DISEÑO E IMPLEMANETACION DE UN SISTEMA DE INFORMACIÓN ACADEMICO-ADMINISTRATIVO PARA LA ESCUELA ANA BOLIVAR DE CONSUEGRA Y EL COLEGIO JOSE CONSUEGRA HIGGINS DEL CORREGIMIENTO DE ISABEL LOPEZ (Atlántico).

Programa:
FORMATIVA IV.

Estudiantes:
Juan Carlos Casas
Ricardo Cortés
Osvaldo Villalba

Profesor:
RICARDO MARIN

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO
SIMÓN BOLÍVAR
PROGRAMA DE INGENIERÍA DE SISTEMAS
X SEMESTRE
BARRANQUILLA
2005

TABLA DEL CONTENIDO

1. PLANTEAMIENTO DEL PROBLEMA
1.1 Descripción del problema.
1.2 Formulación del problema.
1.3 Sistematización del problema.

2. OBJETIVOS.
2.1 Objetivos generales.
2.2 Objetivos específicos.

3. JUSTIFICACION
3.1 Importancia teórica.
3.2 Importancia practica.

4. MARCO DE REFERENCIA
4.1 Marco teórico.
4.2 Marco conceptual.
4.3 Marco legal.

5. METODOLOGIA
5.1 Metodología de investigación.
5.2 Tipos de investigación.
5.3 Técnicas para la recolección de investigación.
5.4 Población.
5.5 Muestra.

6. INGENIERIA DE REQUISITOS
6.1 Descripción del sistema actual.
6.2 Diagrama de flujo del sistema actual(process modeller).
6.3 Identificación de requisitos.
6.4 Análisis de los requisitos.
6.5 Especificación de los requisitos.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

7. INGENIERIA DE INFORMACION
7.1 Misión.
7.2 Visión.
7.3 Historia.
7.4 Política.
7.5 Organigrama.

8. ESPECIFICACION DE LAS ENTIDADES
8.1 Diagrama de flujo de datos.
8.2 Modelo entidad-relación.
8.3 Modelo relacional.
8.4 Diccionario de datos.

9. DISEÑO DEL SISTEMA
9.1 Estructura Funcional (Diagrama jerárquico de funciones).
9.2 Diseño de las interfaces.

10. DESARROLLO Y PRUEBA DEL SISTEMA.

11. RECOMENDACIONES.

CONCLUSION
BIBLIOGRAFIA
ANEXOS

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

1. PLANTEAMIENTO DEL PROBLEMA

1. 1 DESCRIPCION DEL PROBLEMA.

El colegio José Consuegra Higgins es una institución que se dedica a la educación básica secundaria, la cual fue fundado el 1° de Junio de 1981 en un local alquilado por los propietarios de dicha institución, en el año de 1986 se comenzó la construcción de plantel educativo, en 1992 complementaron la formación de educación básica formal fundando así la escuela Ana Bolívar de Consuegra la cual se dedica a la enseñanza básica primaria por lo que hoy en día funcionan en dicha construcción las dos instituciones educativas.

La escuela Ana Bolívar de Consuegra y el Colegio José Consuegra Higgins son instituciones educativas sin animo de lucro, privadas y mixtas, que constan de dos jornadas, la primera es la de la mañana que es la que se encarga de la educación básica primaria y la segunda es de la jornada de la tarde que es la que se encarga de la educación básica secundaria. Esta institución educativa en general está dedicada a ofrecer sus servicios a la comunidad de Isabel López y a sus corregimientos aledaños.

Actualmente en la institución en general se manejan dos procesos de notas y de matricula. En el proceso de matricula el cliente o acudiente solicita al secretario encargado ya sea de la jornada de la mañana o de la tarde los requisitos necesarios para matricular al alumno, luego que el cliente tiene los requisitos al día se los entrega a la secretaria esta le hace entrega de un volante de pago de matricula, después de que el acudiente cancelo el volante se procede a al llenado del libro de matricula en el cual se registran

Los datos del acudiente y del alumno que desea ingresar a la institución, siguiendo los requisitos exigidos por la institución, al momento de estar matriculado el alumno, éste recibirá mensualmente un volante de pago en el cual tendrá que cancelar para asistir a clases.

El segundo proceso, el académico viene siendo el registro de notas, con respecto a este proceso el docente realiza un informe de cada periodo académico, que se basa en logros luego este es entregado al área administrativa (secretaria) encargada de la jornada respectiva, ésta organiza la información de cada alumno para que en la entrega de boletines reciban sus respectivas notas (logros) de cada periodo, todos estos procesos mencionados anteriormente se maneja manualmente. Una vez terminado cada uno de estos procesos debe quedar una copia de todos los recibos de los estudiantes para archivarlos alfabéticamente en los cuales se registran la fecha inicial del proceso.

Actualmente la institución presenta problemas en la realización de los procesos tales como la demora en el llenado de los recibos y evaluación de informes de las notas, además de esto se presenta problemas como la perdida, deterioro y descontrol de los documentos. De acuerdo a la observación que se realizó en dicha institución encontramos falencias en el manejo de los archivos reflejando así una deficiencia del control de sus procesos.

El personal administrativo de la institución considera que el origen de las causas anteriormente mencionadas, se deben a varios factores tales como la falta de una herramienta que facilite y agilice el proceso de control de los datos ya que no se está brindando un adecuado proceso para obtener los datos del estudiante. Teniendo en cuenta las dificultades que se presentan

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

en la institución, puede ser útil la elaboración de un sistema de información que maneje un módulo de notas y módulo de matriculas, para controlar la elaboración de estos procesos en cuanto a matriculas y registro de notas para así obtener un mejor almacenamiento, rapidez y ordenamiento para el manejo de los datos de la institución educativa, brindando así un beneficio tanto para ella como para los estudiantes.

1. 2 FORMULACION DEL PROBLEMA.

¿De que manera se podrá mejorar los procesos Académico-administrativos en la escuela Ana Bolívar de Consuegra y Colegio José Consuegra Higgins del Corregimiento Isabel López (Atlántico)?

1.3 SISTEMATIZACION

- ¿Cómo facilitar los procesos académicos-administrativos que se llevan a cabo en la Institución?

- ¿Cuales son las fortalezas y debilidades de los procedimientos que se realizan en cada uno de los procesos académicos?

- ¿Cual es la información pertinente a cada uno de los procesos Académicos-Administrativos?

- ¿Cuales son las diferentes herramientas que utilizan otros planteles educativos a nivel distrital con respecto a los procesos académicos?

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e Implementar un sistema de información que permita optimizar los procesos Académicos-Administrativos en la escuela Ana Bolívar de Consuegra y el colegio José Consuegra Higgins del corregimiento de Isabel López (Atlántico), con el fin de mejorar el control y manejo de los registros.

2.2 OBJETIVOS ESPECIFICOS

- Diseñar un modulo de clasificación de la información pertinentes a cada uno de los procesos académicos.
- Generar reportes de las notas de cada periodo de los estudiantes.
- Desarrollar una interfaz que le permita a la institución educativa acceder a todos los procedimiento académicos y manejarlo de forma fácil y sencilla.
- Crear ayudas en línea para resolver los problemas que se le pueden presentar al usuario del sistema.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

- Crear un generador de reportes dinámicos.

- Crear un modulo de importación y exportación de datos de la Base de Datos.

- Llevar a cabo un manual de usuario y del sistema.

3. JUSTIFICACIÓN

3.1 IMPORTANCIA TEORICA.

El siguiente proyecto tiene como finalidad ampliar y aplicar los conocimientos de cómo se manejan y utilizan los sistemas de información, y con la implementación de este se podrá lograr tener mayor conocimiento de las herramientas que se aprenden durante la carera y otras que son desconocidas aún.

Debido al estudio realizado en colegio Ana Consuegra de Bolívar y Colegio José Consuegra Higgins del Corregimiento Isabel López (Atlántico) es importante y necesario que se implemente el sistema de información porque les facilitara el manejo y control de los procesos académico-administrativo además que será una ventaja competitiva y ayudará a darle una pronta solución a la problemática se vienen presentando

3.2 IMPORTANCIA PRÁCTICA.

Hoy en día una de las mejores herramientas de trabajo es el software ya que en este se puede almacenar gran cantidad de información y se maneja de manera fácil, rápida y segura. Este sistema de información que se desarrollará manejará todos los procesos referidos a la parte académicos-administrativos que será una forma fácil de realizar las listas de evaluación y los boletines de matriculas además se tendrá una mejor seguridad de los datos

El sistema mejorará la forma como opera este departamento sobre la información de una manera eficaz. Esta actividad es de suma importancia para la institución educativa debido a que la herramienta le permitirá al plantel educativo agilizar el desarrollo de los procesos académicos que se llevan a cabo manualmente en la institución, minimizando así el tiempo empleado para desarrollarlos.

La elaboración de este proyecto permitirá poner en practica los conocimientos para el desarrollo y construcción de un software en particular, software Académico-Administrativo, además de posibilitar la adquisición de nuevos conocimientos. La investigación propuesta pretende conseguir el desarrollo de un herramienta que sea de utilidad o de ayuda al usuario, a realizar de manera fácil y eficaz los procesos de matricula y registro de notas, y además que lo guié y controle en sus diferentes aplicaciones. Una herramienta con estas características permite a la institución utilizar todos los procedimientos que se llevan a cabo en el plantel educativo.

4. MARCO DE REFERENCIA

4.1 MARCO TEORICO.

En los últimos años los Sistemas de Información se han desarrollado de una manera inexplicable, influyendo de alguna forma con la sociedad. Nosotros antes de analizar o dar un concepto de lo que es un Sistema de información debemos conocer el significado de estas dos palabras.

Los sistemas son una serie de elementos que funcionan en conjunto para alcanzar un objetivo o varios objetivos en común, y la información que son todos aquellos datos que adquieren un significado dentro de un contexto. Pero también al hablar de información no podemos pasar por alto los datos que son pieza fundamental en los Sistemas de Información porque si no tenemos datos, no tendríamos información.

Para trabajar en un Sistema de información se tiene que tener muy en cuenta la investigación y sus derivados, por que en el proceso de investigación se basan todos los Proyectos, ya que sin una buena investigación un proyecto no va a ser de buena calidad. La Investigación como proceso tiene varias etapas sucesivas algunas de estas son:

1. Construcción del problema.
2. Diseño metodológico.
3. Trabajo de campo: recolección, registro y almacenamiento de datos.
4. Organización, análisis e interpretación de la información.
5. Comunicación de los hallazgos.

1. Construcción del problema de investigación

Esta primera etapa del proceso de investigación inicia cuando un actor o actores se proponen, explicita y sistemáticamente, construir conocimiento científico. Ocurre ante una situación percibida como problemática en algún nivel del sistema educativo. Lo primero que debe hacer el investigador es iniciar una búsqueda bibliográfica para enriquecer la primera representación de su objeto. La construcción de un problema de investigación es el proceso de delimitación de una situación problemática desde un punto de vista teórico, social y temporal o histórico.

2. Diseño auxiliar de la investigación.

Esta segunda etapa comienza con los tipos de diseño, este va depender de dos factores, En primer lugar del problema construido, es decir, dependerá de los objetivos específicos de la investigación (hipótesis). Cuando las hipótesis apunten a describir o caracterizar un problema, el diseño resultante será de tipo descriptivo; cuando se trata de identificar y seleccionar qué aspectos empíricos del problema resultan más relevantes desde un punto de vista más bien teórico o se trata de contrastar qué teorías explican mejor un fenómeno, se empleará un diseño exploratorio pero cuando el objetivo sea probar que un fenómeno se comprota tal como lo predecía una teoría, el diseño será de tipo demostrativo.

En segundo lugar, el diseño va a depender también de las restricciones externas: el presupuesto, el tiempo, el momento social, político y cultural de la investigación así como de los recursos: humanos, materiales, soporte tecnológico. Finalmente hay que decir que un diseño, relaciona un nivel de análisis (teórico) con un nivel de observación (empírico).

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

Un paso fundamentalmente importante es determinar a qué nivel de análisis se refiere el problema y qué unidades de observación serán tenidas en cuenta en la investigación. Estos niveles pueden ser: las interacciones, los grupos, los roles de papeles o funcionales, las organizaciones, los sistemas sociales, estructuras sociales, sociedades.

3. Trabajo de campo: recolección, registro y almacenamiento de datos.

Esta etapa comienza con las técnicas de recolección de información La cuestión de fondo es definir de qué manera se podrá obtener información para controlar empíricamente la hipótesis formulada. Los tipos de información pueden ser: numérica o textual. Mientras que las fuentes, origen o formas como se producen esos datos es primaria (como la encuesta, entrevista, observación, pruebas y técnicas proyectivas, grupo de discusión, etc.) o secundaria: datos cuantitativos (estadísticas, censos..) o bien cualitativos (actas, diarios, registros, memorias, documentos...) Es la hipótesis la que debe controlar y seleccionar los datos y no al revés.

Formas de almacenamiento de la información: Del concepto a la variable es un paso que está presente en varios momentos del proceso ya que cada diseño y cada técnica requiere especificidades particulares para operacional izar los conceptos.

4. Organización, análisis e interpretación de la información.

Esta etapa comienza con la organización, hoy la organización de los datos es uno de los principales problemas de la investigación. Un gran número de instancias generan información que muchas veces no se utiliza suficientemente más que con fines burocráticos. Los índices, la

modelización y la tipología como esquemas organizadores. Un índice es una suma de variables que permite resumir elocuente y rigurosamente el comportamiento asociado de varios de ellos en un solo número.

Se entiende por modelo de análisis cuando se utiliza algún procedimiento especial. El modelo más simple es el de asociación o correlación entre dos variables en tanto que en los más complejos se utilizan técnicas de factorización, de regresión; análisis jerárquico o el de Cluster. En otros términos, un modelo será siempre el resultado de combinar la definición del problema educativo situacional con buenas referencias bibliográficas consultadas, con un suficiente eso de la información secundaria disponible y con una inteligente producción de datos primarios.

Una tipología parte de la combinación de variables, generalmente nominales u ordinales y se rige muy fuertemente por parte de las hipótesis de trabajo.

Actualmente en Corporación Educativa Mayor del Desarrollo Simón Bolívar se han realizado varios proyectos con respecto a los Sistemas de Información Académico-Administrativo o solamente de Sistemas de Información. A nosotros nos ha llamado la atención dos de estos proyectos los cuales son:

1. SIAA.

2. Diseño e Implementación de un Sistema de Información para el manejo de la Información Académico en el departamento de procesos Académicos de la Corporación Educativa Mayor del Desarrollo Simón

Bolívar realizado por los estudiantes Vladimiro García, Jhon Gómez, Deynizon Martínez y Aníbal Maury .

El primer proyecto tiene como objetivo fundamental el Diseño de un Sistema de Información Académico-Administrativo en la institución de Educación básica primaria N°5 en el municipio de Soledad con el propósito de brindar una solución integral a los entes encargados de la administración académica. También quieren fortalecer los factores claves del éxito de las instituciones prestadoras del servicio educativo.

1 Sistemas de Información Académico-Administrativo para la institución Básica N°5 de Soledad Atlántico, este proyecto se publicó el 30 de Mayo Del 2003 por los estudiantes Jhonny Carreño Mendoza y Kalmides Mendoza León .

El segundo proyecto lo que pretende es Diseñar o Implementar un Sistema que ayude a manipular flujos de información académica de la institución, ya que el que funcionaba anteriormente se consideraba como deficiente para realizar dichas actividades, debido a esto ellos trabajaron en el desarrollo de un sistema que remplazara al que se venia utilizando y que tuviera la capacidad de satisfacer las necesidades requeridas por el usuario.

En la Corporación Educativa Mayor del Desarrollo Simón Bolívar también se están llevado a cabo varios proyectos sobre los Sistemas de Información Académico-Administrativo, como es el caso de unos jóvenes de octavo(8°) semestre de Ingeniería de Sistemas que realiza el proyecto sobre la Sistematización de la Información Académica, Administrativa y de Psicorientación en el colegio distrital San Salvador. Este proyecto tiene como objetivo principal desarrollar un Sistema de Información que administre y gestione de forma integrada la información académica, administrativa y de Psicorientación.

Otro proyecto que se esta realizando actualmente es el de Diseño e Implementación de un Sistema de Información para el Instituto Técnico Comercial Amparo de Dios-Ajaib Singh que tiene como objetivo principal diseñar un sistema de información para agilizar y optimizar los procesos de inscripción, matricula y notas.

