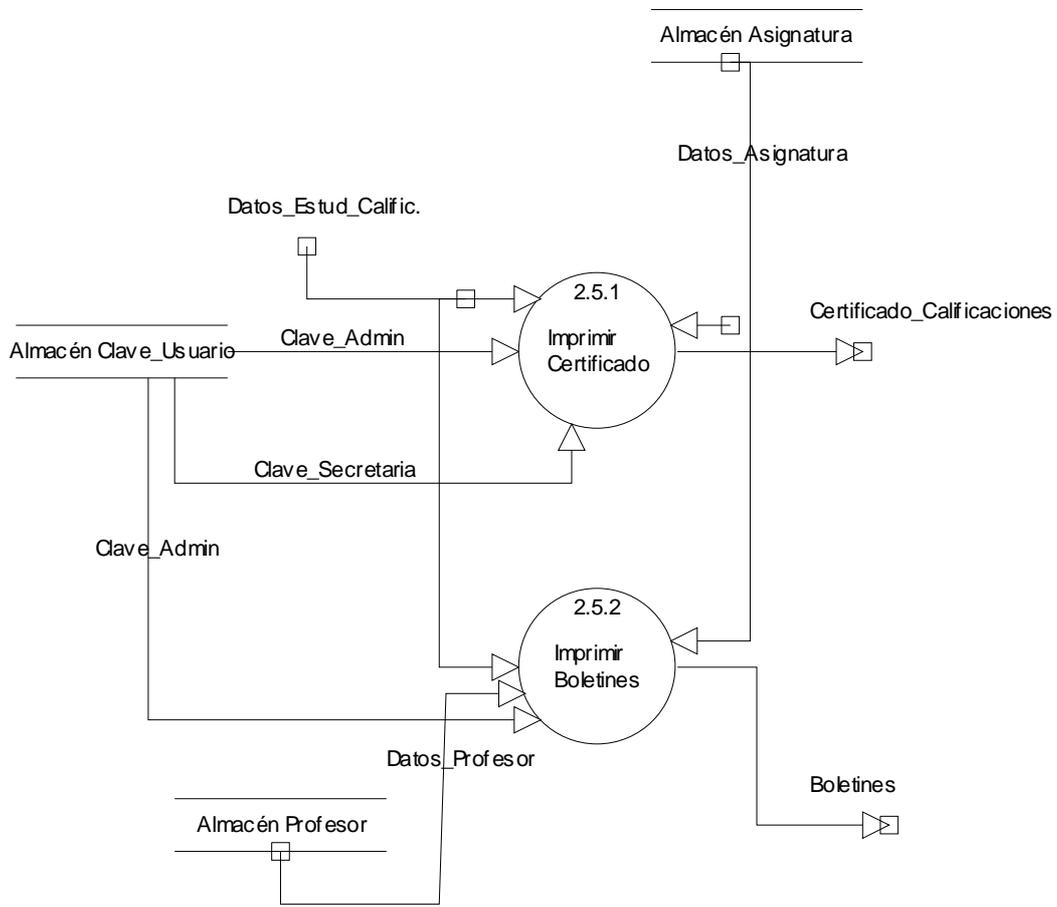
Nivel 2. Fig. No. 2

Fig. 9. Nivel 2. Explosión del Proceso No. 2



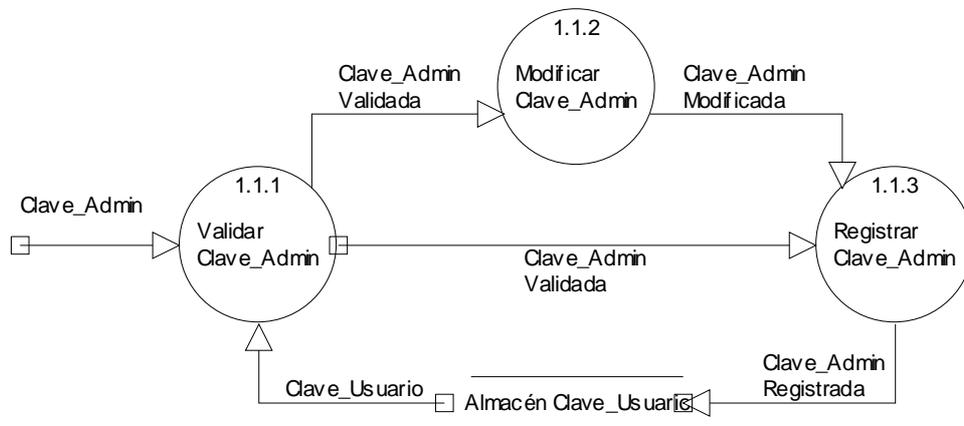
Nivel 2. Fig. No. 3

Fig. 10. Nivel 2. Explosión del Proceso No. 3



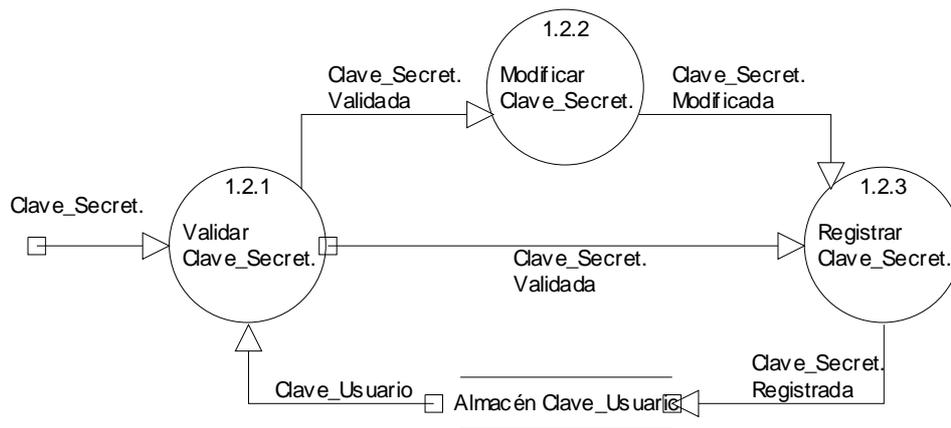
Nivel 2. Fig. No. 4

Fig. 11. Nivel 2. Explosión del Proceso No. 2.5



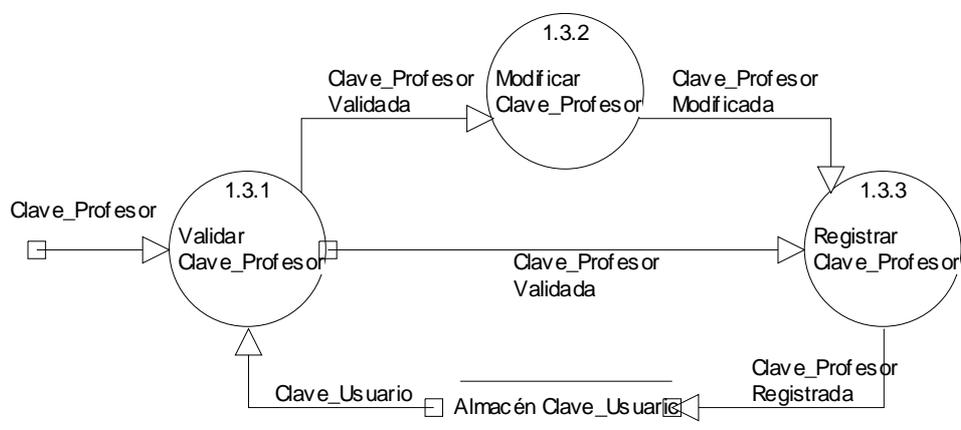
Nivel 3. Fig. No. 1

Fig. 12. Nivel 3. Explosión del Proceso No. 1.1



Nivel 3. Fig. No. 2

Fig. 13. Nivel 3. Explosión del Proceso No. 1.2



Nivel 3. Fig. No. 3

Fig. 14. Nivel 3. Explosión del Proceso No. 1.3

6.4.2 Diccionario de Datos (DD).

- **Datos_Profesor** = Clave_Profesor + Datos_Estudiantes + Datos_Calificaciones.
- **Clave Profesor** = Nombre_Profesor + contraseña.
- **Datos_Estudiante** = Numero_carnet + Nombre + Datos_familia + Datos_salud + Datos_Responsable + Datos_Hermanos + Datos_Desenvolvimiento.
- **Datos_Calificación** = Datos_Estudiante + Datos_Asignaturas
- **Número de Canet** = *Identificador del estudiante compuesto por ocho dígitos*.
- **Nombre:** [A-Z].
- **Datos Familia**= [padre_vive_hogar| madre_vive_hogar].
- **Datos_salud** = [Problema De + Hospitalizado+ Consumo de Medicamentos+ Tratamiento Médico+ Seguro Médico].



- **Problema De**= *Problemas de salud del estudiante*.
- **Hospitalizado** = [si | no].
- **Consumo de Medicamentos**: *Medicamento que consume el estudiante*.
- **Tratamiento Médico**: *Lugar donde recibe tratamiento médico*.
- **Seguro Médico** = [si | no].
- **Datos Responsable**=Nombres+Apellidos+Parentesco+Ocupación+[Teléfono].
- **Datos Hermanos** = *Hermanos del estudiante*.
- **Datos Desarrollo** = comport_estud + pref_jugar_con + act_pref_es + carácter + relacion_mejor_con + dificul_expr + atrae_atencion.
- **comport_estud** = [Receloso | Comunicativo | Impulsivo | Cooperador | Otros].
- **pref_jugar_con** = [Jóvenes de la misma edad | Menores que él o ella].
- **act_pref_es** = *tres actividades que el o la joven prefiere hacer*.
- **carácter** = [Alegre | Agresivo | Comunicativo | Irritable | Otros].
- **relacion_mejor_con** = [Padre | Madre | Hermanos | Otros].
- **dificul_expr** = *Representa las dificultades de expresión que el estudiante pueda tener*.
- **atrae_atencion** = *Especifica si el estudiante atrae la atención y en que forma lo hace*.
- **Datos Asignaturas** = CodAsig + nombreAsig + grado_ano.
- **CodAsig** = [0-9].
- **nombreAsig** = [A-Z].
- **grado_ano** = *Grado en que se impartirá dicha asignatura*.



- **Datos_Secretaria** = Clave_secretaria + Id_Estudiente + Datos_ordenar_Estud + Datos_Registrar_LAF.
- **Clave_Secretaria** = Nombre_Secretaria + Contraseña.
- **Nombre_Secretaria** = [A-Z].
- **Contraseña** = [A-Z | 0-9].
- **Id_Estudiente** = *Numero de carnet del estudiante*.
- **Datos_ordenar_Estud** = anyoLectivo+grado_anyo.
- **anyoLectivo** = *Año lectivo actual*.
- **grado_anyo** = *grado y sección donde quedará ordenado dicho estudiante*.
- **Datos_Administrador** = Clave_Administrador + Id_Estudiente + Datos_Asignatura + Datos_Profesor + Datos_Informe.
- **Datos_Calificaciones** = Numero_Carnet + Codigo_Asignatura + Año_Lectivo + Grado_año + I_Parcial + II_Parcial +III_Parcial +IV_Parcial +Nota_Reparacion+ LAF +LAFR.
- **Clave_Administrador** = Nombre_Administrador + Contraseña.
- **NIP** = *Primer Parcial*.
- **NIP** = [0-100].
- **NIIP** = [0-100].
- **NIIP** = [0-100].
- **NIVP** = [0-100].
- **LAF** = *Libro, Acta y Folio donde se encuentran registradas las notas*.
- **LAFR** = *Libro, Acta y Folio donde se encuentran registradas las notas Reparadas*.



6.4.3 Diagrama Entidad – Relación (E/R).

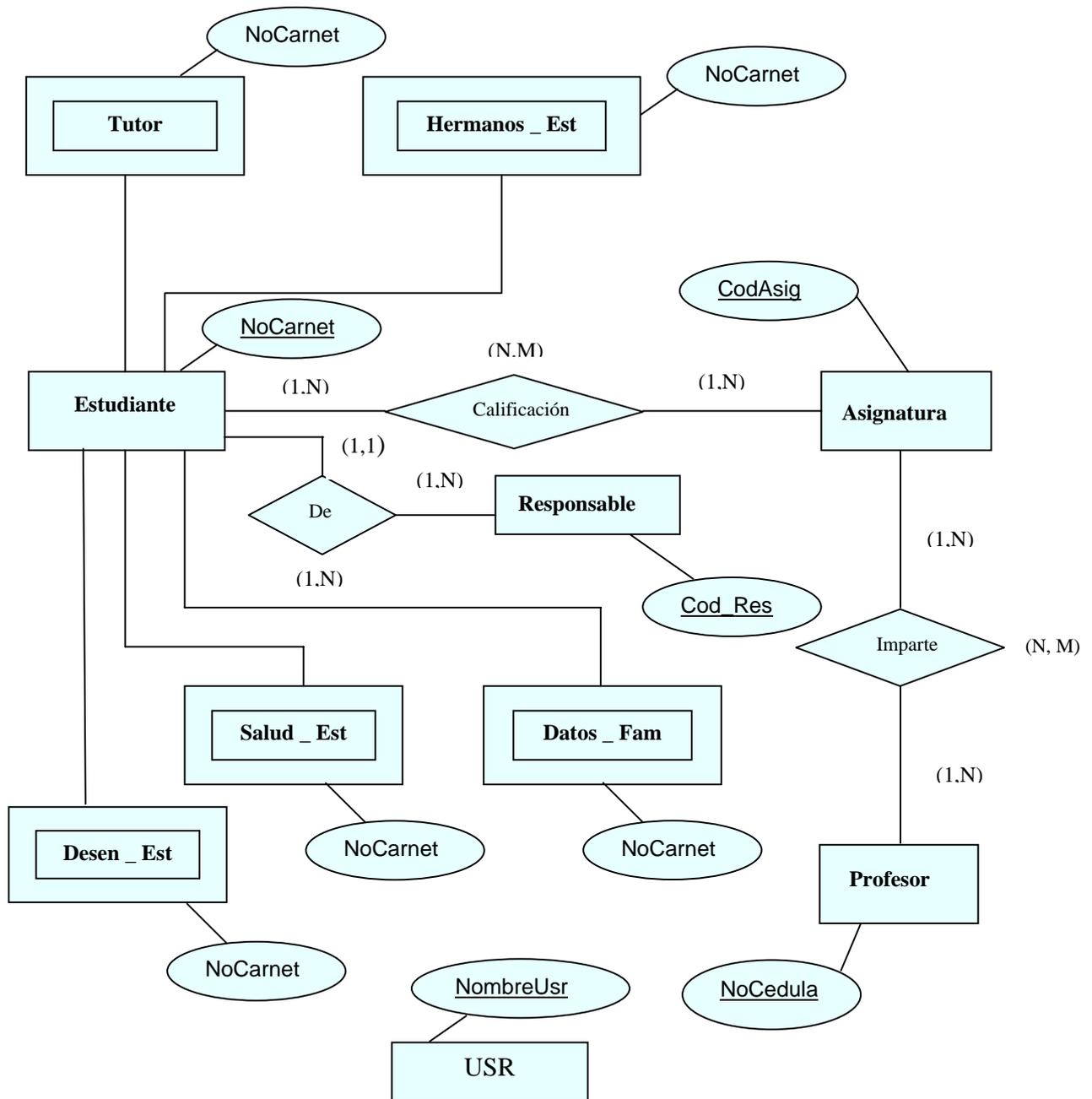


Fig. 15. Diagrama Entidad – Relación (E/R).



6.4.4 Diseño de Datos.

Tabla Responsable		
Nombre	Tipo	Tamaño
cod_resp	numeric	10
nombre_padre	varchar	50
nombre_madre	varchar	50
responsable_P	varchar	50
responsable_M	varchar	50
tel_oficina_Padre	varchar	8
tel_oficina_madre	varchar	8
tel_oficina_casa	varchar	8
ocupacion_padre	varchar	20
ocupacion_madre	varchar	20
estadocivil_padres	varchar	20
direccion_casa	varchar	80

Fig. 16. Tabla Responsable.



Tabla Estudiante		
Nombre	Tipo	Tamaño
nocarnet	varchar	9
nombre	varchar	40
apellido	varchar	40
lugar_nacimiento	varchar	15
sexo	varchar	1
fecha_nacimiento	varchar	10
religion	varchar	30
sacramento	varchar	20
centro_educ_anterior	varchar	50
grado_anyo	varchar	4
fecha_matricula	varchar	10
cod_resp	varchar	10

Fig. 17. Tabla Estudiante.

Tabla Tutor		
Nombre	Tipo	Tamaño
nocarnet	varchar	9
NombreT	varchar	50
ParentescoT	varchar	10
TeIT	varchar	8
OcupacionT	varchar	15

Fig. 18. Tabla Tutor.



Tabla Hermanos_estudiante		
Nombre	Tipo	Tamaño
nocarnet	varchar	9
nombre_herm	varchar	50
grado_anyo_herm	varchar	4

Fig. 19. Tabla Hermanos _ estudiante.

Tabla Datos_familia		
Nombre	Tipo	Tamaño
nocarnet	varchar	9
padre_vive_hogar	varchar	2
causa_no_prespadre	varchar	80
madre_vive_hogar	varchar	2
causa_no_presmadre	varchar	80

Fig. 20. Tabla Datos _ familia.

Tabla Asignatura		
Nombre	Tipo	Tamaño
CodAsig	tinyint	2
nombreAsig	varchar	50
grado_anyo	varchar	4

Fig. 21. Tabla Asignatura.



Tabla Salud_estudiante		
Nombre	Tipo	Tamaño
probl_salud_est	varchar	250
hospitalizado	varchar	2
tiempo_hospit	varchar	25
medicamento	varchar	2
nombre_medic	varchar	35
trata_medic	varchar	2
lugar_trat	varchar	30
seg_medic	varchar	2
nocarnet	varchar	9

Fig. 22. Tabla Salud _ estudiante.

Tabla Desarrollo_estudiante		
Nombre	Tipo	Tamaño
comport_estud	varchar	50
pref_jugar_con	varchar	50
act_pref_est	varchar	100
Caracter	varchar	50
relacion_mejor_con	varchar	40
dificul_expr	varchar	2
dificul_consiste	varchar	40
atrae_atencion	varchar	2
formaAtrae_Aten	varchar	40
nocarnet	varchar	9

Fig. 23. Tabla Desarrollo _ estudiante.



Tabla Calificacion		
Nombre	Tipo	Tamaño
CodAsig	tinyint	2
nocarnet	varchar	9
seccion	varchar	5
anyoLectivo	integer	4
NIP	tinyint	3
NIIP	tinyint	3
NIIP	tinyint	3
NIVP	tinyint	3
NR	tinyint	3
LAF	varchar	15
LAFR	varchar	15

Fig. 24. Tabla Calificación.

Tabla Profesor		
Nombre	Tipo	Tamaño
Nocedula	varchar	16
nombre	varchar	50
Telef	varchar	8
Direccion	varchar	50
Especialidad	varchar	50
ProfGuiaDe	varchar	5
ProfesorDe	varchar	20
NombreUsr	varchar	50

Fig. 25. Tabla Profesor.



Tabla Imparte		
Nombre	Tipo	Tamaño
anyoLectivo	integer	4
grado_anyo	varchar	5
CodAsig	tinyint	2
Nocedula	varchar	16

Fig. 26. Tabla Imparte.

Tabla OrdenarAsigBoletin		
Nombre	Tipo	Tamaño
nombreAsig	varchar	50
grado_anyo	varchar	4
OtraAsig	varchar	2

Fig. 27. Tabla OrdenarAsigBoletin.



Tabla Historial		
Nombre	Tipo	Tamaño
NoCarnet	varchar	9
NombreE	varchar	50
ProfGuia	varchar	50
Asignatura	varchar	50
Anyo_lectivo	integer	4
seccion	varchar	5
NIP	tinyint	3
NIIP	tinyint	3
NIIP	tinyint	3
NIVP	tinyint	3
NReparacion	tinyint	3
LAFFinal	varchar	15
LAFReparacion	varchar	15

Fig. 28. Tabla Historial.



Tabla RespaldoHistorial		
Nombre	Tipo	Tamaño
NoCarnet	varchar	9
NombreE	varchar	50
ProfGuia	varchar	50
Asignatura	varchar	50
Anyo_lectivo	integer	4
seccion	varchar	5
NIP	tinyint	3
NIIP	tinyint	3
NIIP	tinyint	3
NIVP	tinyint	3
NReparacion	tinyint	3
LAFFinal	varchar	15
LAFReparacion	varchar	15

Fig. 29. Tabla RespaldoHistorial.

Tabla Asig_Certificados_P		
Nombre	Tipo	Tamaño
Asignatura	varchar	50

Fig. 30. Tabla Asig _ Certificados _ P.



Tabla Asig_Certificados_S		
Nombre	Tipo	Tamaño
Asignatura	varchar	50

Fig. 31. Tabla Asig _ Certificados _ S.

Tabla IngresoNotas		
Nombre	Tipo	Tamaño
Elemento	varchar	7
valor	varchar	3

Fig. 32. Tabla IngresoNotas.

Tabla auxiliarPlancha		
Nombre	Tipo	Tamaño
parcial	varchar	3
seccion	varchar	4
letra	varchar	1

Fig. 33. Tabla auxiliarPlancha.



Tabla USR		
Nombre	Tipo	Tamaño
NombreUsr	varchar	35
Clave	varchar	32
TipoUsr	varchar	14

Fig. 34. Tabla USR.



6.4.5 Diseño Arquitectónico (DA)

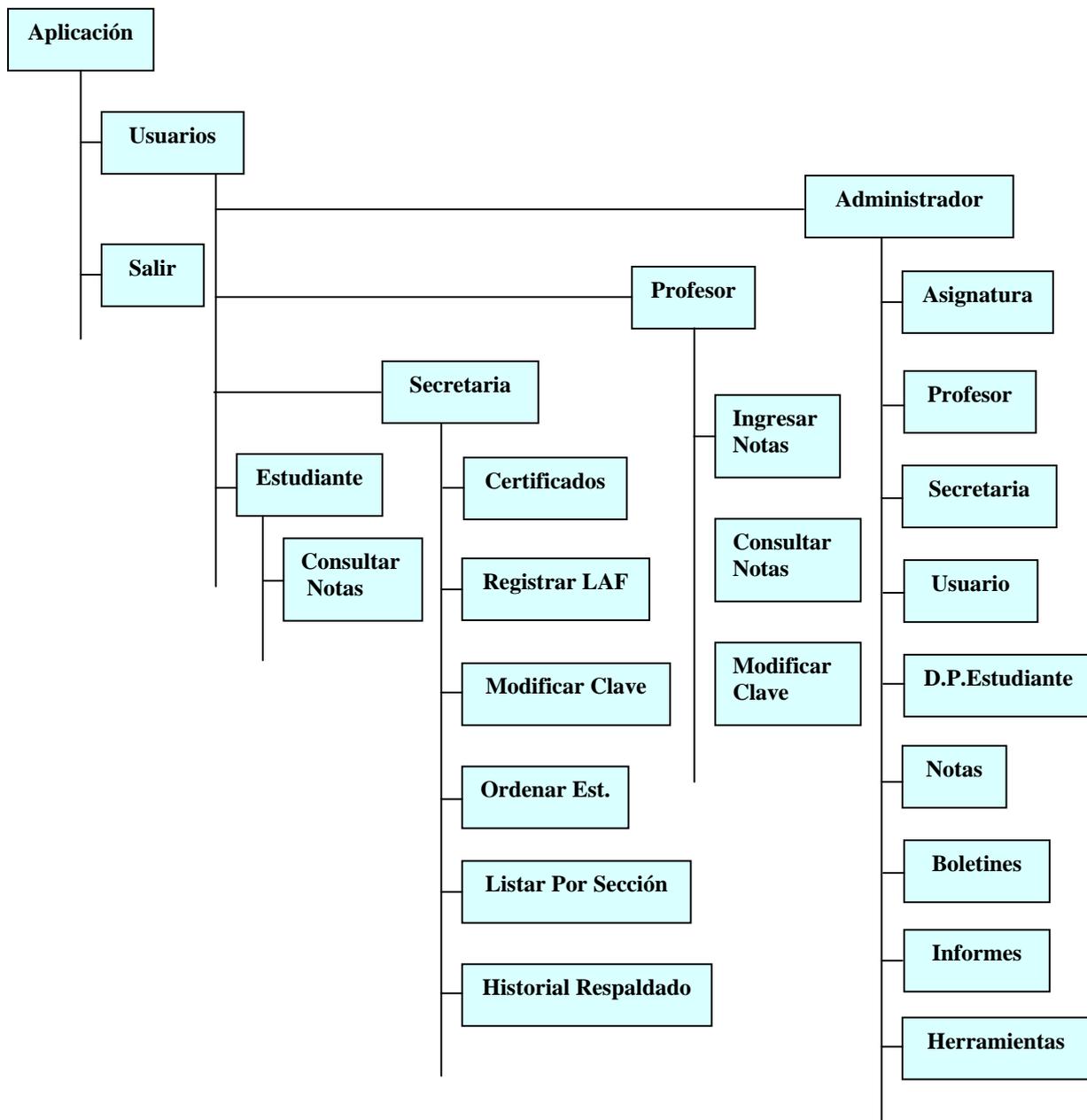


Fig. 35. Diseño Arquitectónico de la Aplicación.



6.4.6 Diseño de Interfaz (DI)

➤ Interfaz Entrada al Sistema.