Otro factor importante en este proyecto es la educación, la cual es parte fundamental en todo ser humano. La educación es una estructura social, esta estructura llamó la atención de los sociólogos desde el inicio de la sociología.

Durkheim fue el primero en plantear que toda sociedad utiliza mecanismos de transmisión para adaptarse a las costumbres, valores, creencias y pautas

de comportamiento, valorados y permitidos socialmente. La educación y en concreto la escuela cumple para este autor la función de integrarse a la vida social, a partir de la homogenización y de la transmisión de valores universales y validos para todos los miembros de la sociedad. Esta función inicial fue ampliándose en la medida que las sociedades se fueron volviendo más complejas y la división del trabajo más especializada.

La educación dejo de ser solo transmisora o socializadora para convertirse en la encargada de asignar, ubicar y seleccionar a los individuos en las diferentes posiciones sociales. Posteriormente para algunos sociólogos, influidos por el marxismo y por Weber, la educación cumple una función de reproducción de las relaciones presentes en la sociedad, sencillamente se convierte en palabras de Althusser en Aparato ideológico del estado.

La educación como reproductora o transformadora de las relaciones sociales será una constante en los estudios sobre educación y sociedad. Ahora bien ¿que hace que la educación pueda ser considerada un fenómeno social? La educación solo se da dentro de la sociedad, es producto de procesos sociales más amplios, estructura y da contenido a las relaciones sociales, forma las identidades culturales individuales y colectivas, condiciona la vida social, las actitudes y la forma en que viven y se relacionan los miembros de la sociedad.

Por lo tanto la educación es un fenómeno social y como tal será abordada por la sociología de la educación,

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

La educación es una institución neutral, donde el individuo producto de su empeño y disposición intelectual, triunfa o fracasa. También la educación puede estudiarse como un sistema donde existen relaciones y estructura de

roles. Para esta corriente todo cambio social, incluyendo el educativo responde a agentes externos ya que es producto de una perturbación o fuerza que obliga a modificar el sistema educativo.

La educación participa en la forma como las sociedades determinan, asignan, seleccionan, evalúan y trasmiten conocimientos. El conocimiento esta sometido al poder de las clases sociales hegemónicas y al control social de mecanismos establecidos por la sociedad. El conocimiento y su difusión es una producción social. Legitimador de las prácticas sociales, y de la institucionalización de determinadas formas de conocer, aprender, analizar a través del currículum.

La sociología de la educación no solo se ha enriquecido de otras ciencias a través de su aporte, sino también de sus métodos para abordar la realidad. La sociología es una ciencia empírica por lo tanto la investigación es parte fundamental de su acontecer cotidiano, la sociología se construye en la investigación continuamente, la sociología emplea diferentes métodos para abordar la realidad, pero estos métodos deben garantizar la confiabilidad de los resultados. Los métodos más empleados son los siguientes: el histórico, estudio de los fenómenos sociales a través del tiempo, la realidad presente a través de los cambios en el pasado que la produjeron, el comparativo

Comparación de fenómenos similares en diferentes sociedades o en situaciones diferentes. el critico racional cuestionamiento critico de los valores y la cultura social.

La sociedad funciona como un todo y como tal todo su elemento o subsistemas están interrelacionados, la educación como parte de este sistema se ve influenciada por múltiples variables que la afectan y la

estructuran de una manera particular. Cada sociedad ha organizado su educación dependiendo de su filosofía social, su cultura, su sistema político, las características de su población y sus modos de producción. Algunas de estas características son:

- Toda educación responde a la concepción de la naturaleza humana, de la sociedad y del conocimiento que se considere valida (filosofía social).

- Toda educación transmitirá los valores, pautas de conducta o comportamiento, costumbres y tradiciones y tipos de conocimiento acordes con la cultura predominante o hegemónica.

- Toda educación formara a los individuos para funcionar dentro de los valores acordes y legitimadores de la filosofía política que prevalezca.

- Toda educación formara y seleccionara para el trabajo partiendo de las relaciones de producción, de las características de su población y del modelo económico que predomine.

Así tenemos que toda educación esta sometida a la influencia de los diferentes subsistemas sociales. La influencia de la educación en el aspecto económico ha sido objeto de estudio de la sociología de la educación. Esta se ha centrado en dos efectos básicos. Efectos Macro-Económicos: relaciones globales entre educación, empleo y crecimiento económico. Y efectos Micro-sociales: consecuencias que para los diferentes grupos sociales tiene su paso por el sistema educativo.

El conocimiento educativo, su contenido, la forma de trasmitirlo y la forma de evaluarlo son la clave para comprender los mecanismos de reproducción cultural. La educación forma parte de la superestructura y cumple una función ideológica, entendiendo por ideología falsa conciencia o sea la inculcación de significados funcionales y necesarios para la reproducción económica, las relaciones económicas y la división del trabajo. La educación es un aparato ideológico del estado y este es su aporte fundamental. El estado como mediador entre el poder de las clases económicamente dominantes y la estructura y contenidos educativos.

Además de ser un factor importante los Sistemas de Información para nuestro proyecto, también tenemos que tener en cuenta todo lo relacionado con las matriculas. Cada institución tiene sus normas y estatutos para realizar las matriculas. A continuación mencionaremos algunos datos importantes acerca de las Matriculas.

El proceso de Matrícula comprende la definición de directrices en cuanto a la aplicación correcta de requisitos establecidos en el Estatuto Orgánico, definición de materiales de matrícula; la elaboración de los documentos necesarios para el trámite de matricula; y el establecimiento de fechas de matricula.

La Dirección de Asuntos Estudiantiles es la responsable de cumplir con el reglamento y está asesorada por una Comisión denominada de "Admisión y Matrícula" integrada por: los titulares de la Oficina de Registro, quien coordina, de la Dirección de Asuntos Estudiantiles y Vicerrectoría Académica, cada titular deberá acreditar a un suplente por períodos completos de un año, ante la Comisión de Admisión y Matrícula.

Son objetivos de la Comisión de Matricula los siguientes:

- Organizar el proceso de matrícula.
- Lograr una coordinación eficiente entre las unidades involucradas en el proceso de matricula.
- Agilizar la toma de decisiones en los procesos de matricula.
- Planificar, organizar y evaluar el proceso de matricula.
- Evitar la duplicidad y ausencia de funciones en el proceso de matricula.

Son funciones de la Comisión de Matrícula las siguientes:

- Coordinar el proceso de matrícula.
- Recopilar, revisar, ordenar y controlar los materiales que se utilizan en proceso de matricula.
- Establecer cronogramas de actividades del proceso de matricula.
- Asesorar a las instancias que deben producir los materiales de matrícula para que éstos tengan la forma y contenidos necesarios.

- Controlar el cumplimiento de los requisitos de ingreso a las Instituciones.

- Fijar la cantidad de documentos necesarios para el proceso de matricula.

- Supervisar el proceso de producción y adquisición de libros y otros materiales didácticos, así como el proceso de entrega de los mismos a los estudiantes.

- Evaluar funcionamiento del proceso de matrícula para introducir las Mejoras correspondientes.

4.1.2 MARCO CONCEPTUAL.

Los conceptos de las palabras más importantes que trabajamos en nuestro proyecto son:

1. **Sistemas**: Los sistemas son una serie de elementos que funcionan en Conjunto para alcanzar un objetivo, o varios objetivos en común, al aceptar una entrada, procesarla y producir una salida de manera organizada.

2. **Información**: Son datos que adquieren significado dentro de un contexto.

3. **Datos**: Son la materia prima en la producción de información.

4. **Matriculas:** Se entiende por matricula el registro que se lleva a cabo en cualquier institución educativa de las personas admitidas para realizar estudios conducentes a grados y a títulos.

5. **Educación:** Es un proceso de formación permanente, personal, cultural y social que se fundamenta en una concepción integral de la persona humana, de su dignidad, de sus derechos y deberes.

6. **Sistemas de información**: Muestra las características mas sobresalientes del sistema, la integración de los datos de entrada, la relación que guardan entre sí y la reutilización de datos de salida para transformarlos en información nueva y necesaria.

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

7. **Diagrama de Flujo de datos**: El diagrama de flujo de datos (DFD), es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos. Siendo éste, una de las herramientas más comúnmente usadas, sobre todo por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejos que los datos que éste maneja.

Es importante tener en mente: los DFD no sólo se pueden utilizar para Modelar sistemas de sistemas de proceso de información, sino también como manera de modelar organizaciones enteras, es decir, como una herramienta para la planeación estratégica y de negocios.Los componentes de un diagrama típico de flujo de datos: Proceso, flujo, almacén, terminador.

8. **Modelo Entidad-Relación: Es** uno de los modelos lógicos basados en objetos y por lo tanto se enfoca primordialmente al los niveles conceptual y de visión. Una de las características de este modelo es que permite representar con claridad las limitantes de los datos. El modelo entidad-Relación es en esencia una herramienta para representar el mundo real por medio de simbologías y expresiones determinadas. Una entidad es un objeto que existe y puede ser distinguido de otro objeto. Una entidad puede ser concreta (un libro, un automóvil, etc) o abstracta (fecha, edad, etc).

9. **Colegio:** Lugar o establecimiento de enseñanza para niños y jóvenes de uno u otro sexo.

10. **Escuela:** Establecimiento publico o privado donde se le da a niños la instrucción primaria.

11. **Investigar:** Es preguntar para aprender.

12. **Investigación:** Es un proceso científico de auto-descripción, también se puede decir que la investigación es una búsqueda de conocimientos ordenada, coherente, de reflexión analítica y confrontación continua de los datos empíricos y el pensamiento abstracto, a fin de explicar los fenómenos de la naturaleza.

13. **Ciencia:** La ciencia es un conjunto de conocimientos en desarrollo, que se obtienen mediante los métodos cognoscitivos, y que se reflejan en conceptos exactos cuya verdad se comprueba y demuestra a través de la práctica social.

4.1.3 MARCO LEGAL.

Otro factor importante tanto para nuestro proyecto como para la sociedad es la Educación. La Educación es un derecho que todo ser humano debe adquirir no importando su Estatus Social.

La ley 115 de 1994 "por la cual se expide la Ley General de la Educación" define en el Articulo 1°. que la educación es un proceso de formación permanente, personal, cultural y social que se fundamenta en una concepción integral de la persona humana, de su dignidad, de sus derechos y deberes.

De conformidad con el artículo de la Constitución Política, define y desarrolla la organización y la prestación de la educación formal en sus niveles preescolar, básica (primaria y secundaria) y media, no formal e informal, dirigida a niños y jóvenes en edad escolar, a adultos, a campesinos, a grupos étnicos, a personas con limitaciones físicas, sensoriales y psíquicas, con capacidades excepcionales, y a personas que requieran rehabilitación social.

---

1. Ley general de la educación.

El Artículo 3°. Nos habla sobre la Prestación del servicio educativo. Nos dice que el servicio educativo será prestado en las instituciones educativas del estado, igualmente los particulares podrán fundar establecimientos educativos en las condiciones que para su creación y gestión establezcan las normas pertinentes y la reglamentación del Gobierno Nacional. De la misma manera el servicio educativo podrá prestarse en instituciones educativas de carácter comunitario, solidario, cooperativo o sin ánimo de lucro.

El articulo 5°. Fines de la Educación. Este artículo nos dice que estando conforme con el artículo 67 de la Constitución Política, la educación se desarrollara atendiendo los siguientes fines:

- El pleno desarrollo de la personalidad sin más limitaciones que las que le imponen los derechos de los demás y el orden jurídico, dentro de un proceso de la información integral, física, psíquica, intelectual, moral y demás valores humanos.

- La formación en el respeto a la vida y a los demás derechos humanos.
- La formación en el respeto a la autoridad legitima y a la ley, a la Cultura Nacional, a la historia colombiana y a los símbolos patrios.

1. Ley general de la educación.

- El acceso al conocimiento, la ciencia, la técnica y demás bienes y valores de la cultura, el fomento a la investigación y el estimulo a la creación artística en sus diferentes manifestaciones.

- La adquisición de una conciencia para la conservación, protección y mejoramiento del medio ambiente, de la calidad de vida y del uso racional de los recursos naturales.

- La formación en la práctica del trabajo, mediante los Conocimientos técnicos y habilidades, así como en la valoración Del mismo como fundamento del desarrollo individual y social.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA

1. Ley general de la educación.

5. METODOLOGIA.

5.1 Métodos de investigación.

El método trabajado en este proyecto es el Deductivo, el cual parte de una Observación general para sacar conclusiones sobre las necesidades que posee la escuela Ana Consuegra de Bolívar y el colegio José Consuegra Higgins con respecto a los procesos Académicos y Administrativos.

5.2 Tipos de investigación.

Existen varios tipos de investigación los cuales son: Histórica, descriptiva, experimental, cuasi-experimental, correlaciona, estudio de caso, expostfacto. El tipo de investigación que utilizamos es el descriptivo, el cual describe las características de un conjunto de sujetos o áreas de interés.

5.3 Técnicas para la recolección de información.

Las técnicas de recolección de información que fueron utilizados son la entrevista y la encuesta. Estas técnicas nos ayudaron a comprender y profundizar la investigación en esta institución, además nos proporcionaron la información necesaria para lograr el objetivo de nuestro proyecto y detectar el mayor número de fallas que se encuentren en esta institución.

5.4 Población.

La escuela Ana Consuegra de Bolívar y el Colegio José Consuegra Higgins Constan de una población de (260) doscientos sesenta alumnos. Que se Reparten de la siguiente forma: jornada matutina (primaria) y la jornada Vespertina (secundaria).

5.5 Muestra.

En la escuela Ana Consuegra de Bolívar y el Colegio José Consuegra Higgins se tomo una muestra de (50) cincuenta encuestas, para realizar un Muestreo, la cual consta de cinco preguntas. Estas preguntas arrojaron los siguientes resultados:

1. ¿Cómo te parece la forma en que la institución lleva a cabo los procesos Académicos actualmente. ?

   - El 80% de la población encuestada contesto BUENO.
   - El 20% de la población encuestada contesto REGULAR.

2. ¿Cómo te parece la forma en que la institución lleva a cabo los procesos Administrativos actualmente. ?

   - El 20% de la población encuestada contesto BUENO.
   - El 60% de la población encuestada contesto REGULAR.
   - El 20% de la población encuestada contesto MALO.

3. ¿Actualmente te gustan como son los boletines y recibos de matriculas. ?

   - El 70% de la población encuestada contesto que SI.

   - El 30% de la población encuestada contesto que NO.

4. ¿Como crees tu que deberían ser estos boletines y recibos. ?

   Realizando una lectura detallada sobre lo que contestaron los Estudiantes de esta institución, llegamos a la conclusión de que el 75% de los encuestados quieren que sus boletines y recibos sean Mas elegantes y que sean impresos.

5. ¿Te gustaría que el colegio empleara una herramienta actualizada para mejorar la cantidad de los boletines y demás recibos. ?

   - El 90% de la población encuestada contesto SI.

   - El 10% de la población encuestada contesto NO.

Después de haber realizado el porcentaje sobre la muestra que tomamos en la Escuela Ana Consuegra de Bolívar y el Colegio José Consuegra Higgins llegamos a la conclusión de que la mayoría de los estudiantes desean que la institución implemente una herramienta tecnológica para mejorar los procesos de notas y matriculas.

6. INGENIERIA DE REQUISITOS.

6.1 Descripción del Sistema Actual.

Actualmente cuando el padre de familia o el acudiente va a matricular a su hijo, se verifica si el estudiante es antiguo o nuevo, si el estudiante es nuevo se le piden los siguientes requisitos: dos fotos, fotocopia de tarjeta de identidad, registro civil, certificado de los años anteriores y certificado medico. Si el estudiante es antiguo solo se le piden el certificado medico y el certificado de que aprobó el año anterior. Luego de que se verifico la antigüedad de los estudiantes inmediatamente se comienza el proceso de matricula, no sin antes examinar si el estudiante se encuentra a paz y salvo. Los requisitos que se necesitan para que el estudiante se encuentre a paz y salvo son: el boletín del año anterior ganado y que no tengan deudas de mensualidades pasadas.

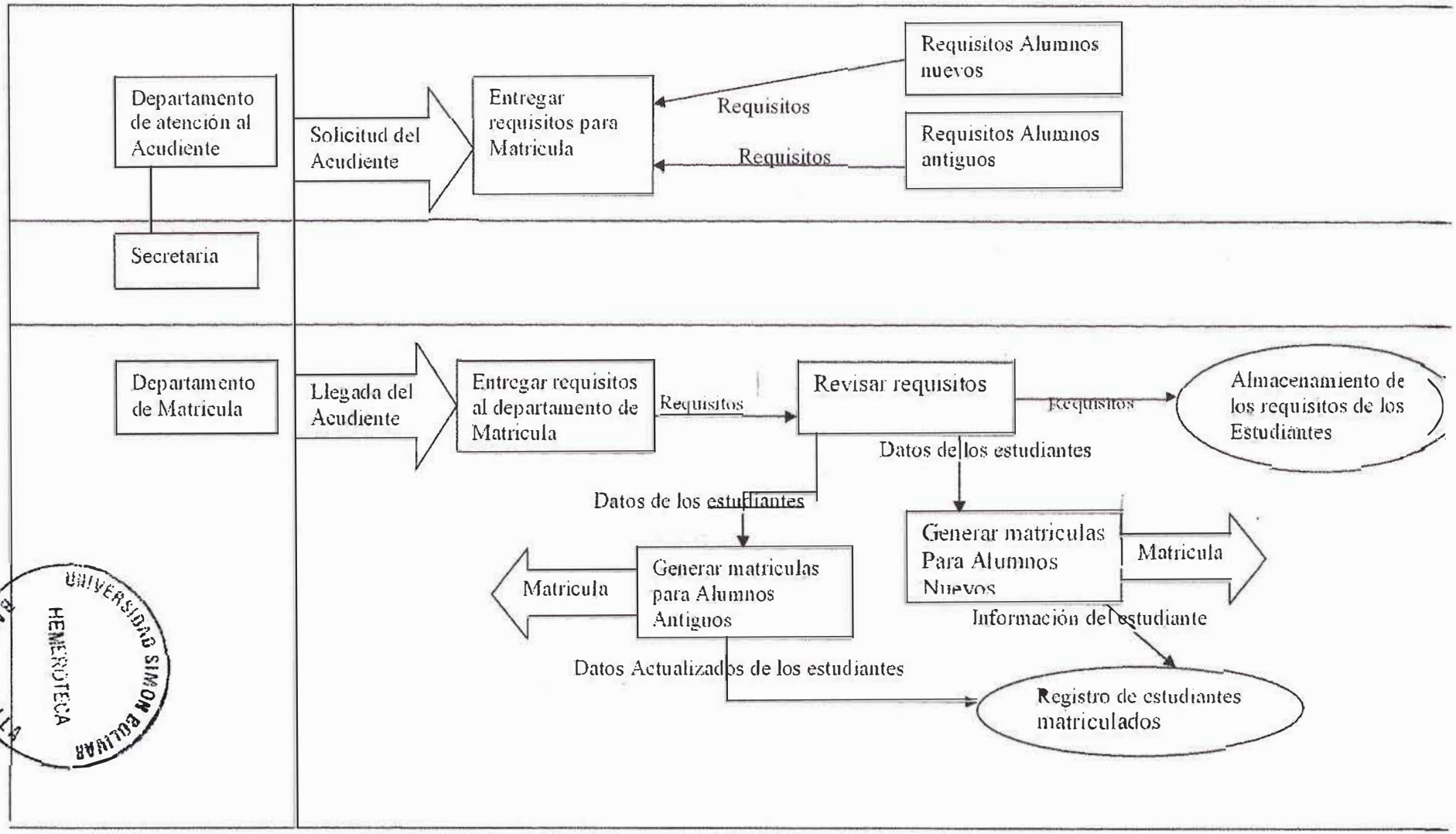
Luego de haber revisado sus referencias se verifica si esta a paz y salvo posteriormente nos orientamos a realizar la matricula. Para la realización de la matricula, el padre de familia o el acudiente firma el libro de matricula. Para la cancelación de las mensualidades el acudiente debe acercarse donde la secretaria para que le realice el volante de pago de la mensualidad, el acudiente se queda con el original y el colegio con la copia y esta es archivada.

Para la realización de las notas, la secretaria es la encargada de hacer entrega de las listas de estudiantes a cada profesor según el grado donde le corresponda dar clase, luego que el profesor recibe esta lista comienza tomar las respectivas notas de cada periodo. El profesor después de haber llenado la lista de notas se las devuelve a la secretaria para que esta realice

los boletines, después de esto la secretaria le entrega los boletines a los directores de grupo de cada curso para que estos se los entreguen a los acudientes en las reuniones de padres de familia. El año lectivo contiene 4 boletines, en el primer periodo el boletín o las notas son entregados verbalmente, los otros tres son entregados impresos pero el último tiene el acumulando de las notas de los boletines anteriores.

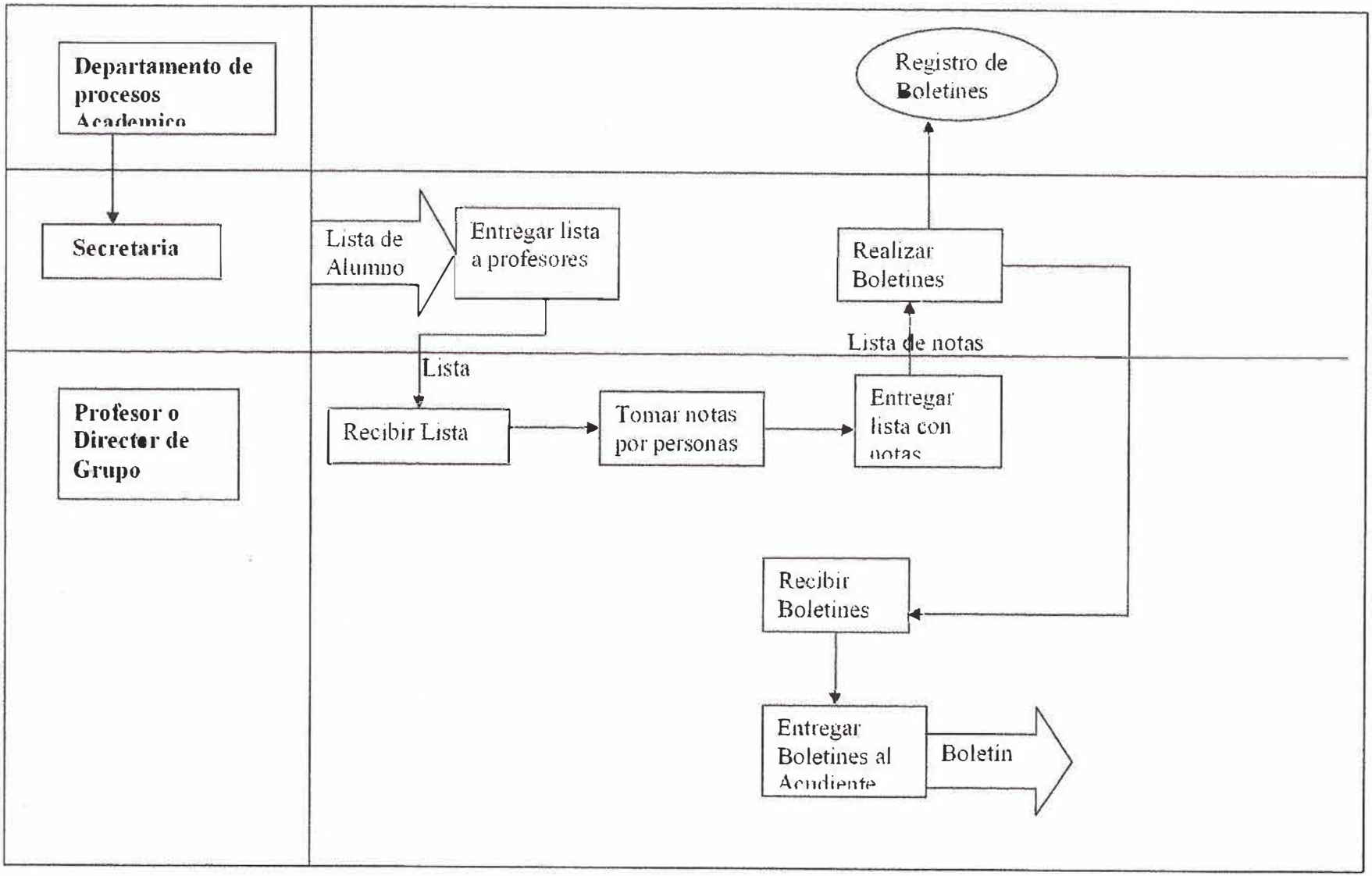
6.2 Diagrama de flujo del sistema actual (process modeller).

Proceso de Matricula.



UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

PROCESO NOTAS.



6.3 Identificación de requisitos.

Antes de realizar el Sistema de Información en la Escuela Ana Bolívar de Consuegra y el Colegio José Consuegra Higgins se debe tener en cuenta y conocer las especificaciones requeridas por el Cliente. La institución desea que este proyecto pueda realizar las siguientes funciones:

- Desarrollar un modulo que sea de fácil manejo para el usuario final.

- Realizar un modulo que realice la matricula de un estudiante.

- Crear un formulario donde se pueda realizar la lista de los alumnos de cada curso y que se puedan imprimir.

- Crear un formulario donde se puedan realizar los boletines y se impriman.

- Desarrollar un formulario que permita guardar y mostrar los profesores que trabajan en la institución, además que muestre el curso y la materia que él dicta en el colegio.

- Realizar un formulario que permita mostrar la historia de matricula de un estudiante.

- Crear un formulario que muestre y guarde los datos ingresados en los boletines.

- Crear ayudas sobre el software en caso de que el usuario final las necesite.

6.4 Análisis de requisitos.

Después de analizar los requisitos pedidos por el cliente podemos concluir que es posible realizar todas las peticiones mencionadas anteriormente, para mejorar los procesos Académico y Administrativo en la escuela Ana Bolívar de Consuegra y el Colegio José Consuegra Higgins del corregimiento de Isabel López (Atlántico).

6.5 Especificación de requisitos.

Los datos que serán sistematizados por el software son los siguientes registros:
Datos de los estudiantes, matricula, volante de pago, docentes, boletines.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

7. INGENIERIA DE INFORMACIÓN.

7.1 Misión.

Es una institución sin "animo de lucro", no oficial dedicada al servicio del proceso de información personal con una concepción integral que permite el desarrollo de todas sus facultades, o rentándolos al servicio de la cultura local, regional, nacional y a la producción del conocimiento critico

7.2 Visión.

La visión de la ESCUELA ANA CONSUEGRA DE BOLIVAR Y EL COLEGIO JOSE CONSUEGRA HIGGINS de Isabel López es la de contribuir al despegue definitivo de la comunidad de la forma mas efectiva mediante la plena ilustración del estudiante. La institución se proyecta desde un principio hasta el futuro trazándose metas de desarrollo institucional. Nuestro lema es orientar para que el alumno construya su propio conocimiento y los utilice para resolver los problemas que le plantee el entorno.

7.3 Historia.

El colegio bachillerato José Consuegra Higgins de Isabel López (antes colegio bachillerato Isabel López) fue fundado según acta N° 001 de enero 12 de 1981, por el consejo de gobierno de la universidad simón bolívar, bajo la dirección de su titular doctor José consuegra Higgins. Destacado hombre de ciencias económicas de América latina, hijo ilustre de Isabel López, quien preocupado por el adelanto intelectual de su tierra natal, se dio a la tarea de fundar este colegio con la aprobación y colaboración de la sala general de la Universidad Simón Bolívar como son: Doña Anita Bolívar de Consuegra, Leonello Marthe, Eduardo plgar (fallecido), Eusebio Consuegra (fallecido), Arsenio Gutiérrez (fallecido), Jorge Bolívar, Jorge Torres, Rafael Bolaños, Manuel Figueroa.

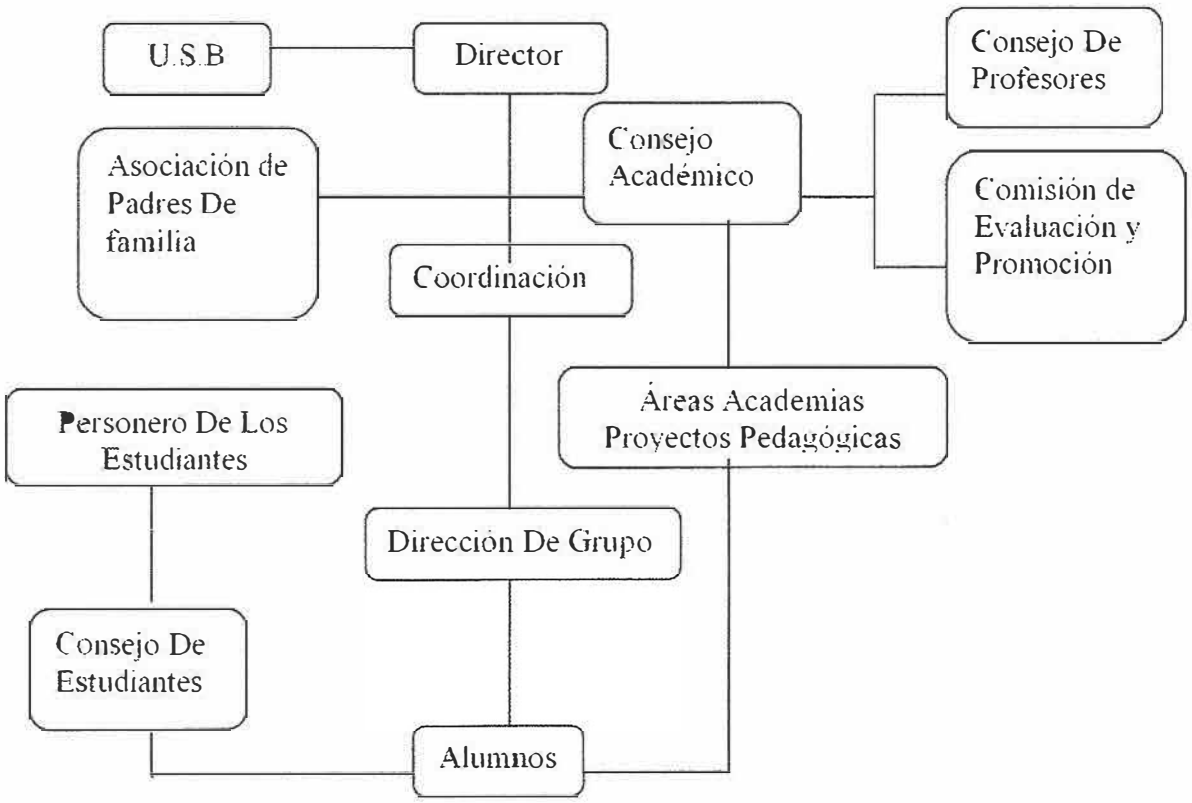
El colegio fue inaugurado el 30 de mayo de 1981, e inicio labores académicas el día 1 de junio del mismo año 1981, bajo la dirección de la licenciada Josefina Cantillo de Piñeres. El colegio inicio labores con 3 grados 6° - 7° - 8° y 63 alumnos. Su primera secretaria fue la señora Emita Peña de Redondo y en el mismo año en el mes de octubre fue remplazada por la señora Ana Cristina Barros Bolívar secretaria general actual. Los primeros profesores que laboraron en la institución fueron: Sulmira Higgins, Manuel Redondo Peña, Diomedes Barros, Nelly Pombo, Pedro Eduardo Ramos, Edgardo Coronado.

En 1985 la licenciada Josefina Cantillo renuncio a su cargo de directora y fue remplazada por el licenciado Diomedes Barros quien es el actual director. En la actualidad el colegio cuenta con un número de 260 alumnos. En el año 1998 el colegio cambio de razón social se llamaba colegio bachillerato de Isabel López y ahora se llama colegio bachillerato José Consuegra Higgins de Isabel López.

7.4 Políticas.

El colegio conciente de su misión educativa y en razón del saber social tiene como fin el perfeccionamiento de la persona, el conocimiento integral y la promoción en la educación en salud, realizando su participación en la proyección comunitaria de los programas salud y seguridad social, con la convicción de reafirmar los valores que potencien el desarrollo de la persona, amante de la vida, la libertad y practicante de la democracia, consiente de su valor y dignidad personal, alcanzando un nivel académico adecuado que le permita fácilmente ingresar a estudios superiores.