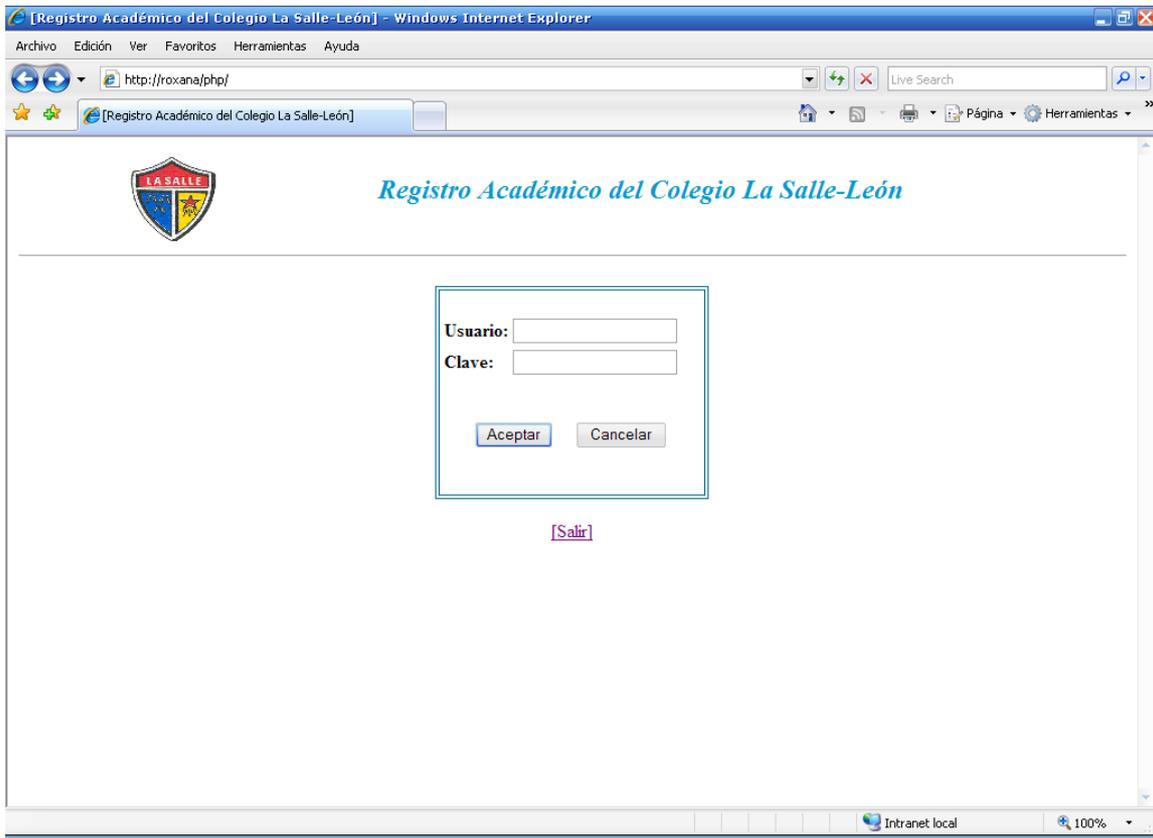


Fig. 36. Entrada al Sistema.



➤ **Interfaz Matrícula.**

[Registro Académico del Colegio La Salle-León] - Windows Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

http://roxana/php/OpcionesUsrMatricula.php

[Registro Académico del Colegio La Salle-León]

COLEGIO LA SALLE **LEÓN, NICARAGUA**

CONTRATO DE INSCRIPCIÓN

Grado: 1 2 3 4 5 6 7 8 9 10 11 Carnet:

DATOS DEL ALUMNO:

Nombres: Apellidos:

Lugar de Nacimiento: Fecha de Nacimiento: D M A

Sexo: M F Religión: Sacramentos: B PC CONF

Centro Educativo Anterior:

¿Actualmente, tiene hermanos estudiando en este colegio? NO SI [Especifique al reverso]

DATOS DE LOS PADRES :

Nombre del padre: Responsable:

Ocupación: Teléf. de Oficina:

Nombre de la madre: Responsable:

Ocupación: Teléf. de Oficina:

Estado Civil de los padres: Casados Unidos Divorciados Soltero(a) Viudo(a)

Dirección de casa: Teléfono:

Listo Intranet local 100%

Fig. 37. Matrícula de Estudiante.



➤ **Interfaz Registrar Usuario.**



Fig. 38. Registrar Usuario del Sistema.



➤ **Interfaz Ingresar Notas.**

The screenshot shows a web browser window titled "[Registro Académico del Colegio La Salle-León] - Windows Internet Explorer". The address bar shows the URL "http://roxana/php/ListarEstIngresarNotaAdmin.php". The page content includes the La Salle logo and the title "Ingreso de Notas a Estudiantes". Below the title, there are three input fields: "Asignatura: Ausencias Injustificadas", "Grado: 1roA", and "Año Lectivo: 2008". A table with 7 columns is displayed below these fields. The columns are: "Num.", "No. CARNET", "NOMBRES Y APELLIDOS", "IP", "IIP", "IIP", and "IVP". The table contains 12 rows of student data, with the last row partially cut off.

Num.	No. CARNET	NOMBRES Y APELLIDOS	IP	IIP	IIP	IVP
1	0000-0000	Aburto Martínez Melissa Marcela	0	0	0	0
2	0000-0077	Aguirre Blandón Angy Elizabeth	0	0	0	0
3	0000-0078	Alvarado Quiroz Ana Gabriela	0	0	0	0
4	0000-0001	Alvarado Gutiérrez Oscar Benito	0	0	0	0
5	0000-0079	Amaya Trujillo Jendryt Dhedylems	0	0	0	0
6	0000-0003	Area Omier Carlos Eduardo	0	0	0	0
7	0000-0036	Arguello Castillo Brandon Jaime Joshua	0	0	0	0
8	0000-0080	Armas Palacios José René	0	0	0	0
9	0000-0081	Artola Hernández Ana Belén	0	0	0	0
10	0000-0082	Avendaño Alvarado Yamilet Del Carmen	0	0	0	0
11	0000-0004	Baca Selva Lucia Carolina	0	0	0	0
12	0000-0008	Baca Torres Rovinsky Manuel	0	0	0	0

Fig. 39. Ingresar Notas.



➤ **Interfaz Consultar Notas.**



Fig. 40. Consultar Notas de Estudiante.



➤ **Interfaz Generar Boletines.**

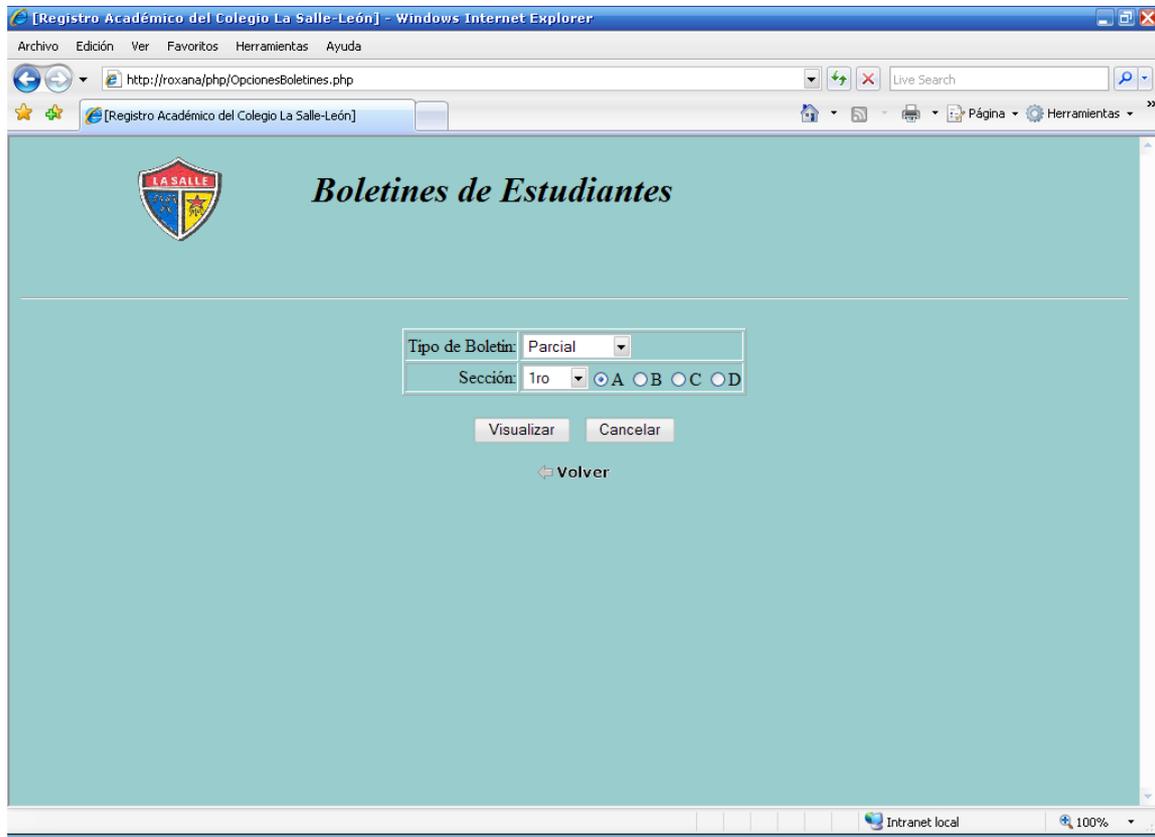


Fig. 41. Generar Boletines.



➤ **Interfaz Generar Certificados.**



Fig. 42. Generar Certificados.



➤ **Interfaz Ordenar Lista Estudiantes.**

Ordenar Lista de Estudiantes

Grado: 1ro. Grado Año Lectivo: 2008

No. CARNET	NOMBRES Y APELLIDOS	SECCION
0000-0000	ABURTO MARTINEZ MELISSA MARCELA	A
0000-0077	AGUIRRE BLANDÓN ANGY ELIZABETH	B
0000-0078	ALAVARADO QUIROZ ANA GABRIELA	C
0000-0001	ALVARADO GUTIERREZ OSCAR BENITO	D
0000-0079	AMAYA TRUJILLO JENDRYT DHEDYLEMS	A
0000-0003	AREA OMIER CARLOS EDUARDO	B
0000-0036	ARGUELLO CASTILLO BRANDON JAIME JOSHUA	D
0000-0080	ARMAS PALACIOS JOSÉ RENÉ	C
0000-0081	ARTOLA HERNÁNDEZ ANA BELÉN	B
0000-0082	AVENDAÑO ALVARADO YAMILETT DEL CARMEN	C
0000-0004	BACA SELVA LUCIA CAROLINA	D
0000-0005	BACA TORRES ROVENSKY MANUEL	A
0000-0006	BARCENAS FLORES ANGEL JOSUE	A

Fig. 43. Ordenar Lista Estudiantes Matriculados.



```

        echo "Error al verificar si el usuario existe.";
        echo "</body></html>";exit;
    }

    if(mysql_num_rows($result0)==1){//si el usuario existe
        $row = mysql_fetch_array($result0);//obtener los datos del usuario que
        entró

        if($row["TipoUsr"]=="Profesor">//si es un profesor
        {
            include("UsrProfesor.php");
        }
        elseif($row["TipoUsr"]=="Matricula">//si se trata del usuario matricula
        {
            include("UsrMatricula.php");
        }
        elseif($row["TipoUsr"]=="SuperAdmin">//si se trata del
        SuperAdministrador
        {
            include("UsrAdmin.php");
            /*echo "<script>open('UsrAdmin.php','SecondWindow',
            'location=yes,menubar=yes,status=yes,toolbar=yes,resizable=yes,top=0,left=0,widt
            h=600,height=300')</script>";*/
        }
        elseif($row["TipoUsr"]=="Administrador">//si se trata de un
        Administrador
        {
            include("UsrAdmin.php");
        }
        elseif($row["TipoUsr"]=="Secretaria">//si se trata de una secretaria
        {
            include("UsrSecretaria.php");
        }
    }
}
else{
    mysql_close($link);//cerrar la conexion a la bd user
    echo "<script>alert(\"Datos de Usuario No Válidos. Intente de
    Nuevo.\");window.history.back()</script>";
}
}
?>
</body>
</html>

```



➤ **Matrícula de Estudiante.**

```

<html>
<head>
<title>[Registro Académico del Colegio La Salle-León]</title>
</head>
<body>
<?php

//recoger valor del boton pulsado
$BotonGuardar=$_REQUEST['btGuardar'];
$BotonActualizar=$_REQUEST['btActualizar'];
$BotonBuscar=$_REQUEST['btbuscar'];
$BotonImprimir=$_REQUEST['btImprimir'];

if($BotonGuardar=="Guardar") //si se pulso guardar
{
    include("variablesMatricula.php");

    if(!$link = conectar("localhost","regacad","root","123456")){ //conectar a
la bd regacad
    echo "Error al establecer la conexion con el servidor MYSQL.";
    echo "</body></html>";
    exit;
    }

    mysql_query("begin",$link);//inicio de la transaccion

    if(!$Resp_consulta=EjecutarConsulta($SelectCarnet,$link)){
    echo "Error al ejecutar la consulta para verificar si el estudiante ya
fue guardado.";
    echo "</body></html>";
    exit;
    }
    if(mysql_num_rows($Resp_consulta)>0){
    echo "<script>alert(\"El estudiante con número de carnet [$NoCarnet]
ya fue guardado. Para actualizar sus datos debe
buscarlo.\");window.close(</script>";
    exit;
    }

    if(Existe_PadreMadre($NombreP,$NombreM))
    {
        //Responsable Existente

```



```

        if(!($result0=EjecutarConsulta($SelectCodResp,$link)){ //obtener
codigo_resp
                echo "Error al ejecutar la consulta para obtener el CodResp.";
                echo "</body></html>";
                exit;
        }
        $CResp=mysql_fetch_array($result0);//obtener codigo del resp
existente
        if(!($result1=EjecutarConsulta($InsertEst."$CResp[0]"),$link)){
//guardar
                //al estudiante con el codigo de resp. existente
                mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
                echo "Error al ejecutar la consulta para insertar al
estudiante con el codigo de resp. existente.";
                echo "</body></html>";
                exit;
        }

        }else{//si no existe el responsable
//Nuevo Responsable
        if(!($result0=EjecutarConsulta($InsertResp,$link)){ //guardar a los
padres
                echo mysql_error();
                mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
                echo "Error al insertar al nuevo responsable.";
                echo "</body></html>";
                exit;
        }
        if(!($result1=EjecutarConsulta($InsertEst."$CodResp"),$link)){
//guardar
                //al estudiante con el codigo de resp.
                mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
                echo "Error al ejecutar la consulta para insertar al
estudiante con el codigo de resp.";
                echo "</body></html>";
                exit;
        }
        }//fin del if Existe_PadreMadre

        if($RespP=="NO" && $RespM=="NO"){//si ninguna de los padres es
resp.
                if(!($result3=EjecutarConsulta($InsertTutor,$link)){//guardar al tutor
                mysql_query("rollback",$link);//deshacer los
cambios realizados por la transaccion