7.5 Organigrama.



8 ESPECIFICACION DE ENTIDADES

Nombre de la tabla: MATRICULAS Nombre largo o real: MATRICULAS Aplicación: PROCESOS ADMINISTRATIVOS				Descripción de la tabla: Almacena la cantidad de estudiantes matriculados su fecha y el valor.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	No matricula	PK	Varchar	5	0	No	Código único de la matricula.
2	Identificación		varchar	15	0	No	Identificación del alumno a matricular
3	Nombres		Varchar	25	0	No	Nombres del estudiante
4	Apellidos		Varchar	25	0	No	Apellidos del estudiante
5	Dirección		Varchar	30	0	No	Dirección del estudiante
6	Teléfono		Varchar	30	0	No	Teléfono del estudiante
7	Sexo		Varchar	5	0	No	sexo del estudiante
8	Curso		Varchar	5	0	No	Curso del estudiante
9	Grupo		Varchar	5	0	No	curso del estudiante
10	Jornada		Varchar	15	0	No	Jornada a la que se va a matricular
11	Fechan		Date	15	0	No	Fecha de nacimiento del estudiante
12	Fecham		Date	15	0	No	Fecha de la matricula del estudiante
13	Estado				0	No	Estado del estudiante

Nombre de la tabla: ESTUDIANTES Nombre largo o real: ESTUDIANTES Aplicación: PROCESOS ACADÉMICOS				Descripción de la tabla: Almacena toda la información del estudiante. Nombre, código, fecha de nacimiento y el sexo.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoest	PK	Varchar	5	0	No	Código único del estudiante
2	Cedulaest		Varchar	20	0	No	cedula del estudiante.
3	Expedidaest		varchar	30	0	No	Lugar de expedición de cedula
4	Nombreest		varchar	30	0	No	Nombre del estudiante
5	Apellidosest		varchar	30	0	No	Apellidos del estudiante
6	Edadest		Varchar	30	0	No	Edad del estudiante
7	Direccion		Varchar	30	0	No	Direccion de la donde vive estudiante
8	Telefonoest		Varchar	30	0	No	Telefono del estudiante
9	Sexoest				0	No	Sexo del estudiante.
10	Codprofesor	FK	Varchar	5	0	No	Código único del profesor
11	Codlogro	FK	Varchar	5	0	No	Código único del logro.
12	Fechaest		Date	10	0	No	Fecha en que el estudiante se registro
13	Estadoest				0	No	Estado del estudiante

Nombre de la tabla: ACUDIENTES Nombre largo o real: ACUDIENTES Aplicación: PROCESOS ACADÉMICOS				Descripción de la tabla: Almacena toda la información del ACUDIENTE. Nombre, código, fecha de nacimiento y el sexo.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoacu	PK	Varchar	25	0	No	Código único del estudiante
2	Cedulaacu		Varchar	25	0	No	cedula del acudiente.
3	Cedulaest	FK	Varchar	20	0	No	cedula del estudiante.
4	Expedidaacu		varchar	30	0	No	Lugar de expedición de cedula
5	Nombreacu		varchar	30	0	No	Nombre del acudiente
6	Apellidosacu		varchar	30	0	No	Apellidos del acudiente
7	Edadacu		Varchar	30	0	No	Edad del acudiente
8	Direccionacu		Varchar	30	0	No	Direccion donde vive acudiente
9	Telefonoacu		Varchar	30	0	No	Telefono del acudiente
10	Sexoacu				0	No	Sexo del acudiente.
11	Fechanacu		Date	10	0	No	Fecha en que el acudiente se registro
12	Estadoacu				0	No	Estado del acudiente

Nombre de la tabla: ASIGNATURAS Nombre largo o real: ASIGNATURAS Aplicación: PROCESOS ACADÉMICOS				Descripción de la tabla: Almacena todos los nombres de las asignaturas con sus respectivos códigos.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	codigoasig	PK	Varchar	30	0	No	Código único de la asignatura.
2	nombreasig		Varchar	25	0	No	Nombre de la asignatura
3	estadoasig		varchar	10	0	No	Estado de la asignatura
3	codcurso	FK	Varchar	5	0	No	Código único del curso.
4	Codigodocente	FK	Varchar	5	0	No	Código único del profesor.

Nombre de la tabla: LOGROS Nombre largo o real: LOGROS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena todo los logros que se manejan en la institución por los profesores.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	codigopen	PK	Varchar	20	0	No	Código único del pensum
2	nombrepen		Varchar	25	0	No	Nombre del pensum
3	Nombrelog		Varchar	100	0	No	Nombre del logro
4	Nombreasig		Varchar	25	0	No	Nombre de la asignatura
5	Nombrecur		varchar	25	0	No	Nombre del curso
6	Nombregru		Varchar	25	0	No	Nombre del grupo

Nombre de la tabla: CURSOS Nombre largo o real: CURSOS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena toda la información con respecto a los cursos que existen en la institución.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigocur	PK	Varchar	25	0	No	Código único del curso
2	Nombrecur		Varchar	25	0	No	Nombre del curso
3	Estadocur				0	No	Estdo del curso

Nombre de la tabla: AREAS Nombre largo o real: AREAS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena toda la información con respecto a las areas que existen en la institución.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoare	PK	Varchar	25	0	No	Código único del area
2	Nombreare		Varchar	25	0	No	Nombre del area
3	Estadoare				0	No	Estado del area

Nombre de la tabla: GRUPOS Nombre largo o real: GRUPOS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena toda la información con respecto a los grupos que existen en la institución.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigogru	PK	Varchar	25	0	No	Código único del grupo
2	Nombregru		Varchar	25	0	No	Nombre del grupo
3	Estadogru				0	No	Estdo del grupo

Nombre de la tabla: PROFESIONES Nombre largo o real: PROFESIONES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena toda la información con respecto a los cursos que existen en la institución.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigopro	PK	Varchar	25	0	No	Código único del la profesión
2	Nombrepro		Varchar	25	0	No	Nombre del la profesion
3	Estadopro				0	No	Estdo del la profesion

Nombre de la tabla: DOCENTES Nombre largo o real: DOCENTES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena el código y el nombre del profesor.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Ceduladoc	PK	Varchar	25	0	No	cedula único del docente
2	Nombredoc		Varchar	20	0	No	Nombre del docente
3	Apellidodoc		Varchar	25	0	No	Apellido del docente
4	Direcciondoc		Varchar	25	0	No	Dirección del docente
5	telefonodoc		varchar	25	0	No	Telefono del docente
6	Expedidadoc		date	30	0	No	Expedida la identificacion docente
7	Edaddoc		Varchar	10	0	No	Edad del docente
8	Sexodoc				0	No	Sexo del docente
9	Fechandoc		Date	10	0	No	Fecha de nacimiento del docente
10	Asignaturadoc	FK	Varchar	30	0	NO	Asignatura que dicta el docente
11	Cursodoc	FK	varchar	10	0	No	Curso del docente
12	Gradodoc	FK	Varchar	10	0	No	Grado al que pertenece el docente
13	Estadodoc				0	No	Estado del docente

Nombre de la tabla: BOLETINES Nombre largo o real: BOLETINES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena las diferentes notas y logros de los estudiantes.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoest	FK	Varchar	25	0	No	Código único del estudiante.
2	nombreest		Varchar	25	0	No	Nombre del estudiante
3	Apellidoest		Varchar	30	0	No	Apellido del estudiante
4	Cursoest		Varchar	10	0	No	Carso del estudiante
5	grupoest		Varchar	10	0	No	Grupo del estudiante
6	Periodo		Varchar	10	0	No	Periodo al que se va realizar boletín
7	Codigodoc	FK	Varchar	25	0	No	Codigo del docente
8	Nombredoc		Varchar	25	0	No	Nombre del docente
9	Nombreasig		Varchar	25	0	No	Nombre asignatura
10	Codigolog	Fk	Varchar	25	0	No	Codigo del logro
11	Nombrelog		Varchar	100	0	No	Nombre del logro

Nombre de la tabla: DOCENTES/ESTUDIANTES Nombre largo o real: DOCENTES/ESTUDIANTES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena la información de la relación entres estas dos tablas.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigodoc	PK	Varchar	25	0	No	Código único del profesor.
2	Codigocur	PK	Varchar	25	0	No	Código único del curso

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA

Nombre de la tabla: ESTUDIANTES/LOGROS Nombre largo o real: ESTUDIANTES/LOGROS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena la información de la relación entres estas dos tablas.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoest	FK	Varchar	25	0	No	Código único del estudiante.
2	Codigolog	FK	Varchar	25	0	No	Código único de logros
3	Letras de notas	FK	Varchar		0	No	Letras con el cual se califican en la institución.

Nombre de la tabla: ESTUDIANTES/DOCENTES Nombre largo o real: ESTUDIANTES/DOCENTES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena la información de la relación entres estas dos tablas.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoest	FK	Varchar	25	0	No	Código único del estudiantes.
2	Codigodocr	FK	Varchar	25	0	No	Código único del profesor

Nombre de la tabla: ASIGNATURAS/ESTUDIANTES Nombre largo o real: ASIGNATURAS/ESTUDIANTES Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena la información de la relación entres estas dos tablas.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigoest	FK	Varchar	25	0	No	Código único del estudiante.
2	Codigoasig	FK	Varchar	25	0	No	Código único de asignatura.

Nombre de la tabla: DOCENTES/ASIGNATURAS Nombre largo o real: DOCENTES/ASIGNATURAS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena la información de la relación entres estas dos tablas.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Codigodoc	FK	Varchar	25	0	No	Código único del docente
2	Codigoasig	FK	Varchar	25	0	No	Código único de asignatura.

Nombre de la tabla: AUDITORIAS Nombre largo o real: AUDITORIAS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: donde se almacenan las registros del sistema			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Descripcion		varchar	25	0	No	Descripción de las auditorias.
2	Tabla	FK	Varchar	25	0	No	
3	Usuario		Varchar	25	0	No	Usuario del sistema
4	Fecha		date	25	0	No	Fecha de auditoria
5	Hora				0	No	Hora de auditoria

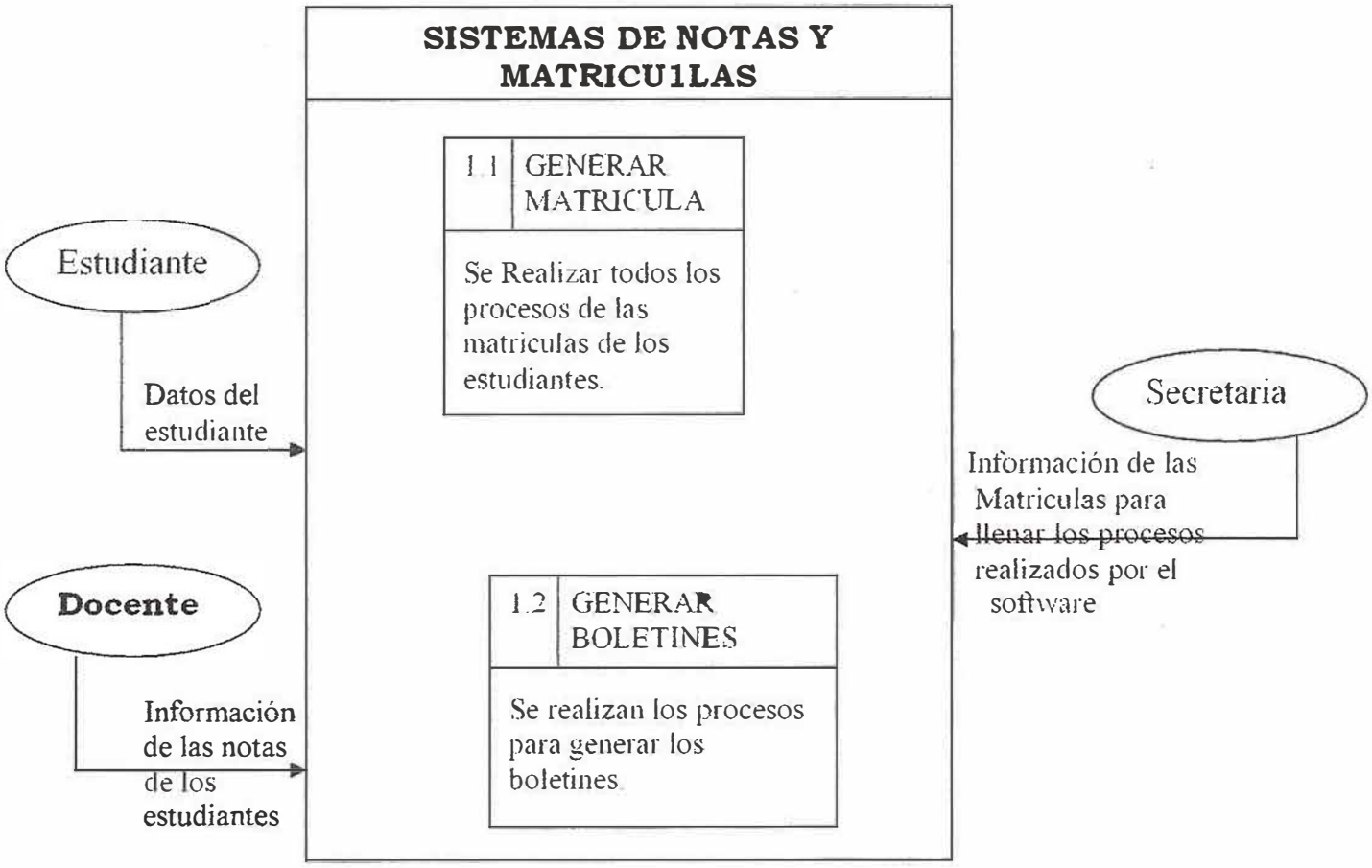
Nombre de la tabla: PENSUM Nombre largo o real: PENSUM Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: en esta tabla se almacenan los diferentes pensum del colegio.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	codigopen	PK	Varchar	25	0	No	Código único del pensum.
2	nombrepen		Varchar	25	0	No	Código único del profesor

Nombre de la tabla: PENSUM/ASIGNATURA Nombre largo o real: PENSUM/ASIGNATURA Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: en esta tabla se almacenan los diferentes pensum del colegio.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	codigopen	PK	Varchar	25	0	No	Código único del pensum.
2	nombrepen		Varchar	25	0	No	Código único del profesor
3	Codigoasig	FK	Varchar	25	0	No	Codigo de la asignatura
4	Nombreasig		Varchar	25	0	No	Nombre de la asignatura