```



```

        echo "Error al insertar al Tutor.";
        echo "</body></html>";
        exit;
    }
}

if(!($result4=EjecutarConsulta($InsertDatosFamilia,$link)){ //guardar
datos family
    mysql_query("rollback",$link);//deshacer los cambios realizados por
la transaccion
    echo "Error al insertar los datos familiares.";
        echo "</body></html>";
        exit;
    }
if(!($result5=EjecutarConsulta($InsertSaludAlumno,$link)){ //guardar
salud alumno
    mysql_query("rollback",$link);//deshacer los cambios realizados por
la transaccion
    echo "Error al insertar los datos de salud.";
        echo "</body></html>";
        exit;
    }
if(!($result6=EjecutarConsulta($InsertDesenvAlumno,$link)){ //guardar
desenvol.
    mysql_query("rollback",$link);//deshacer los cambios realizados por
la transaccion
    echo "Error al insertar los datos de desenvolvimiento.";
        echo "</body></html>";
        exit;
    }
}
//guardar hermanos del estudiante
if(!($Herm1==" || $gradoanno1=="---")){
    if(!($result7=EjecutarConsulta($InsertHerm1,$link)){
        mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
        echo "Error al insertar los hermanos del estudiante.";
            echo "</body></html>";
            exit;
        }
    }
}
if(!($Herm2==" || $gradoanno2=="---")){
    if(!($result8=EjecutarConsulta($InsertHerm2,$link)){
        mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
        echo "Error al insertar los hermanos del estudiante.";
            echo "</body></html>";
            exit;
        }
    }
}

```



```

    }
  }
  if(!($Herm3==" || $gradoanno3=="---")){
    if(!($result9=EjecutarConsulta($InsertHerm3,$link)){
      mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
      echo "Error al insertar los hermanos del estudiante.";
      echo "</body></html>";
      exit;
    }
  }
  mysql_query("commit",$link);//Aceptar los cambios realizados por la transaccion
  mysql_close($link);//cerrar la conexion
  echo "<script>alert(\"Registro de Matricula
Guardado.\");javascript:window.close(</script>");//cerrar la ventana

}elseif($BotonImprimir=="Imprimir")
{
  include("variablesMatricula.php");
  include("ImprimirHojaMatricula.php");//hoja de matricula
}
elseif($BotonBuscar=="Buscar")
{
  include("BuscarEstudiante.php");
}
elseif($BotonActualizar=="Actualizar"){//si se pulsó actualizar

  include("variablesMatricula.php");

  if(!($link = conectar("localhost","regacad","root","123456"))){ //conectar a
la bd regacad
    echo "Error al establecer la conexion con el servidor MYSQL.";
    echo "</body></html>";
    exit;
  }

  mysql_query("begin",$link);//inicio de la transacción

  if(!($result0=EjecutarConsulta($UpdateEst,$link)){//actualizar datos de
estudiante
    mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
    echo "Error al actualizar los datos del estudiante.";
    echo "</body></html>";
    exit;
  }
}

```



```
if($RespP=="NO" && $RespM=="NO"){//si ninguno de los padres es responsable
```

```
    if(!($Result=EjecutarConsulta("select nocarnet from tutor
        where nocarnet like '$NoCarnet', $link))){//verificar si el
        estudiante tiene un tutor que no sea uno de los padres
        mysql_query("rollback", $link);//deshacer los cambios
        realizados por la transaccion
        echo "Error al verificar si el alumno tiene un
        Tutor(Responsable) que no sea uno de los padres.";
        echo "</body></html>";
        exit;
    }
```

```
    if(mysql_num_rows($Result)==1){//si el estudiante tiene tutor,
    entonces lo actualizamos
```

```
        if(!($result1=EjecutarConsulta($UpdateTutor, $link))){//actualizar los
        datos del tutor
        mysql_query("rollback", $link);//deshacer los
        cambios realizados por la transaccion
        echo "Error al actualizar los datos del
        Tutor(Responsable).";
        echo "</body></html>";
        exit;
        }
    }else{
```

```
        if(!($result1=EjecutarConsulta($InsertTutor, $link))){//insertar los datos
        del nuevo tutor
        mysql_query("rollback", $link);//deshacer los
        cambios realizados por la transaccion
        echo "Error al insertar los datos del nuevo
        Tutor(Responsable).";
        echo "</body></html>";
        exit;
        }
    }//fin de si el estudiante tiene tutor
}
```

```
    if(!($result2=EjecutarConsulta($UpdateDatFamily, $link))){//actualizar
    datos de familia
    mysql_query("rollback", $link);//deshacer los cambios
    realizados por la transaccion
    echo "Error al actualizar los datos de Familia.";
```



```

        echo "</body></html>";
        exit;
    }

    if(!($result3=EjecutarConsulta($UpdateSaludEst,$link))){//actualizar
datos de salud del estudiante
        mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
        echo "Error al actualizar los datos de salud del estudiante.";
        echo "</body></html>";
        exit;
    }

    if(!($result4=EjecutarConsulta($UpdateDesenvEst,$link))){//actualizar
datos de desenvolvimiento del estudiante
        mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
        echo "Error al actualizar los datos de desenvolvimiento del
estudiante.";

        echo "</body></html>";
        exit;
    }

    if($TieneHermEst=="SI"){//si el estudiante tiene hermanos en el
colegio
    {

        if(!($result5=EjecutarConsulta($DeleteHermEst,$link))){//eliminar los
hermanos del estudiante
            mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
            echo "Error al eliminar los hermanos del estudiante.";
            echo "</body></html>";
            exit;
        }

        //guardar hermanos del estudiante
        if(!($Herm1=="" || $gradoanno1=="---")){
            if(!($result6=EjecutarConsulta($InsertHerm1,$link)){
                mysql_query("rollback",$link);//deshacer los
cambios realizados por la transaccion
                echo "Error al insertar los hermanos del
estudiante.";

                echo "</body></html>";
                exit;
            }
        }
    }
}

```



```

    }
    if(!($Herm2==" || $gradoanno2=="---")){
        if(!($result7=EjecutarConsulta($InsertHerm2,$link)){
            mysql_query("rollback",$link);//deshacer los
cambios realizados por la transaccion
            echo "Error al insertar los hermanos del
estudiante.";
                echo "</body></html>";
                exit;
            }
        }
    }
    if(!($Herm3==" || $gradoanno3=="---")){
        if(!($result8=EjecutarConsulta($InsertHerm3,$link)){
            mysql_query("rollback",$link);//deshacer los
cambios realizados por la transaccion
            echo "Error al insertar los hermanos del
estudiante.";
                echo "</body></html>";
                exit;
            }
        }
    }
}
else{//si no tiene hermanos

    if(!($result5=EjecutarConsulta($DeleteHermEst,$link))){//eliminar los
hermanos del estudiante
        mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
        echo "Error al eliminar los hermanos del estudiante.";
        echo "</body></html>";
        exit;
    }
}

//fin de si tiene hermanos

if($RespP=="SI" || $RespM=="SI"){//si uno de los padres es el
responsable
    //obtener el codigo de responsable

    if(!($ResultBusqCodRespEst=EjecutarConsulta($SelectCodRespEst,$link))){
        echo "Error al obtener el código de responsable
del estudiante.";
        exit;
    }
}

```



```

        $CodRespEst =
mysql_fetch_array($ResultBusqCodRespEst);//obtener el cod_resp del estudiante

        if(!($result9=EjecutarConsulta($UpdateResp."where
cod_resp like '$CodRespEst[0]",$link))){//actualizar los datos de responsable

                mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
                echo "Error al actualizar los datos de responsable.";
                echo "</body></html>";
                exit;
        }

        if(!($result10=EjecutarConsulta($DeleteTutorEst,$link))){//eliminar al
tutor del estudiante

                mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
                echo "Error al eliminar los datos del tutor(Responsable).";
                echo "</body></html>";
                exit;
        }

    }else{

            //obtener el codigo de responsable

            if(!($ResultBusqCodRespEst=EjecutarConsulta($SelectCodRespEst,$li
nk))){

                    echo "Error al obtener el código de
responsable del estudiante.";

                    exit;

            }
            $CodRespEst =
mysql_fetch_array($ResultBusqCodRespEst);//obtener el cod_resp del estudiante

            if(!($result9=EjecutarConsulta($UpdateResp."where cod_resp like
'$CodRespEst[0]",$link))){//actualizar los datos de responsable
                    mysql_query("rollback",$link);//deshacer los
cambios realizados por la transaccion
                    echo "Error al actualizar los datos de responsable.";
                    echo "</body></html>";
                    exit;
            }

    }//fin de si uno de los padres es responsable

```



```
mysql_query("commit",$link);//Aceptar los cambios realizados por la transaccion
mysql_close($link);//cerrar la conexion a la BD
echo "<script>alert(\"Registro de Matricula
Actualizado.\");javascript:window.close()</script>";//cerrar la ventana
}
?>
</body>
</html>
```

➤ Registrar Usuario.

```
<html>
<head>
<title>[Registro Académico del Colegio La Salle-León]</title>
<style type="text/css">
<!--
body {
    background-color: #99CCCC;
}
-->
</style>
</head>
<body>
<?php
if($_REQUEST['btGuardar']){//si se pulso Guardar

    /**obtener valores de Registrar Usuario**/
    $TipoUser=$_REQUEST['ListaTipoUsr'];//tipo de usuario
    $NomProf=$_REQUEST['ListaProf'];//nombre del profesor
    $NomUser=$_REQUEST['ctUsuario'];//nombre de usuario
    $ClaveUser=md5($_REQUEST['ctClave']);//encriptar la clave del
usuario a partir de
                                                                    //una
semilla generada aleatoriamente por php

    require("Funciones.php");

    if(!($linkRegAcad =
conectar("localhost","regacad","root","123456"))){//conectarnos a la bd regacad
        echo "Error al conectarse con el servidor Mysql.";
        exit;
    }
    //averiguar si el profesor seleccionado ya tiene un nombre de usuario
```



```

        if(!($Resultado=EjecutarConsulta("select nombre from profesor where
nombre='$NomProf' and (not(NombreUsr=*))", $linkRegAcad))){
            echo "Error al verificar si el profesor seleccionado ya tiene un
nombre de usuario.";
            exit;
        }
        if(mysql_num_rows($Resultado)==1){//si el profesor tiene un nombre
de usuario
            echo "<script>alert(\"El Profesor seleccionado ya tiene
asignada una cuenta de usuario.\");window.close()</script>";
            mysql_close($linkRegAcad);//cerrar la conexion a la bd user
            exit;
        }
        mysql_close($linkRegAcad);//cerrar la conexion a la bd regacad

        if(!($linkUser =
conectar("localhost","user","root","123456"))){//conectarnos a la bd user
            echo "Error al conectarse con el servidor Mysql.";
            exit;
        }
        //averiguar si el nombre de usuario existe
        if(!($Resultado=EjecutarConsulta("select NombreUsr from usr where
NombreUsr='$NomUser'", $linkUser))){
            echo "Error al verificar si el usuario existe en la tabla usr.";
            exit;
        }

        if(mysql_num_rows($Resultado)==1){//si el usuario existe
            echo "<script>alert(\"El Nombre de Usuario [$NomUser]
ya Existe. Por favor ingrese otro diferente.\");window.close()</script>";
            mysql_close($linkUser);//cerrar la conexion a la bd user
            exit;
        }

        if($TipoUser=="Profesor"){//si el tipo de usuario es Profesor

            mysql_query("begin", $linkUser);//inicio de la transaccion con la bd
user
            //agregar un nuevo usuario en la tabla usr de la bd user.
            if(!($result1=EjecutarConsulta("insert into
usr(NombreUsr,Clave,TipoUsr)
value('$NomUser','$ClaveUser','$TipoUser')", $linkUser))){
                echo "Error al insertar el nuevo usuario a la tabla usr de la bd
user.";

                mysql_close($linkUser);//cerrar la conexion a la bd user
                exit;
            }
        }
    
```



```

    }

    if(!($linkRegAcad =
conectar("localhost","regacad","root","123456"))){//conectarnos a la bd regacad
        echo "Error al conectarse con el servidor Mysql.";
        exit;
    }
    //asignar un nombre de usuario a un profesor
    if(!($result0=EjecutarConsulta("update profesor set
NombreUsr='$NomUser' where nombre='$NomProf'",$linkRegAcad))){
        echo "Error al asignar el nombre de usuario al profesor.";
        mysql_query("rollback",$linkUser);//deshacer cambios en la
transacción con la bd user
        mysql_close($linkUser);//cerrar la conexion a la bd user
regacad
        exit;
    }
    mysql_close($linkRegAcad);//cerrar la conexion a la bd regacad
    mysql_query("commit",$linkUser);//finalizar la transacción con los
cambios realizados en la bd user
    mysql_close($linkUser);//cerrar la conexion a la bd user

    echo "<script>alert(\"Usuario Registrado.\");window.close()</script>";
}else{//si no es profesor

    if(!($result1=EjecutarConsulta("insert into
usr(NombreUsr,Clave,TipoUsr)
value('$NomUser','$ClaveUser','$TipoUser')",$linkUser))){
        echo "Error al insertar el nuevo usuario a la tabla usr de
la bd user.";
        mysql_close($linkUser);//cerrar la conexion a la bd user
        exit;
    }

    mysql_close($linkUser);//cerrar la conexion a la bd user
    echo "<script>alert(\"Usuario
Registrado.\");window.close()</script>";
}
}elseif($_REQUEST['btModificar']){//si se pulsó modificar

    /**obtener valores de Modificar Clave**/
    $NomUser=$_REQUEST['ctNomUsr'];//nombre de usuario
    $ClaveActual=md5($_REQUEST['ctClaveActual']);//clave actual de
usuario
    $NuevaClave=md5($_REQUEST['ctNuevaClave']);//Nueva Clave de
usuario