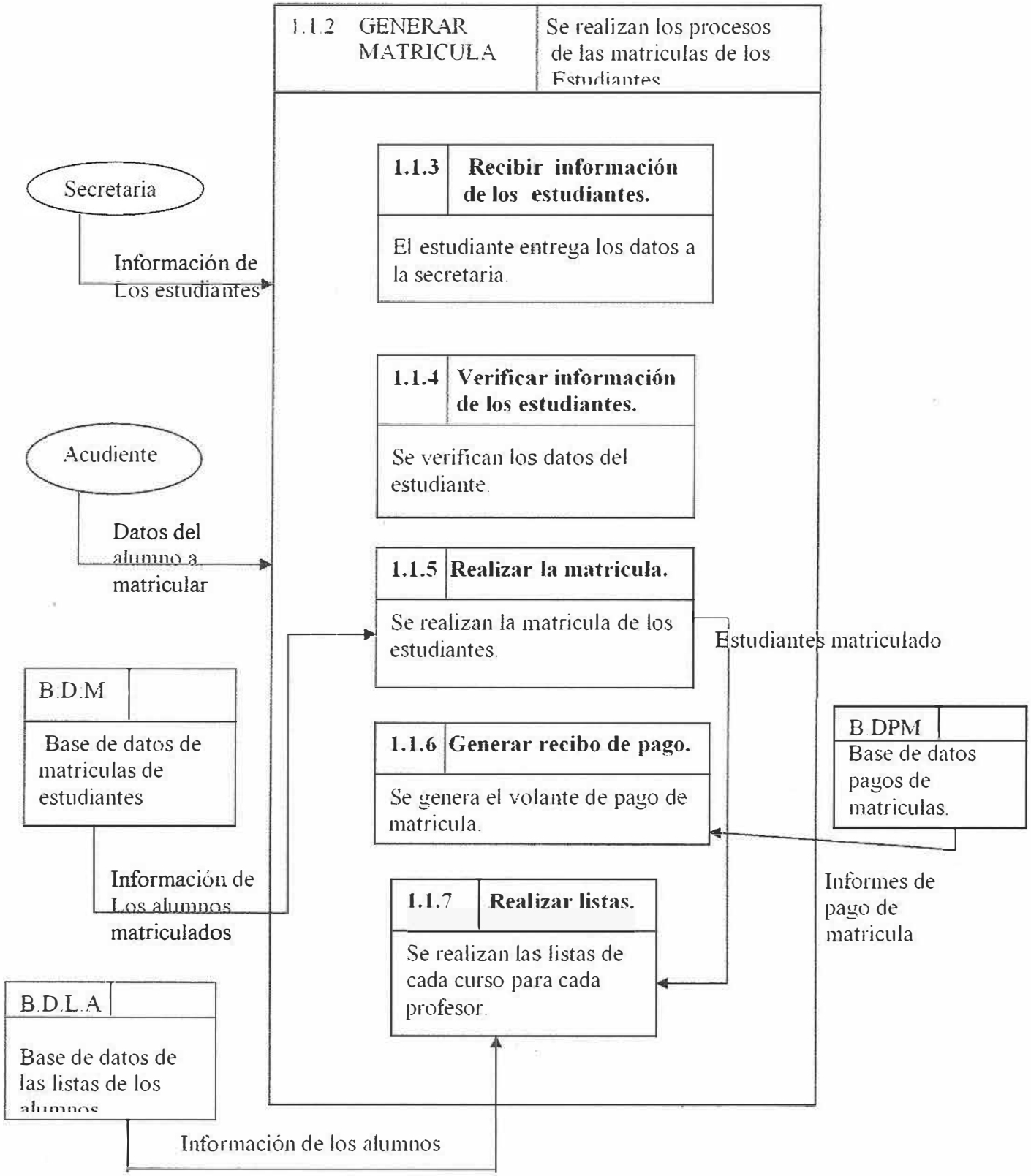
Nombre de la tabla: USUARIOS Nombre largo o real: USUARIOS Aplicación: PROCESOS ACADEMICOS				Descripción de la tabla: Almacena el código y el nombre del profesor.			
No.	Nombre Atributos	Llaves	Tipos de Datos	Long	Dec	¿Nulo?	Descripción
1	Cedula	PK	Varchar	25	0	No	cedula único del usuario
2	Nombre		Varchar	20	0	No	Nombre del usuario
3	Apellido		Varchar	25	0	No	Apellido del usuario
4	Dirección		Varchar	25	0	No	Dirección del usuario
5	Telefono		varchar	25	0	No	Telefono del usuario
7	Edad		Varchar	10	0	No	Edad del usuario
8	Loggin				0	No	Loggin del usuario
9	Password		Date	10	0	No	password
10	Tipo_usuario		Varchar	30	0	NO	Tipo de usuarios
13	Estado				0	No	Estado del usuario