```



```

require("Funciones.php");

if(!($linkUser =
conectar("localhost","user","root","123456"))){//conectarnos a la bd user
    echo "Error al conectarse con el servidor Mysql.";
    exit;
}
//averiguar si la clave actual del usuario es la correcta.
if(!($result=EjecutarConsulta("select * from usr where
(NombreUsr='$NomUser' and Clave='$ClaveActual')",$linkUser)){
    echo "Error al verificar si la clave actual corresponde al
usuario.";
    exit;
}

if(mysql_num_rows($result)==1){//si la clave actual corresponde al
usuario
    //cambiar la clave actual del usuario.
    if(!($result=EjecutarConsulta("update usr set
Clave='$NuevaClave' where NombreUsr='$NomUser'",$linkUser)){
        echo "Error al intentar cambiar la clave del usuario.";
        exit;
    }

    if($NomUser=="Aplicacion" || $NomUser=="Matricula")
        echo "<script>alert(\"La Clave ha sido
modificada.\");window.history.back(</script>";
    else
        echo "<script>alert(\"La Clave de Usuario ha sido
modificada.\");window.history.back(</script>";

    }else{
        echo "<script>alert(\"La Clave Actual del Usuario es
inválida.\");window.history.back(</script>";
    }

    mysql_close($linkUser);//cerrar la conexion a la bd user
}elseif($_REQUEST['btBorrar']){//si se pulsó Borrar

    /**obtener valores de Borrar Usuario**/
    $NomUserBorrar=$_REQUEST['ctNomUsrBorrar'];//nombre de usuario
a Borrar
    $NomUser=$_REQUEST['ctUser'];//nombre de usuario que
actualmente ha iniciado sesion

require("Funciones.php");

```



```

        if(!($linkUser =
conectar("localhost","user","root","123456"))){//conectarnos a la bd user
            echo "Error al conectarse con el servidor Mysql.";
            exit;
        }
        //averiguar si el usuario que se intenta borrar existe.
        if(!($result=EjecutarConsulta("select NombreUsr,TipoUsr from usr
where NombreUsr='$NomUserBorrar",$linkUser))){
en la tabla usr.";
            echo "Error al verificar si el usuario que se intenta borrar existe
en la tabla usr.";
            exit;
        }

        //obtener datos del usuario de entrada.
        if(!($resultUsrEntrada=EjecutarConsulta("select NombreUsr,TipoUsr
from usr where NombreUsr='$NomUser",$linkUser))){
en la tabla usr.";
            echo "Error al verificar si el usuario que se intenta borrar existe
en la tabla usr.";
            exit;
        }

        if(mysql_num_rows($result)==1){//si el usuario que se intenta borrar
existe
            $DatosDeUsr=mysql_fetch_array($result);//datos del usuario a
borrar

            $DatosDeUsrEntrada=mysql_fetch_array($resultUsrEntrada);//datos del
usuario de entrada

            if($DatosDeUsr[0]==$DatosDeUsrEntrada[0]){//si el usuario
que entró se quiere borrar
                mysql_close($linkUser);//cerrar la conexion a la bd user
                echo "<script>alert('\Imposible borrar al Usuario
[$NomUserBorrar]. Actualmente ha iniciado sesión con la
Aplicación.\');window.history.back()</script>";
            }

            if($DatosDeUsr[1]=="Profesor"){//si el usuario que se intenta
borrar es de tipo Profesor

                mysql_query("begin",$linkUser);//inicio de la transaccion
con la bd user

                //borrar al usuario de la tabla usr de la bd user.
                if(!($result=EjecutarConsulta("delete from usr where
NombreUsr='$NomUserBorrar",$linkUser))){

```



```

        echo "Error al intentar borrar al usuario.";
        mysql_close($linkUser);//cerrar la conexion a la
bd user
        exit;
    }

    if(!$linkRegAcad =
conectar("localhost","regacad","root","123456")){//conectarnos a la bd regacad
        echo "Error al conectarse con el servidor Mysql.";
        exit;
    }
    //quitar el nombre de usuario del profesor en la tabla
profesor
    if(!$result=EjecutarConsulta("update profesor set
NombreUsr=" where NombreUsr='$NomUserBorrar"', $linkRegAcad)){
        echo "Error al quitar el nombre de usuario del
profesor en la tabla profesor.";
        mysql_query("rollback", $linkUser);//deshacer
cambios en la transacción con la bd user
        mysql_close($linkRegAcad);//cerrar la conexion a
la bd regacad
        mysql_close($linkUser);//cerrar la conexion a la
bd user
        exit;
    }
    mysql_close($linkRegAcad);//cerrar la conexion a la bd
regacad
    mysql_query("commit", $linkUser);//finalizar la
transacción con los cambios realizados en la bd user
    mysql_close($linkUser);//cerrar la conexion a la bd user

    }elseif($DatosDeUsr[1]=="Secretaria"){//si el usuario que se
intenta borrar es de tipo Secretaria

        //borrar al usuario de la tabla usr de la bd user.
        if(!$result=EjecutarConsulta("delete from usr where
NombreUsr='$NomUserBorrar"', $linkUser)){
            echo "Error al intentar borrar al usuario.";
            mysql_close($linkUser);//cerrar la conexion a la
bd user
            exit;
        }
        mysql_close($linkUser);//cerrar la conexion a la bd user
    }elseif($DatosDeUsr[1]=="Administrador"){//si el usuario que
se intenta borrar es de tipo Administrador

```



```

if($NomUser=="Administrador"){//si el que inició sesión
es el SuperAdministrador
//borrar al usuario de la tabla usr de la bd user.
if(!($result=EjecutarConsulta("delete from usr
where NombreUsr='$NomUserBorrar'",$linkUser))){
echo "Error al intentar borrar al usuario.";
mysql_close($linkUser);//cerrar la
conexion a la bd user
exit;
}
mysql_close($linkUser);//cerrar la conexion a la
bd user
}else{
mysql_close($linkUser);//cerrar la conexion a la
bd user
echo "<script>alert(\"Imposible que un
Administrador borre a otro Administrador. Únicamente el SuperAdministrador
puede realizar esta operación.\");window.history.back()</script>";
}
}elseif($DatosDeUsr[1]=="Matricula"){//si el usuario a borrar es
de tipo Matricula
echo "<script>alert(\"Imposible borrar al Usuario
[$NomUserBorrar].\");window.history.back()</script>";
elseif($DatosDeUsr[1]=="SuperAdmin"){//si el usuario que se
intenta borrar es de tipo SuperAdmin
echo "<script>alert(\"Imposible borrar al Usuario
[$NomUserBorrar], es el SuperAdministrador de la
Aplicación.\");window.history.back()</script>";
elseif($DatosDeUsr[1]=="Herramienta"){//si el usuario que se
intenta borrar es de tipo Herramienta
echo "<script>alert(\"Imposible borrar al Usuario
[$NomUserBorrar].\");window.history.back()</script>";

echo "<script>alert(\"Usuario Borrado
Satisfactoriamente.\");window.history.back()</script>";
}else{//si el usuario que se intenta borrar no existe
mysql_close($linkUser);//cerrar la conexion a la bd user
echo "<script>alert(\"El Usuario que intenta Borrar no
existe.\");window.history.back()</script>";
}
}
?>
</body>
</html>

```



➤ **Ingresar Notas.**

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>[Registro Académico del Colegio La Salle-León]</title>
<style type="text/css">
<!--
.Estilo3 {color: #000000}
body {
    background-color: #99cccc;
}
-->
</style>
</head>

<body>
<?php

$contocul=$_REQUEST['contadoroculto'];
$AsigOcul=$_REQUEST['Asignaturaoculto'];
$Secc=$_REQUEST['grad'];
$AnLec=$_REQUEST['anno!'];
$btG=$_REQUEST['btguardar'];

require("variables.php");//incluir variables a usar para consultas y conexion a la BD

if(!($link=conectar("localhost","regacad","root","123456"))){ //conectar a la bd
regacad
    echo "Error al establecer la conexion con el servidor MYSQL.";
    echo "</body></html>";
    exit;
}

if($btG)
{
    $i=0;

    mysql_query("begin",$link);//inicio de la transaccion
    while($i < $contocul)
    {

        $var1=$_POST["txtParcial1$i"];
        $var2=$_POST["txtParcial2$i"];
    }
}

```



```

$var3=$_POST["txtParcial3$i"];
$var4=$_POST["txtParcial4$i"];
$var5=$_POST["txtParcial5$i"];

$carnet=$_POST["carnetoculto$i"];
$NRocul=$_POST["NRoculto$i"];

if($var1>100 || $var2>100 || $var3>100 || $var4>100)
{
    mysql_query("rollback",$link);//deshacer los cambios
realizados por la transaccion
    echo "<script>alert(\"Nota No Valida.\");
        history.back();</script>";
    exit;
}

if($NRocul!="NR")
{
    $notas=EjecutarConsulta("update calificacion set
NIP='$var1',NIIP='$var2',NIIP='$var3',NIVP='$var4' where nocarnet='$carnet' and
CodAsig in(select CodAsig from asignatura where nombreAsig like
'$AsigOcul')",$link);
    if(!$notas)
    {
        mysql_query("rollback");//Deshacer los cambio de
la transaccion
        echo"Error al Guardar en la tabla calificacion";
        exit;
    }
    $notashist=EjecutarConsulta("update historial set
NIP='$var1',NIIP='$var2',NIIP='$var3',NIVP='$var4' where NoCarnet='$carnet' and
Asignatura='$AsigOcul' and Anyo_lectivo='$AnLec' and seccion='$Secc'", $link);
    if(!$notashist)
    {
        mysql_query("rollback");//Deshacer los cambio de
la transaccion
        echo"Error al Guardar en la tabla Historial";
        exit;
    }
}

else
{
    $notas=EjecutarConsulta("update calificacion set
NR='$var5' where nocarnet='$carnet' and CodAsig in(select CodAsig from
asignatura where nombreAsig like '$AsigOcul')",$link);
    if(!$notas)

```



```

        {
            mysql_query("rollback");//Deshacer los cambio de
la transaccion
            echo"Error al Guardar en la tabla calificacion";
            exit;
        }
        $notashist=EjecutarConsulta("update historial set
NReparacion='$var5' where NoCarnet='$carnet' and Asignatura='$AsigOcul' and
Anyo_lectivo='$AnLec' and seccion='$Secc'", $link);
        if(!$notashist)
        {
            mysql_query("rollback");//Deshacer los cambio de
la transaccion
            echo"Error al Guardar en la tabla Historial";
            exit;
        }
    }
    $i++;
}
mysql_query("commit", $link);//Aceptar los cambios realizados por la
transaccion
mysql_close($link);
echo"<script> alert(\"Datos Guardados.\");
        history.go(-2);</script>";
}
?>
</body>
</html>

```

➤ Consultar Notas.