8.1 DIGARMA DE FLUJO DE DATOS.

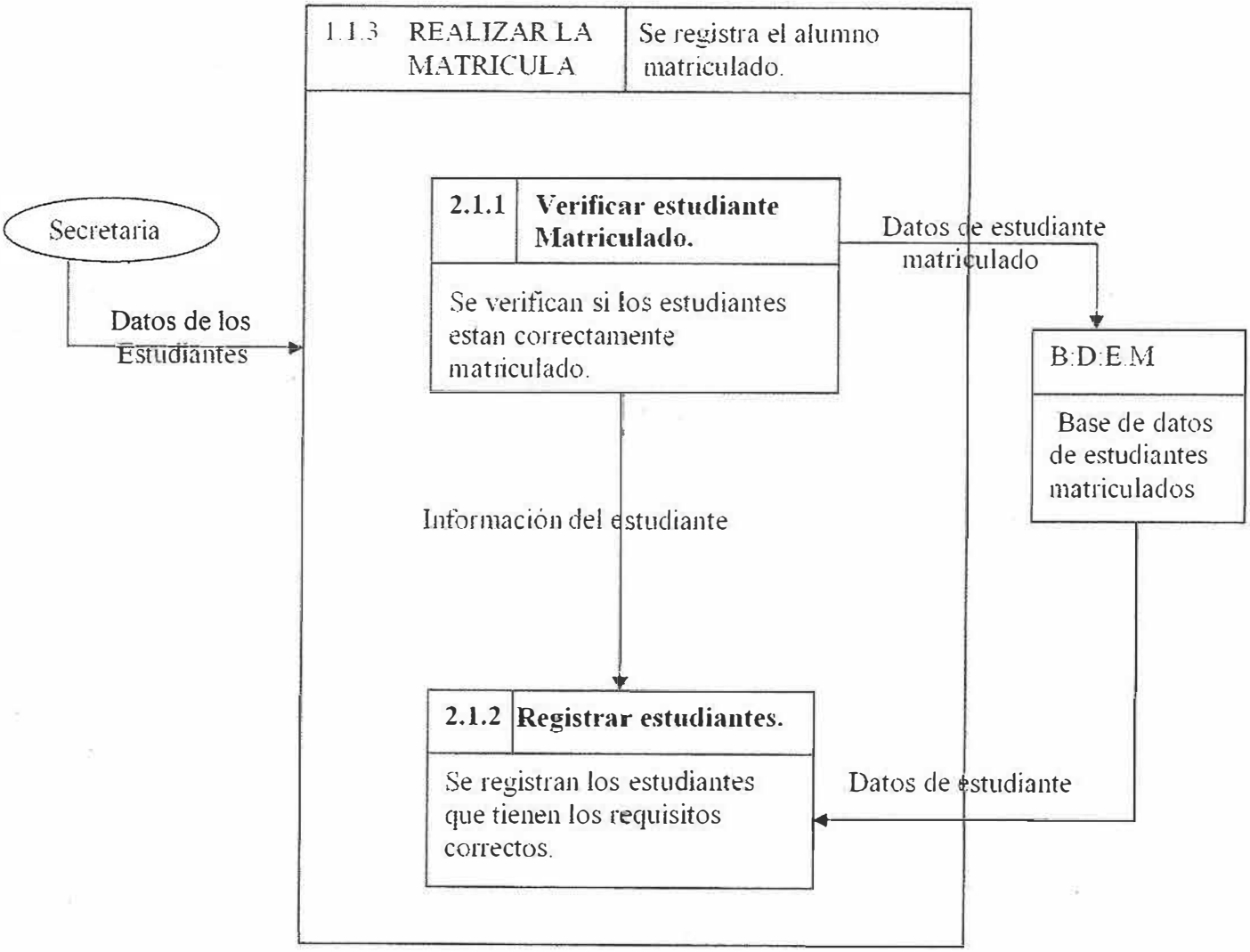
NIVEL 0



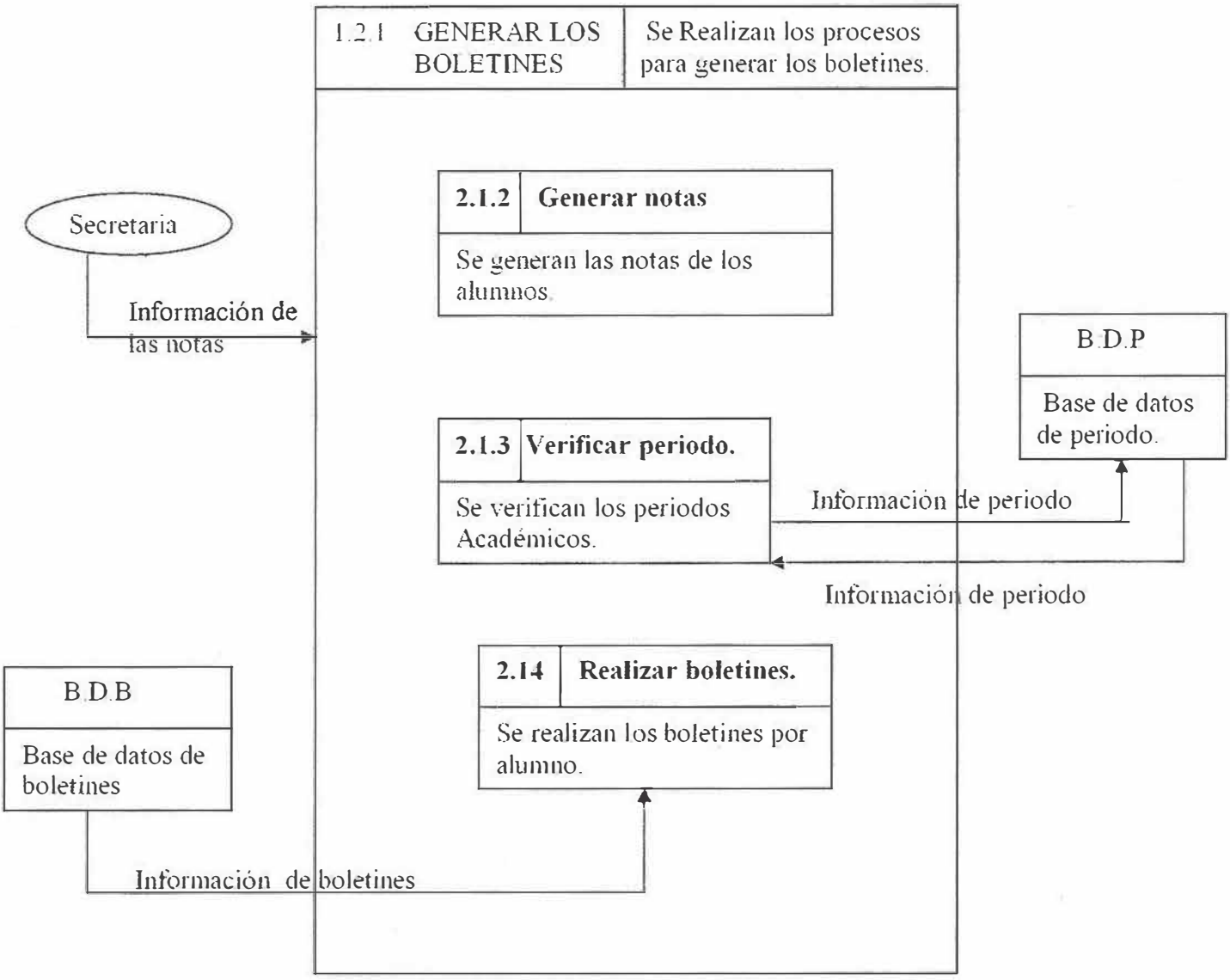
NIVEL 1 MATRICULA



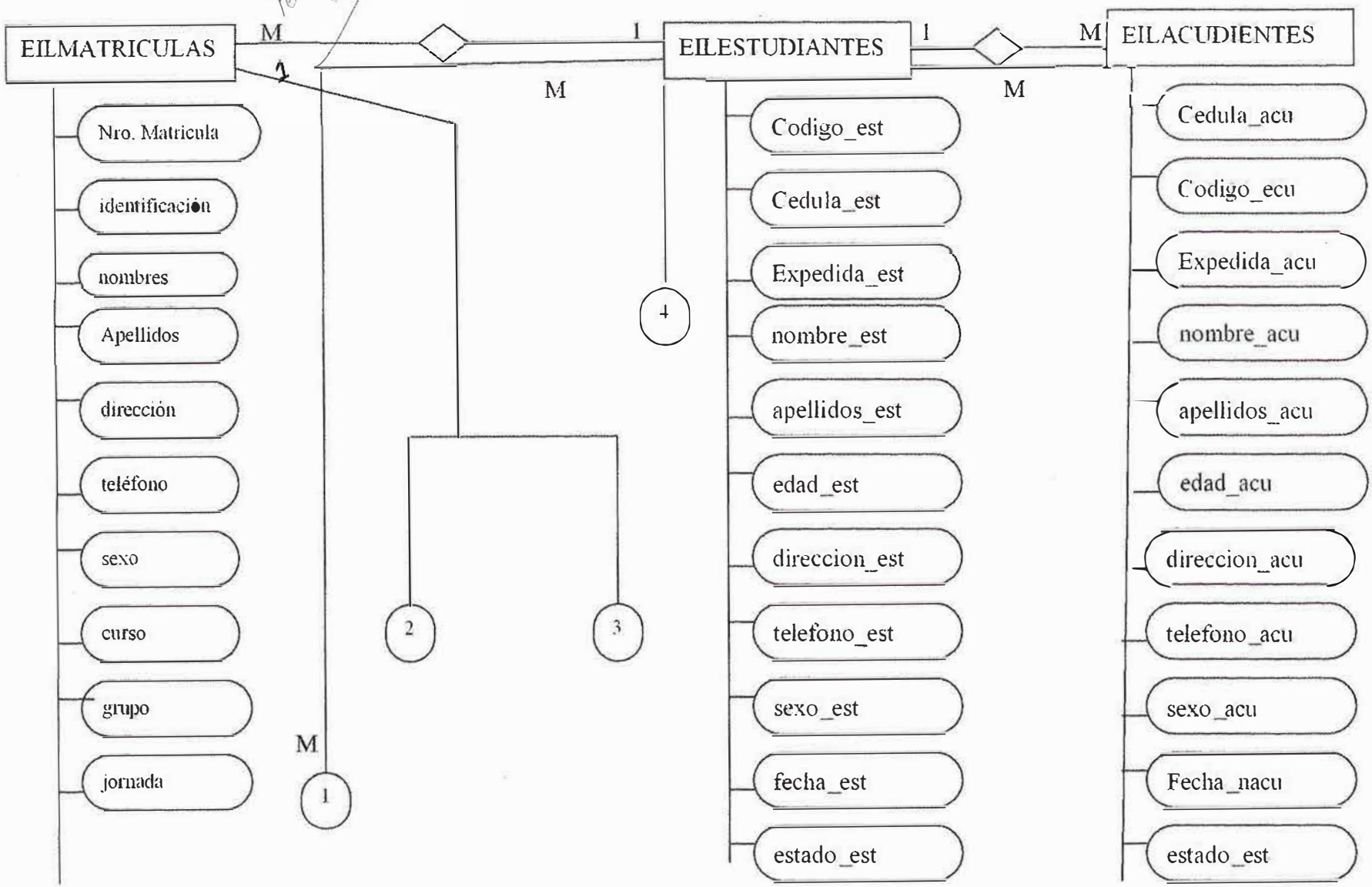
NIVEL II MATRICULAS

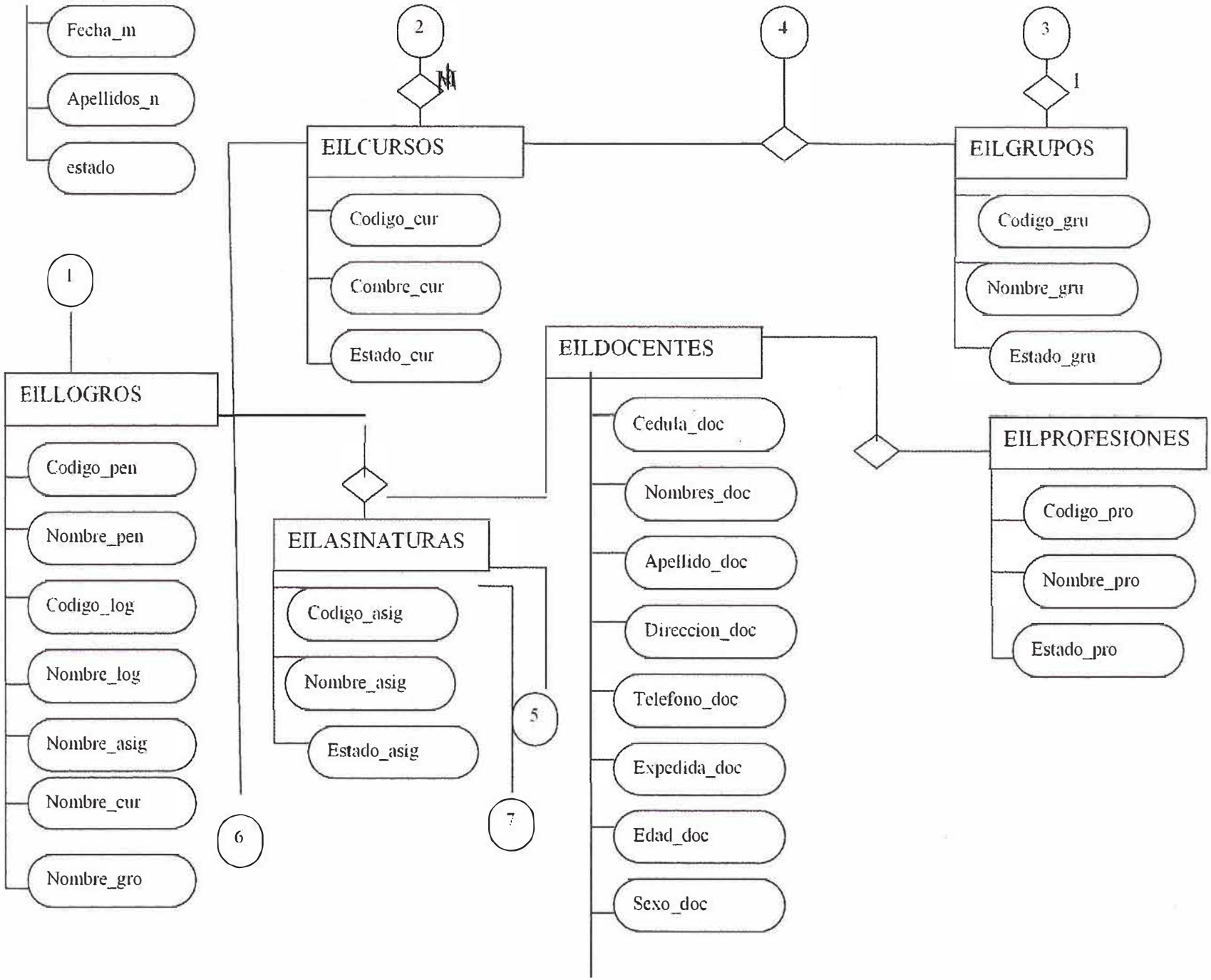


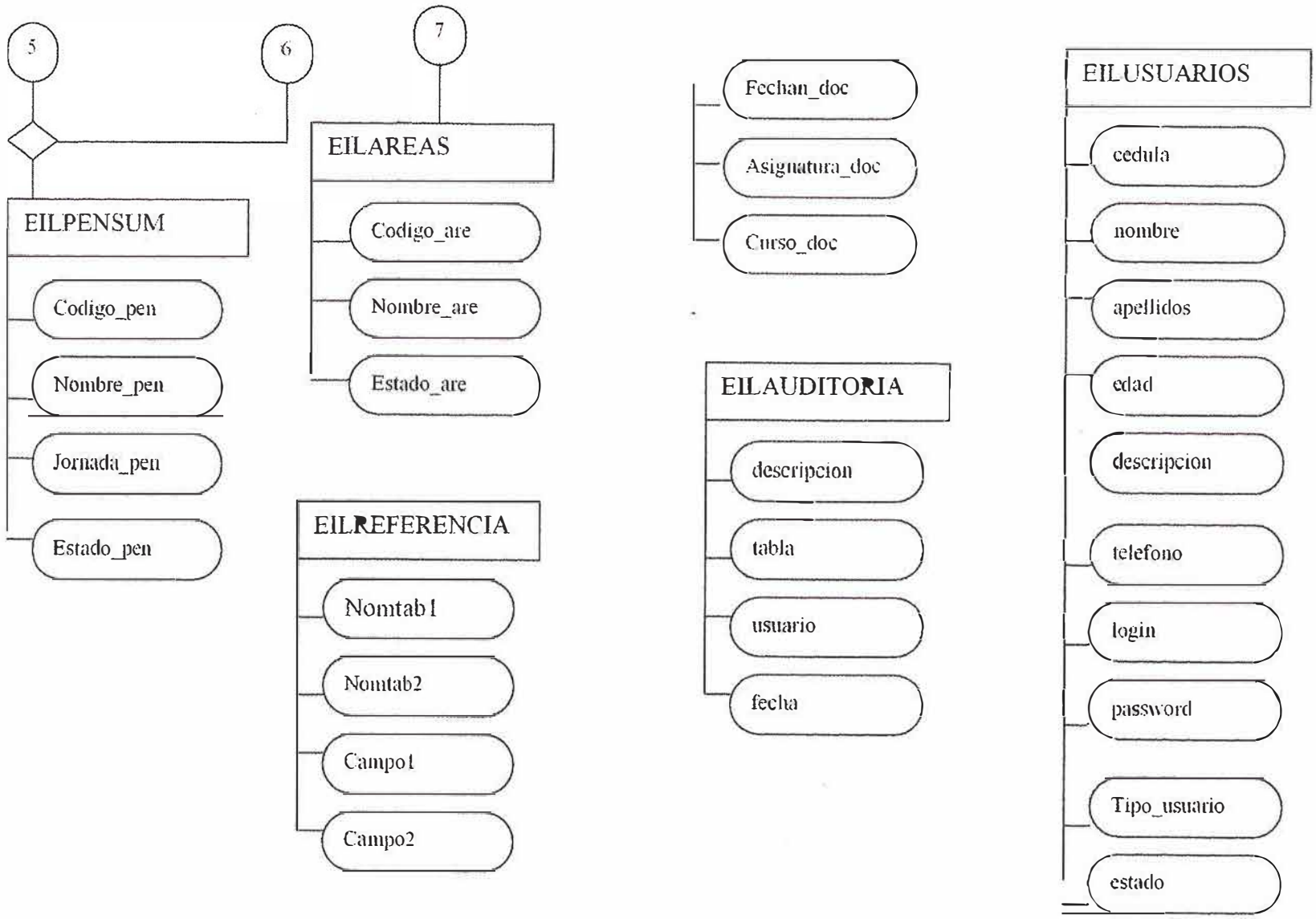
NIVEL 1 NOTAS



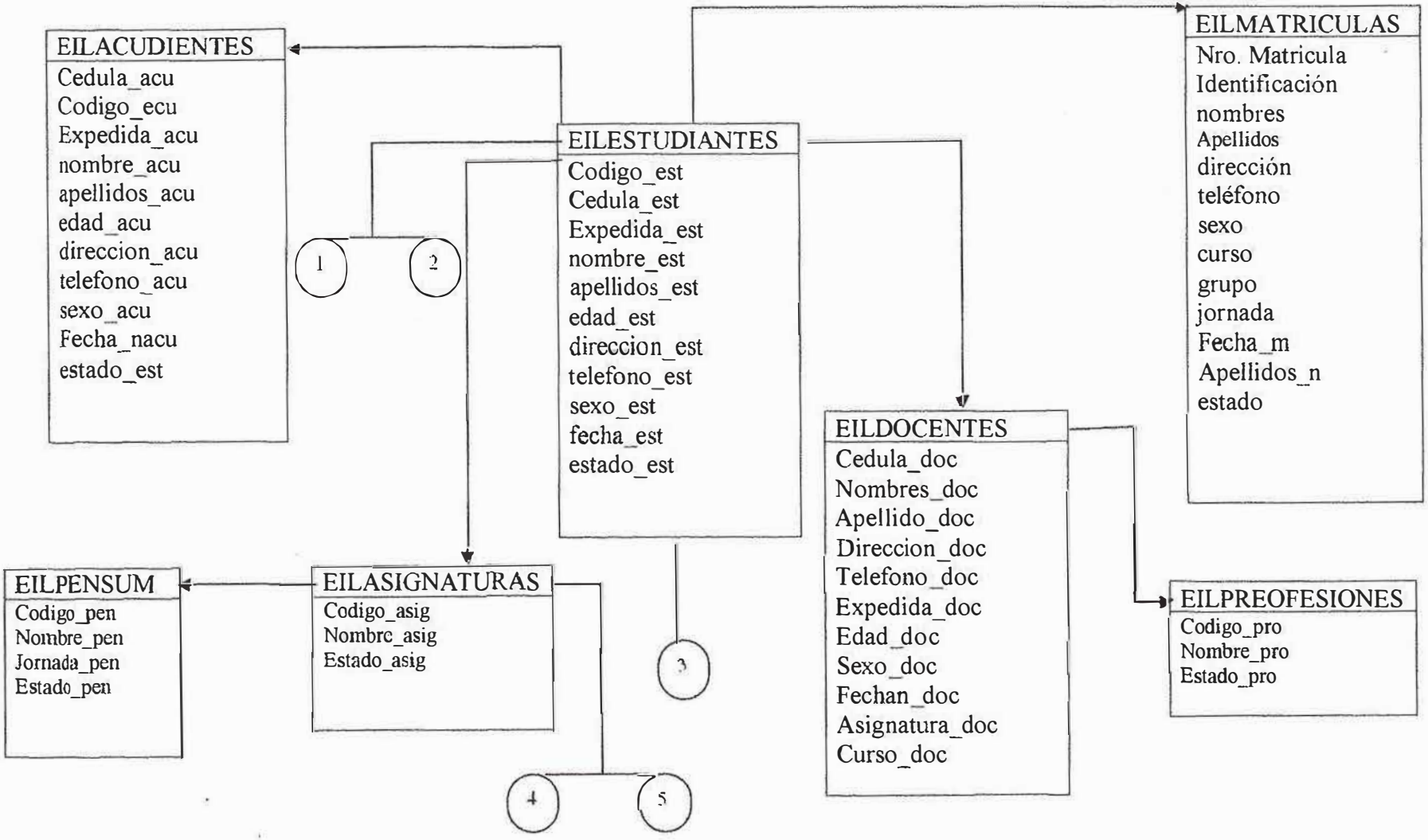
8.2 MODELO ENTIDAD-RELACION

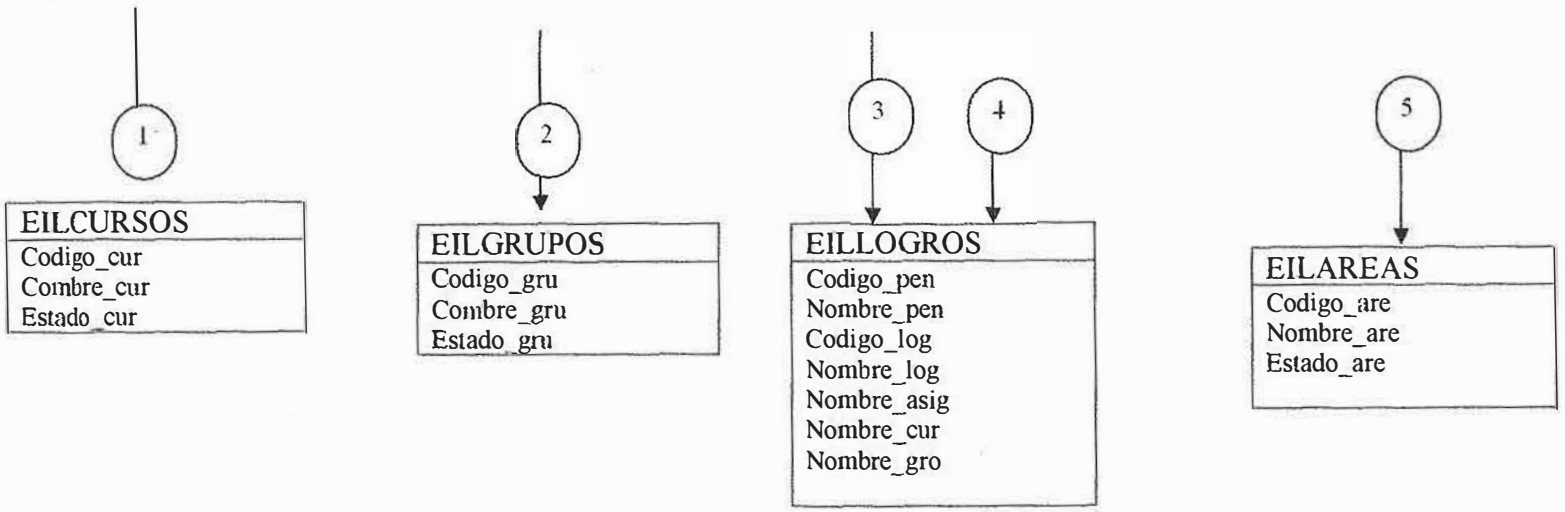



Fecha_m
Apellidos_n
estado
2
N
4
3
1
EILCURSOS
Codigo_cur
Combre_cur
Estado_cur
EILGRUPOS
Codigo_gru
Nombre_gru
Estado_gru
1
EILLOGROS
Codigo_pen
Nombre_pen
Codigo_log
Nombre_log
Nombre_asig
Nombre_cur
Nombre_gro
EILDOCENTES
Cedula_doc
Nombres_doc
Apellido_doc
Direccion_doc
Telefono_doc
Expedida_doc
Edad_doc
Sexo_doc
EILPROFESIONES
Codigo_pro
Nombre_pro
Estado_pro
EILASINATURAS
Codigo_asig
Nombre_asig
Estado_asig
5
6
7


5
6
7
EILPENSUM
Codigo_pen
Nombre_pen
Jornada_pen
Estado_pen
EILAREAS
Codigo_are
Nombre_are
Estado_are
EILREFERENCIA
Nomtab1
Nomtab2
Campo1
Campo2
Fechan_doc
Asignatura_doc
Curso_doc
EILAUDITORIA
descripcion
tabla
usuario
fecha
EILUSUARIOS
cedula
nombre
apellidos
edad
descripcion
telefono
login
password
Tipo_usuario
estado

8.4 MODELO RELACIONAL





EILUSUARIOS
cedula
nombre
apellidos
edad
descripcion
telefono
login
password
Tipo_usuario
estado

EILAUDITORIA
descripcion
tabla
usuario
fecha

EILREFERENCIA
Nomtab1
Nomtab2
Campo1
Campo2

8.5 DICCIONARIO DE DATOS.

Flujo de datos	Datos del alumno
Descripción	Los datos de los estudiantes matriculados en la institución.
Proviene de	La entidad: estudiantes
para	El proceso de diseño e implementación de un sistema de información para...

Flujo de datos	Información de notas de estudiantes
Descripción	Es la información sobre las notas de los estudiantes que existen en la institución.
Proviene de	La entidad: Profesor
para	El proceso de diseño e implementación

Flujo de datos	Información de matriculas.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene de	La entidad: Secretaria.
para	El proceso de diseño e implementación.

Proceso	1.1. Generar matricula
Descripción	Genera todos los procesos de matriculas de los estudiantes que estén en la institución.
Entrada	Datos del alumno, información de notas de estudiantes, información de matriculas.
Resumen lógico	Genera los distintos procesos que necesita el sistema para el desarrollo de matriculas.

Proceso	1.2. Generar boletines
Descripción	Genera todos los procesos de notas de los estudiantes que estén en la institución.
Entrada	Datos del alumno, información de notas de estudiantes, información de matriculas.
Resumen lógico	Genera los distintos procesos que necesita el sistema para el desarrollo de boletines.

Flujo de datos	Datos del alumno
Descripción	Los datos de los estudiantes matriculados en la institución.
Proviene de	La entidad: estudiantes
para	El proceso de diseño e implementación de un sistema de información para...

Flujo de datos	Información de notas de estudiantes
Descripción	Es la información sobre las notas de los estudiantes que existen en la institución.
Proviene de	La entidad: Profesor
para	El proceso de diseño e implementación

Flujo de datos	Información de matriculas.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene de	La entidad: Secretaria.
para	El proceso de diseño e implementación.

Proceso	1.1.2 Recibir información del estudiante.
Descripción	Procesa la información que el estudiante entrega a la secretaria
Entrada	Datos del alumno a matricular, información de estudiantes,
Resumen lógico	La información es entregada a la secretaria para continuar con el proceso de matricula.

Proceso	1.1.3 Verificar información del estudiante.
Descripción	Procesa la información de verificación de documentos que el estudiante entrega a la secretaria
Entrada	Datos del alumno a matricular, información de estudiantes,
Resumen lógico	La información es verificada por la secretaria para continuar con el proceso de matricula.

Proceso	1.1.4 Realizar la matricula.
Descripción	Procesa la información del estudiante después de ser matriculado.
Entrada	Datos del alumno a matricular, información de estudiantes,
salida	Información de los alumnos matriculados.
Resumen lógico	La información es verificada por la secretaria para continuar con el proceso de matricula.