```

<html>
<head>
<title>[Registro Académico del Colegio La Salle-León]</title>
<style type="text/css">
<!--
body {
        background-color: #99CCCC;
    }
-->
</style>
</head>
<body>
<?php

```



```

if($_REQUEST['btAceptar']){//si se pulsó aceptar

//*****obtener valores de consultar notas de estudiante*****//
    $NotasDe=$_REQUEST['NotasDe'];//notas a consultar
    $Grado=$_REQUEST['grado'];//grado seleccionado
    $NoCarnet=$_REQUEST['ctCarnet'];//número de carnet del estudiante

    require("Funciones.php");

    if(!($linkRegAcad =
conectar("localhost","regacad","root","123456"))){//conectarnos a la bd regacad
        echo "Error al conectarse con el servidor Mysql.";
        exit;
    }

    if($NotasDe=="Año Lectivo"){//si el estudiante pide las notas del año
actual
        //obtener el año lectivo del estudiante de la tabla
calificacion
        if(!($DatoAnyoLectivo=EjecutarConsulta("select anyoLectivo
from calificacion where nocarnet='$NoCarnet' limit 1",$linkRegAcad))){
            echo "Error al obtener el año lectivo del estudiante de la
tabla calificacion.";
            mysql_close($linkRegAcad);//cerrar la conexion con la
bd regacad
            exit;
        }

        if(mysql_num_rows($DatoAnyoLectivo)==1){//si la consulta
anterior devolvió una fila

            $AnyoLectivo=mysql_fetch_array($DatoAnyoLectivo);//obtener el valor
de la consulta

            //obtener las notas de los parciales del estudiante
            if(!($ResultadoParciales=EjecutarConsulta("select
Asignatura,NIP,NIIP,NIIP,NIVP from historial where NoCarnet='$NoCarnet' and
Anyo_lectivo='$AnyoLectivo[0]' order by asignatura",$linkRegAcad))){
                echo "Error al obtener las notas de los parciales
del estudiante.";
                mysql_close($linkRegAcad);//cerrar la conexion
con la bd regacad
                exit;
            }

```



```

if(mysql_num_rows($ResultadoParciales)>0){//si
el estudiante tiene sus parciales

```

```

echo "<table width=\"100%\" border=\"0\"
align=\"center\"><tr><td align=\"center\">";

```

```

echo "<table width=\"80%\"><tr><td><div
align=\"left\"><img src=\"SALLEColor.png\" width=80
height=77></div></td><td><div align=left><h1><strong><i>Calificaciones
Obtenidas</i></strong></h1></div></td></tr></table><br><br>";

```

```

echo "<hr width=\"100%\"></hr>";

```

```

echo "<table align=\"center\"><tr><td>";

```

```

//***construir la tabla para mostrar los
parciales***//

```

```

echo "<table align=\"center\" width=500
border=1>";

```

```

echo "<tr
align=center><td><b>ASIGNATURAS</b></td><td><b>IP</b></td><td><b>IIP</b>
</td><td><b>IIIP</b></td><td><b>IVP</b></td></tr>";

```

```

while($NotasParciales=mysql_fetch_array($ResultadoParciales)){//obte
ner el valor de la consulta

```

```

echo
"<tr><td>$NotasParciales[0]</td>
width=30>$NotasParciales[1]</td>
width=30>$NotasParciales[2]</td>
width=30>$NotasParciales[3]</td>
width=30>$NotasParciales[4]</td></tr>";
}

```

```

echo "</table>";//fin de la tabla con los

```

```

parciales

```

```

echo "</td></tr></table>";

```

```

echo "</td></tr></table>";

```

```

echo "<br><center><input type=\"image\"

```

```

name=\"imgVolver\" src=\"volver.gif\"

```

```

onclick=\"javascript:window.close()\"></center> ";

```

```

}elseif(mysql_num_rows($ResultadoParciales)==0){//si no hay ninguna
fila devuelta

```

```

mysql_close($linkRegAcad);//cerrar la
conexion con la bd regacad

```



```

        echo "<script>alert(\"Aún no se han ingresado
notas para el estudiante con No.Carnet [\$NoCarnet] durante este año
lectivo.\");window.close()</script>";
    }

    }elseif(mysql_num_rows(\$DatoAnyoLectivo)==0){//si no
devolvió ninguna fila
        mysql_close(\$linkRegAcad);//cerrar la conexion
con la bd regacad
        echo "<script>alert(\"Estudiante con No.Carnet
[\$NoCarnet] no tiene registro de notas del grado
actual.\");window.close()</script>";
    }

    }elseif(\$NotasDe=="Grados Anteriores"){//si el estudiante pide las
notas de grados anteriores

        //obtener la fecha de matricula mayor
        if(!(\$DatoFechaMat=EjecutarConsulta("select
max(fecha_matricula) from estudiante",\$linkRegAcad))){
            echo "Error al obtener la fecha de matricula mayor.";
            mysql_close(\$linkRegAcad);//cerrar la conexion con la
bd regacad
            exit;
        }

        \$FechaMat=mysql_fetch_array(\$DatoFechaMat);//obtener
valor de la fecha de matricula mayor
        \$DividirFechaMat=split("/",\$FechaMat[0]);//dividir la cadena
fecha en dd/mm/aaaa
        \$AnyoMatMayor=\$DividirFechaMat[2]);//asignar año de
matricula mayor

        //averiguar si el estudiante ya cursó el grado seleccionado
        if(!(\$DatoGrado=EjecutarConsulta("select nocarnet from
historial where nocarnet='\$NoCarnet' and seccion like '\$Grado%' and
(not(anyo_lectivo='\$AnyoMatMayor')) group by nocarnet",\$linkRegAcad))){
            echo "Error al verificar si el estudiante ya cursó el grado
seleccionado.";
            mysql_close(\$linkRegAcad);//cerrar la conexion con la
bd regacad
            exit;
        }

        if(mysql_num_rows(\$DatoGrado)==1){//si la consulta anterior
devolvió una fila

```



```

//obtener el año lectivo mayor del estudiante, ya
que el estudiante pudo haber repetido un grado
        if(!($DatoAnyoLectivo=EjecutarConsulta("select
anyo_lectivo from historial where nocarnet='$NoCarnet' and seccion like '$Grado%'
and (not(anyo_lectivo='$AnyoMatMayor')) group by nocarnet,anyo_lectivo,seccion
order by anyo_lectivo desc",$linkRegAcad))){
estudiante.";
        echo "Error al obtener el año lectivo mayor del
mysql_close($linkRegAcad);//cerrar la conexion
con la bd regacad
        exit;
    }

    $AnyoLectivo=mysql_fetch_array($DatoAnyoLectivo);//obtener valor del
año lectivo del estudiante

        //obtener las notas de los parciales del estudiante
        if(!($ResultadoParciales=EjecutarConsulta("select
Asignatura,NIP,NIIP,NIIP,NIVP,NReparacion from historial where
NoCarnet='$NoCarnet' and Anyo_lectivo='$AnyoLectivo[0]' and seccion like
'$Grado%' order by asignatura",$linkRegAcad))){
del estudiante.";
        echo "Error al obtener las notas de los parciales
mysql_close($linkRegAcad);//cerrar la conexion
con la bd regacad
        exit;
    }

    if(mysql_num_rows($ResultadoParciales)>0){//si
el estudiante tiene sus parciales

        echo "<table width=\"100%\" border=\"0\"
align=\"center\"><tr><td align=\"center\">";
        echo "<table width=\"80%\"><tr><td><div
align=\"left\"><img src=\"SALLEColor.png\" width=80
height=77></div></td><td><div align=left><h1><strong><i>Calificaciones
Obtenidas</i></strong></h1></div></td></tr></table><br><br>";
        echo "<hr width=\"100%\"></hr>";
        echo "<table align=\"center\"><tr><td>";

        //***construir la tabla para mostrar los
parciales, notas finales y de reparación***//
        echo "<table align=\"center\" width=500
border=1>";

```



```

                                echo "<tr
align=center><td><b>ASIGNATURAS</b></td><td><b>IP</b></td><td><b>IIP</b>
></td><td><b>IIIP</b></td><td><b>IVP</b></td><td><b>NF</b></td><td><b>NR
</b></td></tr>";

                                while($NotasParciales=mysql_fetch_array($ResultadoParciales)){//obte
ner el valor de la consulta

                                $NotaFinal=($NotasParciales[1]+$NotasParciales[2]+$NotasParciales[3
]+$NotasParciales[4])/4;

                                echo
                                "<tr><td>$NotasParciales[0]</td>
                                <td align=center
                                width=30>$NotasParciales[1]</td>
                                <td align=center
                                width=30>$NotasParciales[2]</td>
                                <td align=center
                                width=30>$NotasParciales[3]</td>
                                <td align=center
                                width=30>$NotasParciales[4]</td>
                                <td align=center
                                width=30>".round($NotaFinal)."</td>
                                <td align=center
                                width=30>$NotasParciales[5]</td></tr>";
                                }
                                echo "</table>";//fin de la tabla con los
parciales, notas finales y de reparación

                                echo "</td></tr></table>";
                                echo "</td></tr></table>";
                                echo "<br><center><input type=\"image\"
name=\"imgVolver\" src=\"volver.gif\"
onclick=\"javascript:window.close()\"></center> ";

                                }

                                }elseif(mysql_num_rows($DatoGrado)==0){//si no devolvió
ninguna fila
                                mysql_close($linkRegAcad);//cerrar la conexión
con la bd regacad
                                echo "<script>alert(\"Estudiante con No.Carnet
[$NoCarnet] no tiene registro de notas del grado
seleccionado.\");window.close()</script>";
                                }

                                }

```



```

mysql_close($linkRegAcad);//cerrar la conexion con la bd regacad
}
?>
</body>
</html>

```

➤ Generar Boletines.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>[Registro Académico del Colegio La Salle-León]</title>
</head>

<body>
<?php
////////////////////////////////////
////////////////////////////////////OCULTO PARA EL RESTO DE
BOLETINES////////////////////////////////////
$U=$_REQUEST['txtNUmEst'];//cantidad de estudiantes
$annoL=$_REQUEST['txtanyolec'];//anyo lectivo
$Gr=$_REQUEST['txtGrado1'];//grado
$Sec=$_REQUEST['txtseccion'];//seccion
$PG=$_REQUEST['txtProfGuia'];//profesor guía
$V=$_REQUEST['txtNum'];//numero de asignatura materia
$w=$_REQUEST['txtNumotras'];//numero de asignatura otras

require("variables.php");//incluir variables a usar para consultas y conexion a la BD
$link=conectar("localhost","regacad","root","123456");//conexion a la bd regacad

$K=1;
while($K<=$U)
{
    $NombreEstudiante1=$_POST["txtNombre$K"];
    $CarnetEstudiante1=$_POST["txtCarnet$K"];
    if( $CarnetEstudiante1)
    {

echo "<table width='580' height='639' align='center'>
<tr>
<td height='633' bgcolor='#E1E1E1'>
<table >

```




```
$B1=$_POST["txtnombreasig$IIII"];

```

```
$notas1=EjecutarConsulta("select NIP,NIIP,NIIP,NIVP from
calificacion where NoCarnet='$CarnetEstudiante1' and seccion='$Gr$Sec' and
Anyolectivo='$annoL' and codasig=(select codasig from asignatura where
nombreasig='$B1' and grado_ano='$Gr')", $link);

```

```
$row1=mysql_fetch_array($notas1);
echo"<tr>
  <td><font size=2>$B1 </font></td>
  <td align=center><font size=2>$row1[0]</font></td>
  <td align=center><font size=2>$row1[1]</font></td>
  <td align=center><font size=2>$row1[2]</font></td>
  <td align=center><font size=2>$row1[3]</font></td>
</tr>";
  $IIII++;
}

  echo"<tr>
  <td align=\"center\"><font size=2><b>Otros</b></font></td>
  <td align=\"center\"><font size=2><b>1ra</b></font></td>
  <td align=\"center\"><font size=2><b>2da</b></font></td>
  <td align=\"center\"><font size=2><b>3ra</b></font></td>
  <td align=\"center\"><font size=2><b>4ta</b></font></td>
</tr>";

  $IIII=1;
  while($IIII<=$w)
  {
    $B2=$_POST["txtnombreasignatura$IIII"];
    $notas13=EjecutarConsulta("select NIP,NIIP,NIIP,NIVP from calificacion
where NoCarnet='$CarnetEstudiante1' and seccion='$Gr$Sec' and
Anyolectivo='$annoL' and codasig=(select codasig from asignatura where
nombreasig='$B2' and grado_ano='$Gr')", $link);
    $row13=mysql_fetch_array($notas13);
    echo"<tr>
      <td><font size=2>$B2 </font></td>
      <td align=center><font size=2>$row13[0]</font></td>
      <td align=center><font size=2>$row13[1]</font></td>
      <td align=center><font size=2>$row13[2]</font></td>
      <td align=center><font size=2>$row13[3]</font></td>
    </tr>";
    $IIII++;
  }

echo"</table>"

```




```

require("variables.php");//incluir variables a usar para consultas y
conexion a la BD
$link=conectar("localhost","regacad","root","123456");//conexion a la
bd regacad

//consulta para saber la cantidad de alumnos de dicho grado/año
$result1=EjecutarConsulta($SelectGrado_ano1,$link);
$row = mysql_num_rows($result1);

mysql_query("begin",$link);//iniciar la transaccion
while($cont<$row)
{
//recoger valores de ordenar lista estudiantes
$CarnetEst=$_POST["txtNocarnet$cont"];
$AulaEst=$_POST["txtAula$cont"];

//verificar si el estudiante ha sido guardado y así evitar los duplicados en calific.
if(mysql_num_rows($result2=EjecutarConsulta("select nocarnet,seccion
from calificacion where nocarnet='$CarnetEst'
order by nocarnet",$link))>0)
{
$row1=mysql_fetch_array($result2);
echo "<script>alert('\Estudiante $CarnetEst ya Existe. Pertenece a
$row1[1].\');</script>";
mysql_query("rollback",$link);//deshace la transaccion
echo "<script>history.back();</script>";
break;
}

//seleccionar asignatura
$result0=EjecutarConsulta($seleccionar_asignatura,$link);

if(mysql_num_rows($result0)>0)
{
mysql_query("begin");//inicio de la transaccion
while($cod_asignatura=mysql_fetch_array($result0)
{
$resultnombre=EjecutarConsulta("select nombre,apellido from
estudiante where nocarnet='$CarnetEst'", $link);

$nombreapellido=mysql_fetch_array($resultnombre);
//Guardar datos de ordenar alumnos
$result1=EjecutarConsulta("insert into
calificacion(CodAsig,nocarnet,seccion,anyolectivo,NIP,NIIP,NIIP,NIVP,NR,LAF,LA
FR)
value('$cod_asignatura[0]','$CarnetEst','$grañ$AulaEst','$var_lectivo1','0','0','0','0','
0','"')",$link);

```



```

        if(!$result1)
        {
            mysql_query("rollback");//Deshacer los cambio de la
transaccion
            echo"Error al Insertar en Calificacion";
            exit;
        }

        $result1historial=EjecutarConsulta("insert into
historial(nocarnet,nombreE,profguia,asignatura,anyo_lectivo,seccion,NIP,NIIP,NIII
P,NIVP,NReparacion,LAFFinal,LAFReparacion)
value('$CarnetEst','$nombreaapellido[0]".
".".$nombreaapellido[1]',"$cod_asignatura[1]','$var_lectivo1','$grañ$AulaEst','0','0','0
','0','0','")",$link);

        if(!$result1historial)
        {
            mysql_query("rollback");//Deshacer los cambio de la
transaccion
            echo"Error al Insertar en Historial";
            exit;
        }
    }
}
else
{
    echo "<script>alert(\"No Hay Asignatura Registrada Para Este
Grado.\");</script>";
    echo "<script> history.back()</script>";
}
    $cont++;
}
mysql_query("commit",$link);//finalizar la transaccion
mysql_close($link);
echo "<script>alert(\"Los Datos Fueron Guardados\");</script>";

echo "<script> history.back()</script>";
echo"</body>
</html>";
}

if($editar)//si se pulso el boton Editar Registro
{
    echo"<html>
<head>
<title>Editar Datos de Estudiantes por Aula</title>
<style type=\"text/css\">

```



```

<!--
body {
    background-color: #99cccc;
}
-->
</style>

<script>
function ValidarCarnet(VC)
{
    if(VC.ctcarnet.value=="")
    {
        alert("El Carnet del Estudiante es requerido.");
        return false;
    }

    if(VC.ctcarnet.value.search(/([d]{4})-([d]{4})/)==-1)//formato del
carnet
    {
        alert("Número de Carnet no válido.");
        VC.ctcarnet.select()
        return false;
    }
}
</script>
</head>
<body>;
    echo "<table width=\"80%\" align=center><tr><td><div
align=\"left\"><img src=\"SALLEColor.png\" width=80
height=77></div></td><td><div align=left><h1><strong><i>Editar Estudiante de la
Lista Ordenada</i></strong></h1></div></td></tr></table><br><br>";

    echo"
<br>
<hr width=100%></hr>
    <form action=\"OrdenarGuardarEditado.php\" onsubmit=\"return
ValidarCarnet(this)\" method=\"post\" name=\"frmGrado\">
        <center> <table border=1>
        <tr>
        <td>
            Introduzca El Carnet:&nbsp;<input name=\"ctcarnet\" type=\"text\" id=\"ctcarnet\"
size=\"9\" maxlength=\"9\">
            </td>
            <td>Seccion:&nbsp;<select name=txtAula>
            <option selected>A</option>
            <option>B</option>
            <option>C</option>

```



```

<option>D</option>
</select></strong></td>
</tr>
</table>
<br>
<input name=\"btguardar\" type=\"submit\" value=\"Guardar\"/>
<input name=\"ctga1\" type=\"hidden\" value=\"\".$var_grado_ano1.\"/>
</center>
</form>
";
echo "<center><input type=\"image\" name=\"imgVolvermod\"
src=\"volver.gif\" onclick=\"javascript:window.history.back()\"
alt=\"Regresar\"/></center>";
echo"</body>
</html>";
}
?>

```

6.6 Pruebas.

Realizamos varias pruebas con la aplicación que desarrollamos, alojándola en el servidor del Colegio La Salle, con el fin de acceder a la aplicación desde varios equipos clientes conectados a la misma red local o intranet instalada en dicho Colegio. Al alojar la aplicación en el servidor vimos la necesidad de instalar y configurar el servicio de paginas web de Windows Server 2003 (Internet Information Server 6.0), de igual manera lo hicimos para el servicio de Sistema de Nombre de Dominio (DNS), y por ultimo el sistema gestor de Bases de Datos (MySql Server 5.0).

Una vez instalada y configurada la aplicación, ésta fue probada por el Personal Docente y Director del Colegio Durante una Semana. Luego de este tiempo los resultados esperados fueron satisfactorios puesto que se ingresaron datos y obtuvieron informes sin ningún problema, garantizando con esto el correcto funcionamiento y cumplimiento con la aplicación desarrollada valiéndonos indispensablemente de normas de gestión de calidad (ISO 9001).



6.7 Mantenimiento.

Uno de los miembros de nuestro grupo realizará una visita al Colegio La Salle durante cierto periodo de tiempo con el objetivo de ver si el software presenta alguna anomalía. Se garantizará que si ocurre algún problema con el software y si se le realiza cambios, éstos se efectúen correctamente, documentándolos e informando a los demás miembros sobre dichos cambios.



7. CONCLUSIONES.

- Con el desarrollo de este software el Colegio La Salle de León se ha puesto a la vanguardia de otros Colegio del país.
- El Colegio podrá brindar mejor servicio a todos los estudiantes del mismo.
- La secretaria académica del Colegio tiene mayor facilidad del manejo de notas, puesto que el sistema genera los certificados de notas de manera automatizada.
- Al concluir con nuestro trabajo-tesis consideramos que hemos cumplido con los objetivos planteados, ya que hemos seguido paso a paso lo que dice la Ingeniería del Software.
- Al llevar a cabo el desarrollo de nuestro Sistema y haber utilizado **Windows Server 2003** (Sistema Operativo), **PHP** (Lenguaje de Programación), **MySql Server 5.0** (Sistema Gestor de Base de Datos), **IIS** (Servidor), **Macromedia Dreamweaver 8** (Editor de código php), concluimos que son herramientas potentes que permiten crear de forma sencilla y rápida sistemas de cualquier índole.
- Gracias a la utilización de estas herramientas (**PHP, MySql Server 5.0, IIS, Macromedia Dreamweaver 8**) se concluyó con el desarrollo de la aplicación para la Automatización del Registro Académico del Colegio La Salle – León lo cual a sido de gran ayuda para enriquecer nuestros conocimientos y darnos a conocer como Ingenieros en Sistema que somos.



8. RECOMENDACIONES.

Al concluir nuestro trabajo monográfico recomendamos:

- Los usuarios del sistema deberán tener un grado mínimo de conocimiento de informática.
- Dar soporte técnico al Servidor y a la Intranet del Colegio para proteger y mantener la integridad de los datos.
- Las claves de acceso deben ser solamente del dominio de la persona o personas que interactúen con la aplicación.
- Un administrador de la aplicación debe realizar respaldos o copias de seguridad periódicamente de la base de datos con la que trabaja la misma, y guardarla en un dispositivo de almacenamiento extraíble para uso futuro.



9. BIBLIOGRAFÍA

- <http://es.wikipedia.org/wiki/MySql>
- <http://www.milonic.com>
- <http://www.nvu.com>
- <http://www.php.net>
- <http://www.desarrolloweb.com>
- http://es.wikipedia.org/wiki/Internet_Information_Server
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=lamp>
- <http://dev.mysql.com>
- <http://www.tufuncion.com/ide-php>



10. ANEXOS

10.1 Glosario.

A

- **Active x:** (control). Componente que se puede insertar en una página Web para proporcionar una funcionalidad que no está directamente disponible en HTML, como secuencias de animación. Los controles ActiveX se pueden implementar en diferentes lenguajes de programación. Pequeños programas que permiten mostrar páginas web dinámicas en el PC y que suplen las limitaciones que, al respecto, tiene el lenguaje HTML. Los controles ActiveX tienen que descargarse al disco duro del ordenador para que los documentos que los utilizan puedan visualizarse.
- **Apache:** Servidor web de código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el NCSA. La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «a patchy server» -un servidor de remiendos-, es decir un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «kill app» del software libre en el ámbito de los servidores y el ejemplo de software libre de mayor éxito, por delante incluso del kernel Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache.
- **ASCII:** American Standard Code for Information Interchange. Uno de los sistemas de codificación más utilizados y de uso interno del ordenador. Cuando pulsamos una tecla se genera un código, por ejemplo la "A" es para el ordenador el 65 en código ASCII. El utilizado en la actualidad, el ampliado, llega hasta el código 255 a diferencia del original que alcanza la mitad, dejando sin representación caracteres como las eñes o vocales acentuadas.

B

- **Base de datos:** Conjunto de información para varios usuarios. Suele admitir la selección de accesos aleatorios y múltiples "vistas" o niveles de abstracción de los datos subyacentes.
- **Browser:** Un software de cliente que es empleado para aprovechar diversos recursos de Internet. Es comúnmente llamado **Navegador**.



C

- **CGI:** (Interfaz de gateway común): Interfaz para programadores que crean archivos de comandos o aplicaciones que se ejecutan internamente en un servidor de Web. Estos archivos de comandos pueden generar texto y otros tipos de datos de forma inmediata, en respuesta a una entrada del usuario, o bien tomando la información de una base de datos.
- **Cliente:** Usuario de un servicio de la red. También se utiliza para designar aquel ordenador que depende de otro para alguno de (o todos) sus servicios).
- **Comando:** Viene de command, en inglés, en castellano debería de traducirse por mandato. En cualquier caso es una orden que se da, generalmente al sistema operativo de la computadora, para que realice la tarea que esa orden conlleva.

D

- **DHCP:** Son las siglas en inglés de Protocolo de configuración dinámica de servidores (). Es un protocolo de red en el que un servidor provee los parámetros de configuración a las computadoras conectadas a la red informática que los requieran (máscara, puerta de enlace y otros) y también incluye un mecanismo de asignación de direcciones de IP.
- **Dirección:** Código exclusivo asignado a la ubicación de un archivo almacenado, un dispositivo en un sistema o red, o cualquier origen de datos de una red.
- **Dirección IP:** Dirección de 32 bits del protocolo Internet asignada a un host. La dirección IP tiene un componente del host y un componente de la red.
- **Dirección URL:** (Uniform Resource Locator): Formato de las direcciones de sitios que muestra el nombre del servidor en el que se almacenan los archivos del sitio, la ruta de acceso al directorio del archivo y su nombre.
- **Directorio:** Parte de la estructura para organizar la información en los soportes de almacenamiento. En Sistemas Operativos tipo "árbol", como Unix o MS-DOS, un directorio es un tipo especial de fichero, con capacidad o programado para poder contener otros ficheros. Están divididos en el directorio raíz, que es el inicial y único (se define como \ o / dependiendo del Sistema) y las "ramas son los subdirectorios". El directorio actual es en el que nos encontramos, es decir, el activo.
- **DNS: (Domain Name System)** Método usado para convertir nombres Internet en sus correspondientes números Internet.
- **Dominio: (domain)** Un componente en la jerarquía de nombres. Sintácticamente, un dominio consiste en una secuencia de nombres u otras palabras separadas por puntos.
- **DOWNLOAD:** transferencia de datos de archivos, información o programas de Internet, de un equipo central al del usuario.



E

- **E-MAIL:** abreviatura de correo electrónico, el servicio más usado de la Red, que transmite mensajes instantáneos (textos o imágenes) a cualquier lugar del planeta.
- **Explorador:** Programa de aplicación que proporciona una interfaz gráfica interactiva para buscar localizar, ver y administrar la información a través de una red.

F

- **FICHERO:** Sinónimo de archivo y esencia del almacenamiento informático. Delimitado por una cabecera y una marca de final, lugar donde puede estar ubicada cualquier tipo de información, ya sea texto, programas, imágenes, sonidos, etc.
- **FTP:** (File Transfer Protocol) Protocolo utilizado para transferir archivos a través de una amplia variedad de sistemas.

G

- **GOPHER:** Es una herramienta creada para encontrar la información a los recursos que se necesitan. Se ideó en la Universidad de Minnesota y le pusieron el nombre de Gopher, que es el nombre de una especie de ratoncito americano. Funciona presentado en pantalla un menú de opciones, cuyos títulos dan una idea clara del tema o tipos de recursos a los que hacer referencia. Cuando se accede a un servidor Gopher, aparecen una serie de directorios temáticos con títulos indicadores que nos facilitan dónde debemos dirigirnos para encontrar la información que queremos localizar. Tiene incorporadas algunas herramientas de búsqueda, como Verónica.

H

- **Hipertexto:** información específica de una página a la que se accede mediante un link. Haciendo un clic sobre el lugar destacado en la pantalla, aparece toda la información por encima de la página principal a la que uno había entrado.
- **Hipervínculo:** Conexiones entre una información y otra.
- **HOME O HOMEPAGE:** la página por la cual se accede a un sitio.
- **HOST:** Es el nombre genérico dado a una computadora que esta conectada a Internet y cumple la función de Servidor. Permite que su información sea accedida por otras maquinas, a través de un numero IP o un nombre.
- **HTML** (Hypertext Markup Language): Lenguaje de "etiquetas" en el que se asigna formato a las páginas de Web y se distribuye la información.



- **HTTP:** (Protocolo de transferencia de hipertexto): Método mediante el que se transfieren documentos desde el sistema host o servidor a los exploradores y usuarios individuales.

I

- **ICONO:** Representación gráfica de un elemento, generalmente una opción a elegir, que sustituye o complementa al texto escrito. Su característica es que ha de ser lo suficientemente significativo, y se emplean bien ficheros .ico o .bmp normalmente de 16x16 o de 32x32 bits.
- **IETF:** El IETF (Internet Engineering Task Force, en castellano Grupo de Trabajo en Ingeniería de Internet) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad. Fue creada en EE.UU. en 1986.
- **INTERNET:** red global que conecta miles de millones de de computadoras con direcciones únicas para que intercambien datos. Se estima que hay en el mundo cerca de 306 millones de usuarios.
- **IP:** (Protocolo Internet) Define la unidad de información enviada entre sistemas, que proporciona un servicio de entrega de paquetes básico.