Proceso	1.1.5 Generar recibo de pago.
Descripción	Procesa la información para la generación de recibos después que el alumno sea matriculado.
Entrada	Datos del alumno a matricular, información de estudiantes,
salida	Información de los pagos de matricula.
Resumen lógico	Se genera los recibos de pagos después de que el alumnos se haya cancelado el valor de la matricula.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

Proceso	1.1.6 Realizar lista de estudiantes.
Descripción	Procesa la información para la realización de listas de los estudiantes matriculados.
Entrada	Datos del alumno a matricular, información de estudiantes, información de estudiantes matriculados.
salida	Información de los alumnos.
Resumen lógico	Se generan las lista después de que el alumnos haya cancelado y matriculado.

Almacén de datos	Estudiantes matriculados
Descripción	almacena la información del estudiante después de ser matriculado.
. Flujosde datos recibidos	Información de los alumnos matriculados.
Flujos de datos proporcionado	ninguno

Almacén de datos	Pagos de matriculas
Descripción	almacena la información de los pagos de los estudiantes.
Flujos de datos recibidos	Información de los pagos de matricula.
Flujos de datos proporcionado	ninguno

Almacén de datos	Pagos de matriculas
Descripción	almacena la información de las listas de estudiantes matriculados en la institución.
Flujos de datos recibidos	Información de los alumnos.
Flujos de datos proporcionado	ninguno

Flujo de datos	Información de alumnos matriculados.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene del	proceso: realizar matriculas.
para	Base de datos de estudiantes matriculados.

Flujo de datos	Información de los pagos de matricula.
Descripción	Es la información sobre los pagos de matriculas de los estudiantes que existen en la institución.
Proviene del	proceso: generar recibos de pagos.
para	Base de datos de pagos de matricula.

Flujo de datos	Información de alumnos.
Descripción	Es la información sobre los estudiantes que existen en la institución.
Proviene del	proceso: realizar listas.
para	Base de datos de listas de estudiantes matriculados.

Proceso	1.2.1 Verificar estudiante matriculado.
Descripción	Se verifica la información para la realización de matricula.
Entrada	Datos del alumno, datos del estudiante.
salida	Información de los alumnos.
Resumen lógico	Se verifica la información de los documentos de los estudiantes para la elaboración de la matricula.

Proceso	1.2.2 Registrar estudiantes.
Descripción	Se registran los estudiantes que tengan los requisitos al dia
Entrada	Información de los alumnos.
salida	Datos de estudiantes matriculados.
Resumen lógico	Después de ser verificados los documentos se procede a registrar a los estudiantes.

Flujo de datos	Datos de estudiante matriculado.
Descripción	Es la información sobre los estudiantes registrados
Proviene del	proceso: registrar estudiantes.
para	Base de datos de estudiantes matriculados.

Flujo de datos	Información de estudiante.
Descripción	Es la información sobre los estudiantes que fueron verificados por la institución.
Proviene del	proceso: verificar estudiantes matriculados.
para	Registrar estudiantes.

Flujo de datos	Datos de estudiantes.
Descripción	Es la información sobre los estudiantes que existen en la institución.
Proviene del	Base de datos estudiantes matriculados.
para	Verificar estudiantes matriculados.

Almacén de datos	Estudiantes matriculados.
Descripción	almacena la información de los estudiantes matriculados en la institución.
Flujos de datos recibidos	Datos de estudiantes matriculados
Flujos de datos proporcionado	Datos de estudiante.

Proceso	1.2.3 Generar notas.
Descripción	Se generan las notas de los estudiantes para la generación de los boletines.
Entrada	Información de notas.
salida	ninguno.
Resumen lógico	Se genera las diferentes notas por materia y alumnos para la elaboración de boletines.

Flujo de datos	Información de notas.
Descripción	Es la información sobre las notas de los estudiantes
Proviene del	La entidad: secretaria.
para	Generar notas.

Proceso	1.2.4 Verificar periodo
Descripción	se verifica el periodo de los estudiantes para la elaboración de los boletines.
Entrada	Información de periodo.
salida	Información de periodo.
Resumen lógico	Se verifica el periodo del estudiante.

Flujo de datos	Información de periodo.
Descripción	Es la información sobre los periodos de cada uno de los estudiantes en la institución.
Proviene del	Proceso: verificar periodo.
para	Base de datos de periodo académico.

Flujo de datos	Información de periodo.
Descripción	Es la información sobre los periodos de cada uno de los estudiantes en la institución.
Proviene de la	Base de datos de periodo académico.
para	verificar periodo.

Proceso	1.2.5 Realizar boletines
Descripción	Se realizan los boletines de cada estudiantes con sus respectivas notas.
Entrada	Ninguno.
salida	Información de boletines.
Resumen lógico	Se realizan los boletines de los estudiantes.

Flujo de datos	Información de boletines.
Descripción	Es la información sobre los boletines de cada uno de los estudiantes en la institución.
Proviene del	Proceso: realizar boletines.
para	Base de datos de boletines.

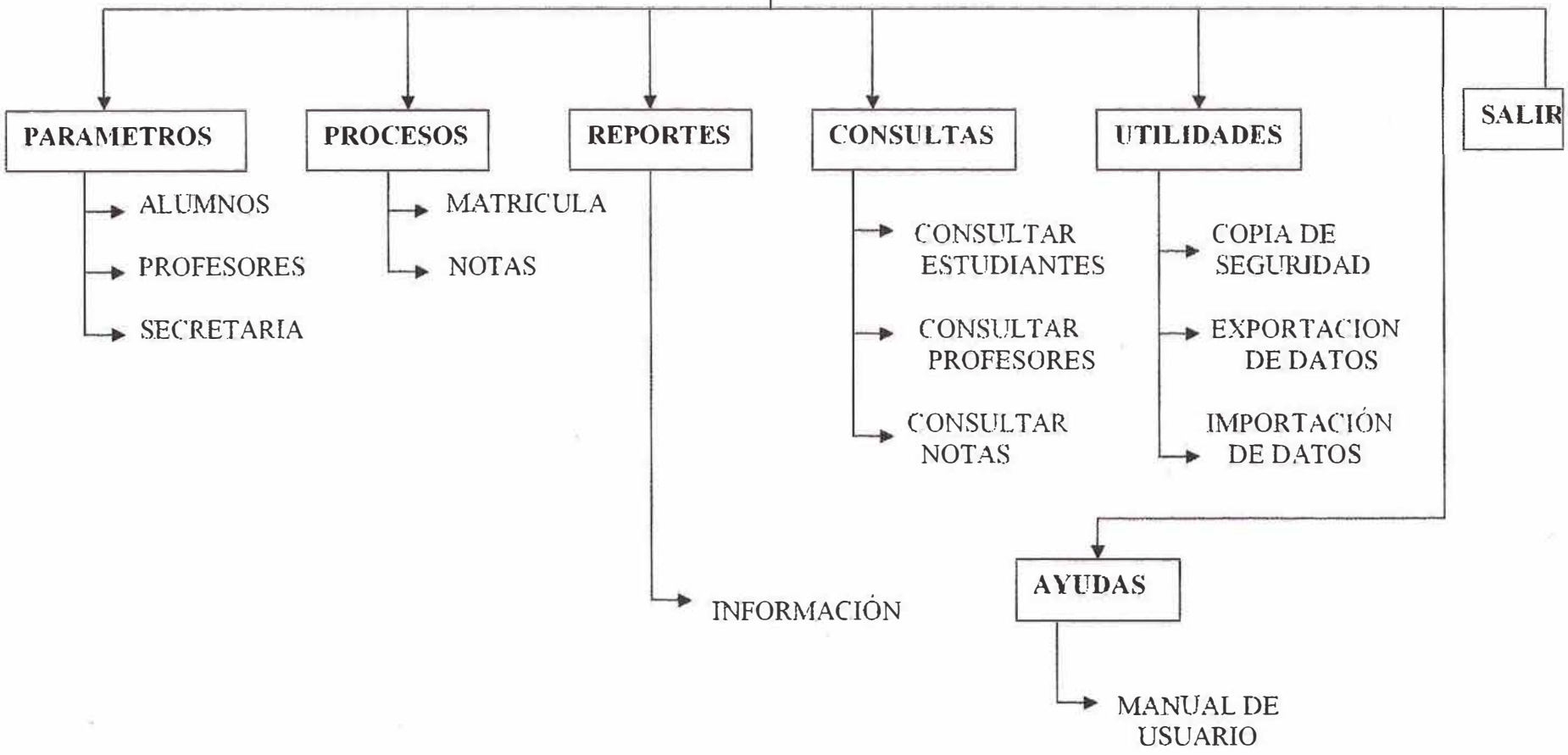
Almacén de datos	Boletines.
Descripción	almacena la información de los boletines de cada estudiante matriculado en la institución.
Flujos de datos recibidos	Información de boletines.
Flujos de datos proporcionados	Ninguno.

Almacén de datos	Periodo académico.
Descripción	Almacena la información de los periodos académicos de los estudiantes matriculados en la institución.
Flujos de datos recibidos	Información de periodo.
Flujos de datos proporcionado	Información de periodo.

9 DISEÑO DEL SISTEMA.

9.1 ESTRUCTURA FUNCIONAL(Diagrama jerárquico de funciones(menú)).

DISEÑO E IMPLEMENTACION SISTEMA DE INFORMACIÓN ACADEMICO-ADMINISTRATIVO PARA LA ESCUELA ANA BOLIVAR DE CONSUEGRA Y EL COLEGIO JOSE CONSUEGRA HIGGINS DEL CORREGIMIENTO DE ISABEL LOPEZ (Atlántico).

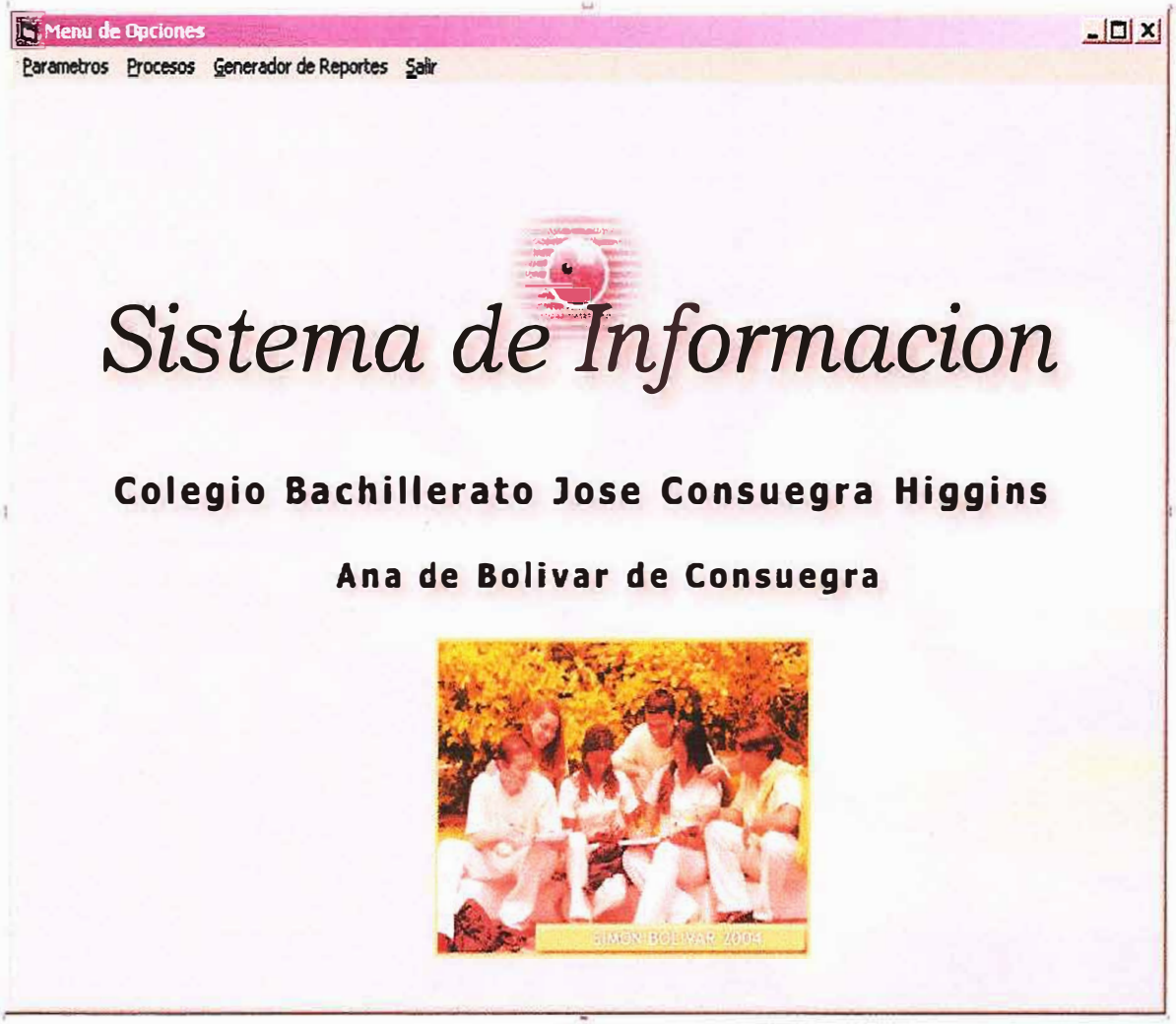


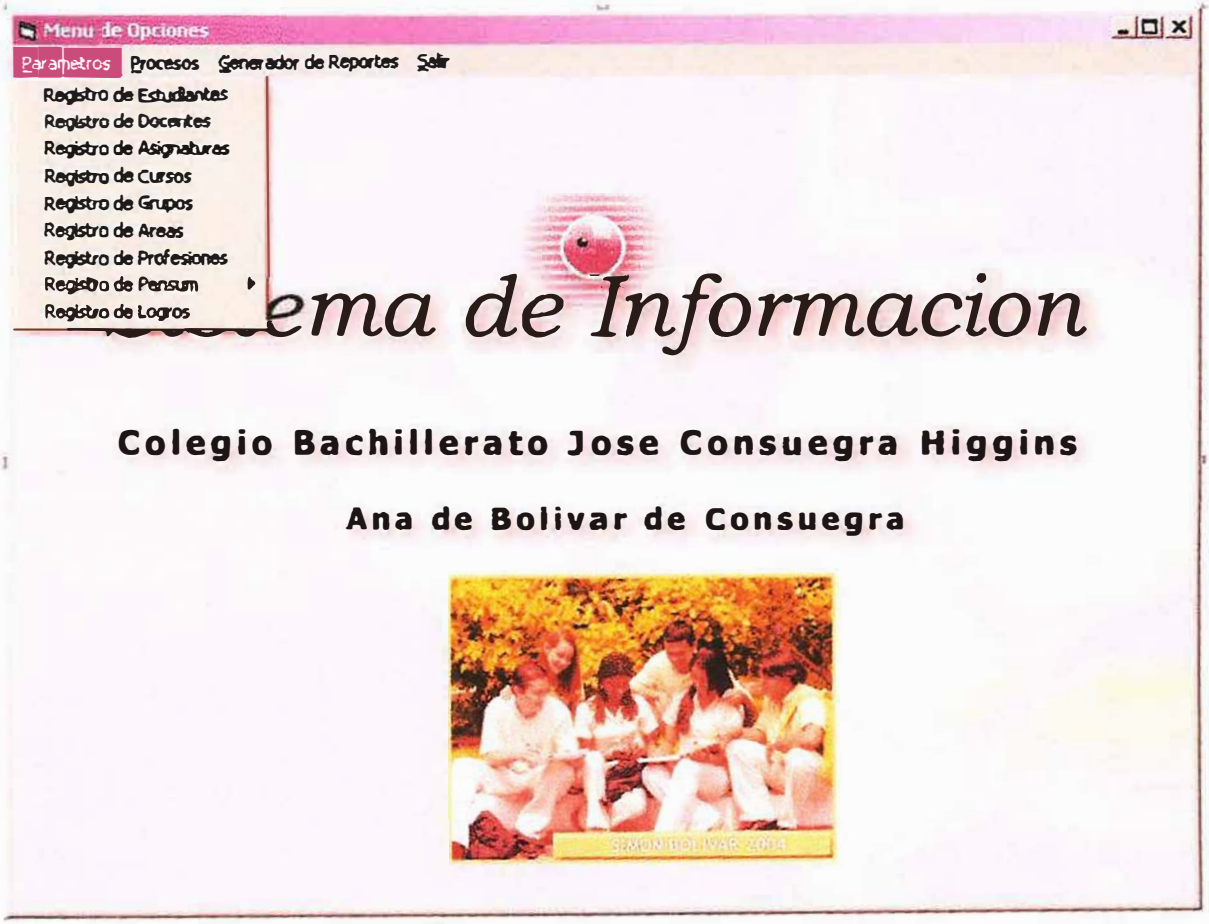
9.3 DISEÑO DE LAS INTERFASES.