J

- **JSP:** (*Java Server Pages*) Páginas activas de SUN, se basan en un lenguaje de scripting en el lado del servidor que se comunica con clases Java, objetos RMI, CORBA, etc. La metodología de trabajo esperada es la misma que con Visual Basic con la diferencia de que esta vez se trata de una plataforma mucho más abierta.

L

- **Link:** Refiriéndonos a Internet y páginas Web es un unión entre varios documentos dentro de un mismo servidor, o con mayor frecuencia, la posibilidad de acceder mediante la pulsación de una palabra o frase, generalmente resaltada y subrayada, a otra página situada en un ordenador distinto y ubicado en cualquier lugar del mundo, ya que en el momento de la creación de ese enlace se le ha asignado

M

- **Macro:** En el entorno Microsoft son una serie de instrucciones que se ejecutan de forma automática respondiendo a determinados sucesos. Es una forma de automatización de procesos cómoda para el usuario, pero como contrapartida, es una fuente de introducción de código no deseado, tipo virus.
- **MODEM:** dispositivo que permite a una PC transmitir datos por vía telefónica.



N

- **Navegador:** En el ámbito de Internet, es un software capaz de visualizar la información en formato de Web, es decir, pueden utilizar hipertexto y los protocolos propios de Internet, de manera que no sólo se ciñen a páginas en html, sino que igual manejan FTP, que SMTP, grupos de noticias, ayudados por otros software pueden manejar cualquier tipo de multimedia, etc.
- **NNTP:** (Network News Transfer Protocol). Protocolo empleado en Internet para transferir artículos en los grupos de noticias o News.

O

- **ON LINE / OFF LINE:** Conectado o desconectado a Internet.
- **OSI:** Open Systems Interconnection. Sistema creado por ISO en el que se especifica estándares de funcionamiento de cada una de las partes o capas en las que puede constar una arquitectura de red. Las divide en siete niveles denominados: físico, de enlace, de red, de transporte, de sesión, de presentación y de aplicación. Son numerados del 1 al 7 y las referencias se suelen hacer sobre el número, por ej. el nivel 3 OSI se entiende como el de red.

P

- **Página Web:** documento publicado en la Web.
- **Perl:** Lenguaje de programación muy utilizado para la elaboración de aplicaciones CGI. Es multiplataforma y funciona bajo UNIX.
- **PHP:** Personal Home Page. Página de inicio personal. Esta era la denominación del primer PHP realizado por Rasmus Lerdorf. En este momento estamos en la versión 4 de este lenguaje de programación destinado a la Red, y que una vez interpretado por el servidor web genera código HTML.
- **Plataforma:** Es un término de carácter genérico que designa normalmente una arquitectura de hardware, aunque también se usa a veces para sistemas operativos o para el conjunto de ambos. Los ordenadores VAX de la firma Digital, por ejemplo, serían una plataforma en la que se pueden soportar aplicaciones que, a su vez, corren (Ver: Correr) en otras plataformas.
- **Plug-and-Play:** Tecnología que permite la auto detección de dispositivos tales como tarjetas de expansión por parte del ordenador, con objeto de facilitar su instalación.
- **Protocolo:** Descripción formal del formato de los mensajes y de las reglas que dos ordenadores deben seguir para intercambiar mensajes.



Q

- **QoS:** La **QoS** (Quality of Service, Calidad de Servicio) garantiza que se transmitirá cierta cantidad de datos en un tiempo dado.

R

- **RAID:** (Redundant Array Of Independen/ Inexpensive Disks) es un término inglés que hace referencia a un conjunto de discos redundantes independientes/baratos. Este tipo de dispositivos se utilizan para aumentar la integridad de los datos en los discos, mejorar la tolerancia a los fallos y errores y mejorar el rendimiento. En general permiten proveer discos virtuales de un tamaño mucho mayor al de los discos comúnmente disponibles.
- **Router:** Un ordenador dedicado u otro dispositivo que envía paquetes desde un lugar a otro, teniendo en cuenta el estado actual de la red.

S

- **Seguridad:** Mecanismos de control que evitan el uso no autorizado de recursos.
- **Servidor:** En una red, estación host de datos que proporciona servicios a otras estaciones.
- **Sitio:** espacio en la Web con diversidad de contenidos y que puede tener varias páginas, además de su Homepage.
- **Smiley:** O Emoticon. Son pequeños dibujos formados con unos pocos caracteres, que se han hecho muy populares y casi imprescindibles en el lenguaje de Internet, para expresar distintos estados de ánimo o poner énfasis especial en alguna frase concreta.
- **Soporte:** Formato de distribución y almacenamiento de información (p. ej. cinta de vídeo, disquete, disco óptico, impresora, etc.). Una ampliación de la capacidad de comunicación de la humanidad. Es el mensaje.

T

- **TCO (TCP):** Comúnmente conocido como TCO por sus siglas en inglés (**Total Cost of ownership, Costo Total de Propiedad - CTP**) es una medida diseñada por el **Grupo Gartner** a finales de los 70's y ampliamente difundida a principios de los 80's para evaluar el costo total en que se incurre al adquirir, inicialmente un sistema de información, y ahora se utiliza su concepto para todas las compras.
- **TCP/IP:** Protocolo de control de transmisiones/Protocolo Internet. Es el protocolo estándar de comunicaciones en red utilizado para conectar sistemas informáticos a través de Internet.



- **Telnet:** Programa de red que ofrece una forma de conectarse y trabajar desde otro equipo. Al conectarse a otro sistema, los usuarios pueden tener acceso a servicios de Internet que quizás no tengan en sus propios equipos.
- **Tiempo real:** Rápida transmisión y proceso de datos orientados a eventos y transacciones a medida que se producen, en contraposición a almacenarse y retransmitirse o procesarse por lotes.

U

- **UNIX:** Un sistema operativo diseñado para ser usado por un grupo de varias personal al mismo tiempo (multi-usuario) que maneja **TPC/IP**. Es el sistema operativo más común en los servidores **Internet**.
- **URL:** La manera estándar de asignar direcciones de cualquier recurso en Internet que forma parte del **WWW**.
- **Usenet:** (USEr NETwork) Grupos de debate de Internet. Uno de los primeros formatos de "correo electrónico colectivo". Actualmente hay unos 10000 grupos de debate diferentes.

V

- **VPN:** acrónimo de Virtual Private Network, que en castellano significa Red Privada Virtual (RPV), es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.# .En estadística se conoce como VPN al valor predictivo negativo de una prueba diagnóstica.



10.2 Constancia de satisfacción del Cliente (Colegio La Salle - León).



COLEGIO LA SALLE

APDO. 110 LEON - NICARAGUA

TELS. 311-2626, 2792, 7291 FAX 311-5443
E-MAIL: clsleon@ibw.com.ni

León Nicaragua 31 de Julio de 2008

A QUIEN INTERESE:

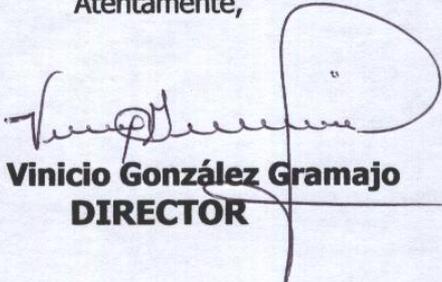
El Director del Colegio La Salle de la ciudad de León Nicaragua, por medio de la presente HACE CONSTAR QUE:

**Deybin Bayardo Chavarria Estrada
Douglas Guillermo Molina Castro
Narciso Antonio Bonilla Torrez
Wilton Alfredo Guardado Castillo**

realizaron su trabajo monográfico de tesis titulado "AUTOMATIZACIÓN DEL REGISTRO ACADÉMICO DEL COLEGIO LA SALLE" en las fechas comprendidas entre el 16 de Agosto del año dos mil siete al treinta y uno de julio del año dos mil ocho.

Extiendo firma y sello la presente constancia el hoja de papel membretado de la institución el treinta y uno de julio del año dos mil ocho.

Atentamente,


**Hno. Vinicio González Gramajo
DIRECTOR**





10.3 Informes que brinda la Aplicación.

- **Matrícula.**



COLEGIO LA SALLE

LEÓN, NICARAGUA

CONTRATO DE INSCRIPCIÓN

Grado: 1 2 3 4 5 6 7 8 9 10 11 Carnet:

DATOS DEL ALUMNO:

Nombres y Apellidos:
 Lugar de Nacimiento: Fecha de Nacimiento.:
 Sexo: M F Religión: Sacramentos: B PC CONF
 Centro Educativo Anterior:
 ¿Actualmente, tiene hermanos estudiando en este colegio? NO SI [Especifique al reverso]

DATOS DE LOS PADRES :

Nombre del padre: Responsable:
 Ocupación: Teléf. de Oficina:
 Nombre de la madre: Responsable:
 Ocupación: Teléf. de Oficina:
 Estado Civil de los padres: Casados Unidos Divorciados Soltero(a) Viudo(a)
 Dirección de casa: Tel.:

ACEPTACIÓN DEL CONTRATO: Por este medio, CERTIFICAMOS ACEPTAR:

- El Ideario Lasallista (Filosofía del centro).
- Los Reglamentos del Colegio.
- Notificar en Secretaría en el caso de retiro del Alumno(a) del Centro.
- Cancelar las cuotas en las fechas establecidas y otros pagos del ciclo escolar.

(f) _____ Vº, Bº, (f) _____
 Firma de uno de los Padres y/o Responsable Legal Director
 León, de del

DATOS FAMILIARES:

1. ¿El padre vive en el hogar? SI NO Causa de la no presencia:
 2. ¿La madre vive en el hogar? SI NO Causa de la no presencia:

DATOS DEL ALUMNO:

1. ¿Tiene su hijo(a) problemas de: (Marcar en la Casilla)

<input type="checkbox"/> Anemia	<input type="checkbox"/> Epilepsia	<input type="checkbox"/> Asma	<input type="checkbox"/> Audición
<input type="checkbox"/> Visión	<input type="checkbox"/> Nerviosismo	<input type="checkbox"/> Problema de Lenguaje	<input type="checkbox"/> Dolores de Cabeza
<input type="checkbox"/> Mareos Frecuentes	<input type="checkbox"/> Desmayos	<input checked="" type="checkbox"/> Otros	



2. ¿Ha estado hospitalizado? SI NO ¿Cuánto tiempo?
3. ¿Consume algún medicamento? SI NO ¿Cuál?
4. ¿Está en tratamiento médico? SI NO ¿Dónde?
5. ¿Tiene seguro médico? SI NO

DESENVOLVIMIENTO DEL ALUMNO(A):

1. ¿El o la joven se comporta con los demás? Receloso Comunicativo Impulsivo Cooperador Otros
2. Prefiere jugar con: Jóvenes de la misma edad Menores que él o ella
3. Sus actividades preferidas en orden de importancia: (Enumere 3)
4. Disposición usual del carácter: Alegre Agresivo Comunicativo Irritable Otros
5. Se relaciona mejor con: Padre Madre Hermanos Otros
6. ¿Tiene dificultad al expresarse? SI NO ¿En qué consiste?
7. ¿Tiende a atraer la atención de las personas? SI NO ¿En qué forma?

Hermanos en el Colegio

GRADO

<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

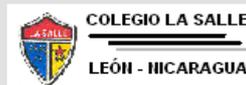
USO EXCLUSIVO DE LA DIRECCIÓN

<input type="text"/>



- **Boletín Final**

La secretaria del Colegio La Salle de León, Nicaragua hace constar que de acuerdo al plan de estudios del grado correspondiente, las calificaciones obtenidas por el aprendiente son las siguientes:



Carnet: 0000-0000

Nombre y Apellidos: MELISSA MARCELA ABURTO MARTINEZ

Grado: 6to. Grado

Sección: A

Año Lectivo: 2013

Materia	1ra	2da	3ra	4ta	Final	Clasificación
Educacion En La Fe	85	87	87	89	87	Aprobado
Lengua Y Literatura	0	0	0	0	0	Reprobado
Matematica	0	0	0	0	0	Reprobado
Ciencias De La Vida Y Del Ambiente	85	87	98	89	90	Aprobado
Medio Social Y Espacio Geografico	0	0	0	0	0	Reprobado
Convivencia Y Civismo	89	89	98	87	91	Aprobado
Educacion Fisica, Recreacion Y Deportes	85	74	87	74	80	Aprobado
Orientacion Tecnica	0	0	0	0	0	Reprobado
Productividad	0	0	0	0	0	Reprobado
Lengua Extranjera	0	0	0	0	0	Reprobado
Expresion Cultural Y Artistica	90	0	0	0	23	Reprobado
Comportamiento	89	75	89	58	78	Aprobado
Otros	1ra	2da	3ra	4ta	Final	Clasificación
Ausencias Justificadas	80	89	85	98	352	
Ausencias Injustificadas	60	98	95	92	345	
Compromisos De Los PP. FF.	89	98	98	89	94	Aprobado

Profesor Guía: Douglas Guillermo Molina Castro.

Area de Informática

Fue promovido Felicitaciones

Debe presentarse a recuperación

Se le permitirá continuar en el Colegio, condicionado

Por Comportamiento

Por Rendimiento

ESCALA DE EQUIVALENCIAS

90 - 100 = Sobresaliente

80 - 89 = Muy Bueno

70 - 79 = Bueno

60 - 69 = Regular

CALIFICACIÓN PARA APROBAR = 60