CONTRASEÑA

En esta pantalla nos muestra el inicio de el programa en donde se deben digitar el nombre de usuario y la contraseña que nos sirve para tener una mejor seguridad de la información guardada y que personas ajenas al departamento administrativo y de notas no puedan ingresar al sistema deliberadamente.

MENU


Menu de Opciones
Parametros Procesos Generador de Reportes Salir
Sistema de Informacion
Colegio Bachillerato Jose Consuegra Higgins
Ana de Bolivar de Consuegra
SIMON BOLIVAR 2004


Menu de Opciones
Parametros
Procesos
Generador de Reportes
Salir
Registro de Estudiantes
Registro de Docentes
Registro de Asignaturas
Registro de Cursos
Registro de Grupos
Registro de Areas
Registro de Profesiones
Registro de Pensum
Registro de Logros
ema de Informacion
Colegio Bachillerato Jose Consuegra Higgins
Ana de Bolivar de Consuegra

Menu de Opciones

Parametros | Procesos | Generador de Reportes | Salir

- Registro de Estudiantes
- Registro de Docentes
- Registro de Asignaturas
- Registro de Cursos
- Registro de Grupos
- Registro de Areas
- Registro de Profesiones
- Registro de Pensum
  - Nuevo Pensum
  - Asignacion de Asignaturas
- Registro de Logros

e Informacion

Colegio Bachillerato Jose Consuegra Higgins

Ana de Bolivar de Consuegra

Menu de Opciones

Parametros Procesos Generador de Reportes Salir

- Registro de Matriculas
- Registro de Notas

Sistema de Informacion

Colegio Bachillerato Jose Consuegra Higgins

Ana de Bolivar de Consuegra

Menu de Opciones

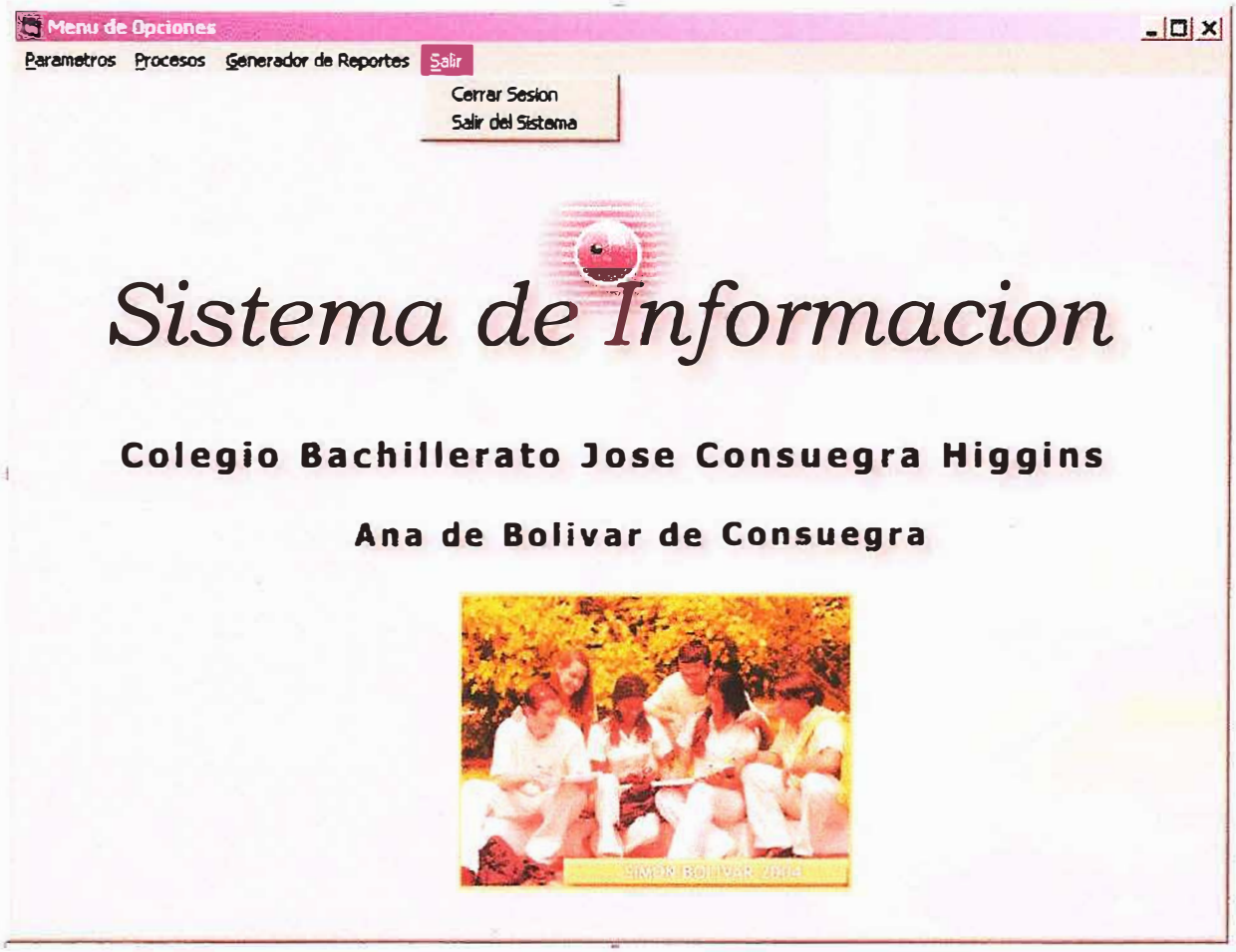
Parametros Procesos Generador de Reportes Salir

- Registro de Audiroria
- Creacion de Usuarios
- Exportar / Importar

Sistema de Informacion

Colegio Bachillerato Jose Consuegra Higgins

Ana de Bolivar de Consuegra



En estos formularios se encuentran las diferentes opciones que se manejan en el sistema de información del colegio

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

FORMULARIO DE MATRICULAS" DATOS PERSONALES"

Este es el formulario de las matriculas que se divide en tres sesiones conformado por nos botones que sirven para distintas instrucciones como lo son el guardar que guarda la información en pantalla, limpiar que limpia la información insertada, el imprimir que imprime la información en pantalla y el salir para salir del sistema, también tiene unas pestañas que son:

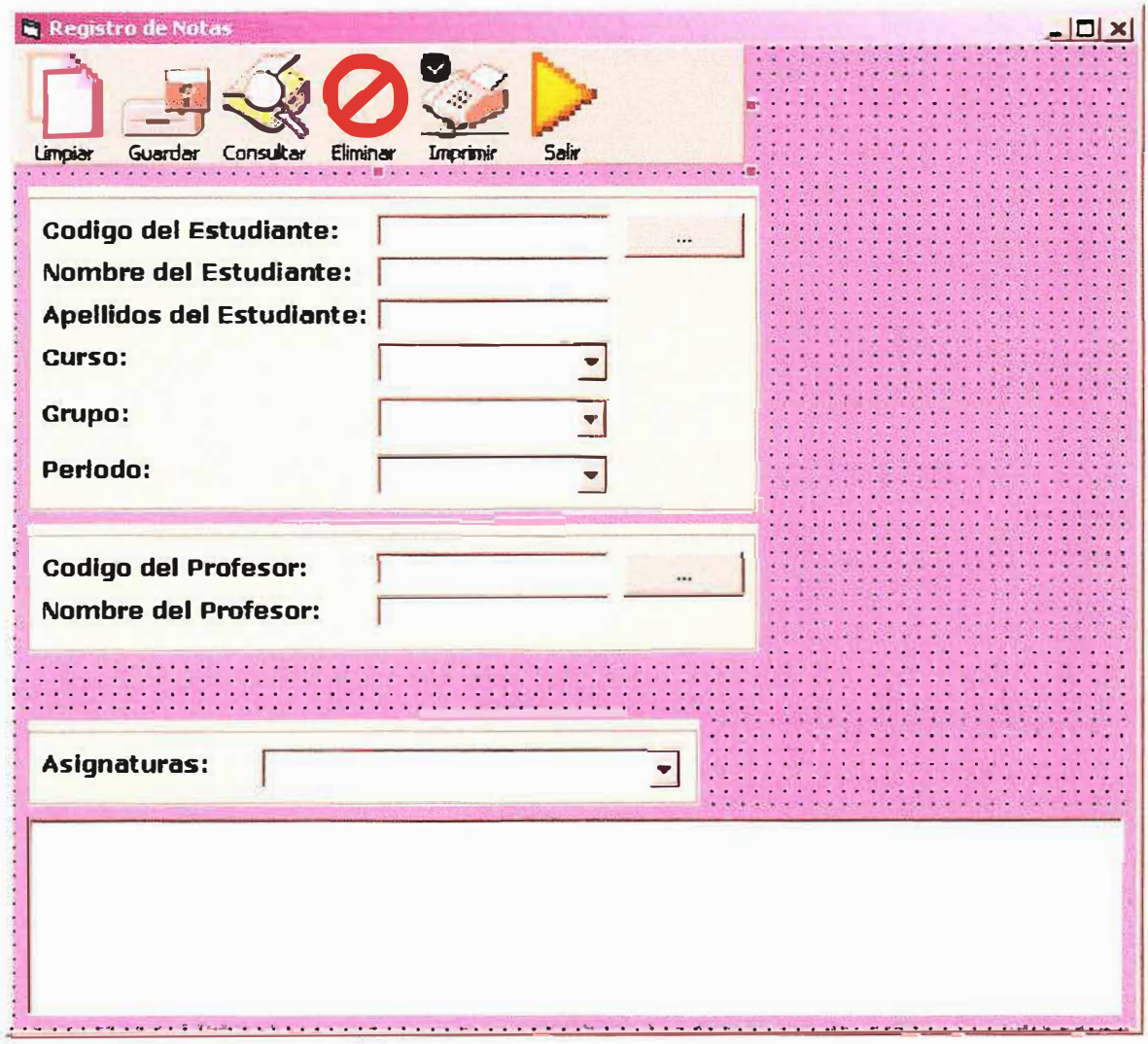
Información Personal: la información personal esta conformada por los siguientes atributos ; código , nombre(s) , apellido(s) , CC o T.I , expedida en, dirección, religión, lugar de nacimiento , fecha, teléfono , colegios anteriores, matriculado(a) en el grado, inicia estudios en este plantel, estudió anteriormente en este plantel, edad actual sexo y las firmas de los participantes de la matricula.

DATOS DEL ESTUDIANTE Y ACUDIENTE

Datos del Acudiente: este muestra información del acudiente del alumno también tiene unos atributos que son: nombre(s), apellido(s) , cedula, expedida en, dirección, teléfono, lugar de nacimiento, fecha y sexo.

Datos del Estudiante: este muestra información del acudiente del alumno también tiene unos atributos que son: nombre(s), apellido(s) , cedula, expedida en, dirección, teléfono, lugar de nacimiento, fecha y sexo

NOTAS



En esta pantalla mostrara y se harán los boletines de la institución y también se podrá registran los logros de los alumnos, esta constituida por los mismos botones descritos en la matricula y tiene como atributos el código que este cuando se la ingrese mostrara el nombre y apellidos del estudiante listo para ser llenado.

DOCENTES

En esta pantalla se ingresaran los datos básicos de los docentes como son: el nombre, fecha de nacimiento apellidos, fecha, C.C , expedida en, dirección, teléfono y el sexo. También tendrá los botones que sirven para distintas instrucciones como lo son el guardar que guarda la información en pantalla, limpiar que limpia la información insertada, el imprimir que imprime la información en pantalla y el salir para salir del sistema,

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA

AREAS

En este formulario se ingresan las áreas que se manejan en el colegio y la escuela y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

REGISTRO DE CURSOS

En este formulario se ingresan los cursos que se manejan en el colegio y la escuela y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

REGISTRO DE PENSUM

En este formulario se ingresan los pensum que se manejan en el colegio y la escuela y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

Registro de grupos

En este formulario se ingresan los grupos que se manejan en el colegio y la escuela y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

REGISTRO DE ASIGNATURAS

En este formulario se ingresan la asignaturas que se manejan en el colegio y la escuela y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

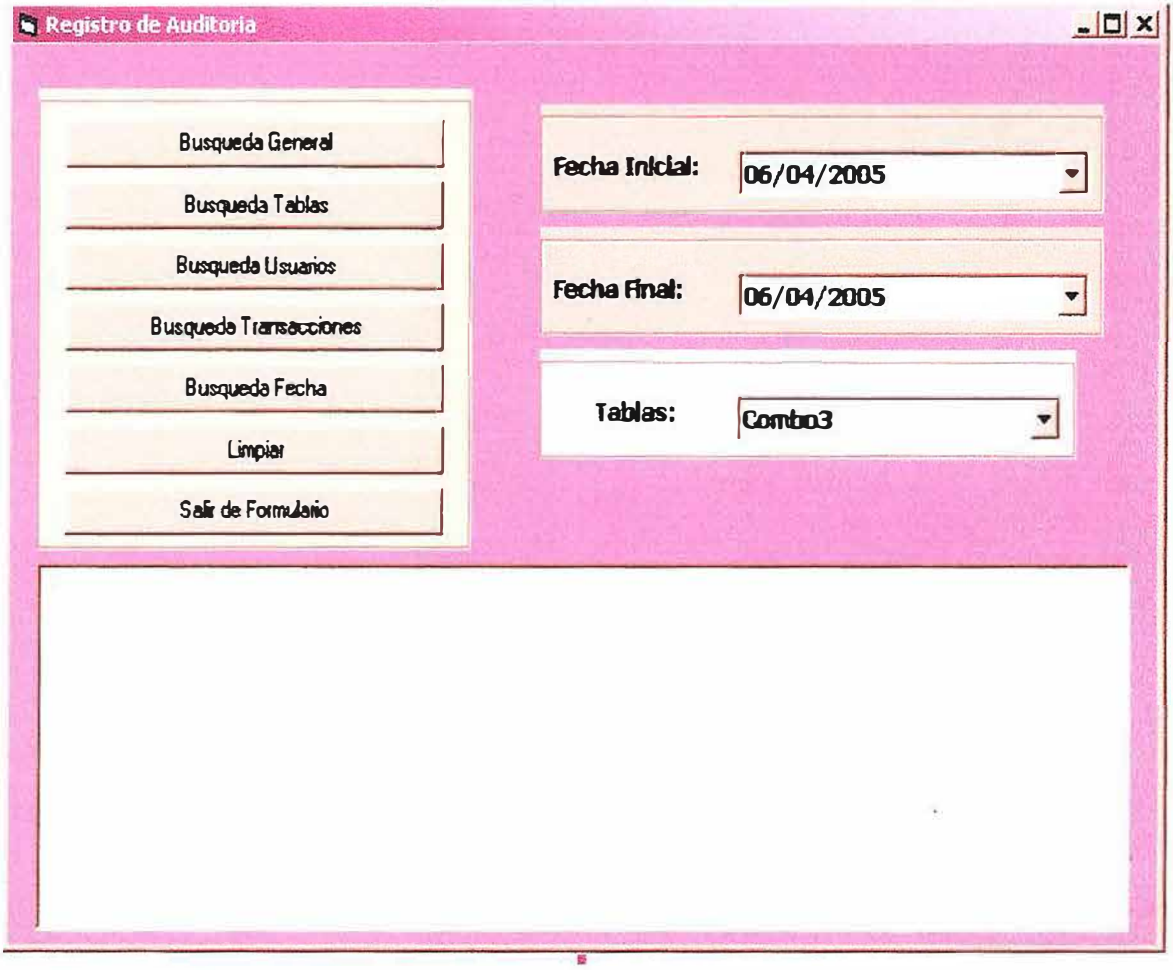
Registro de profesiones

En este formulario se ingresan las profesiones que tienen los docentes para poder dictar una asignatura adecuada, y tiene un campo estado para saber el estado en que se encuentra activo o inactivo.

ASIGNACIÓN DE ASIGNATURAS AL PENSUM

En este formulario es donde se asignan las asignaturas a un pensum determinado.

REGISTRO DE AUDITORIAS

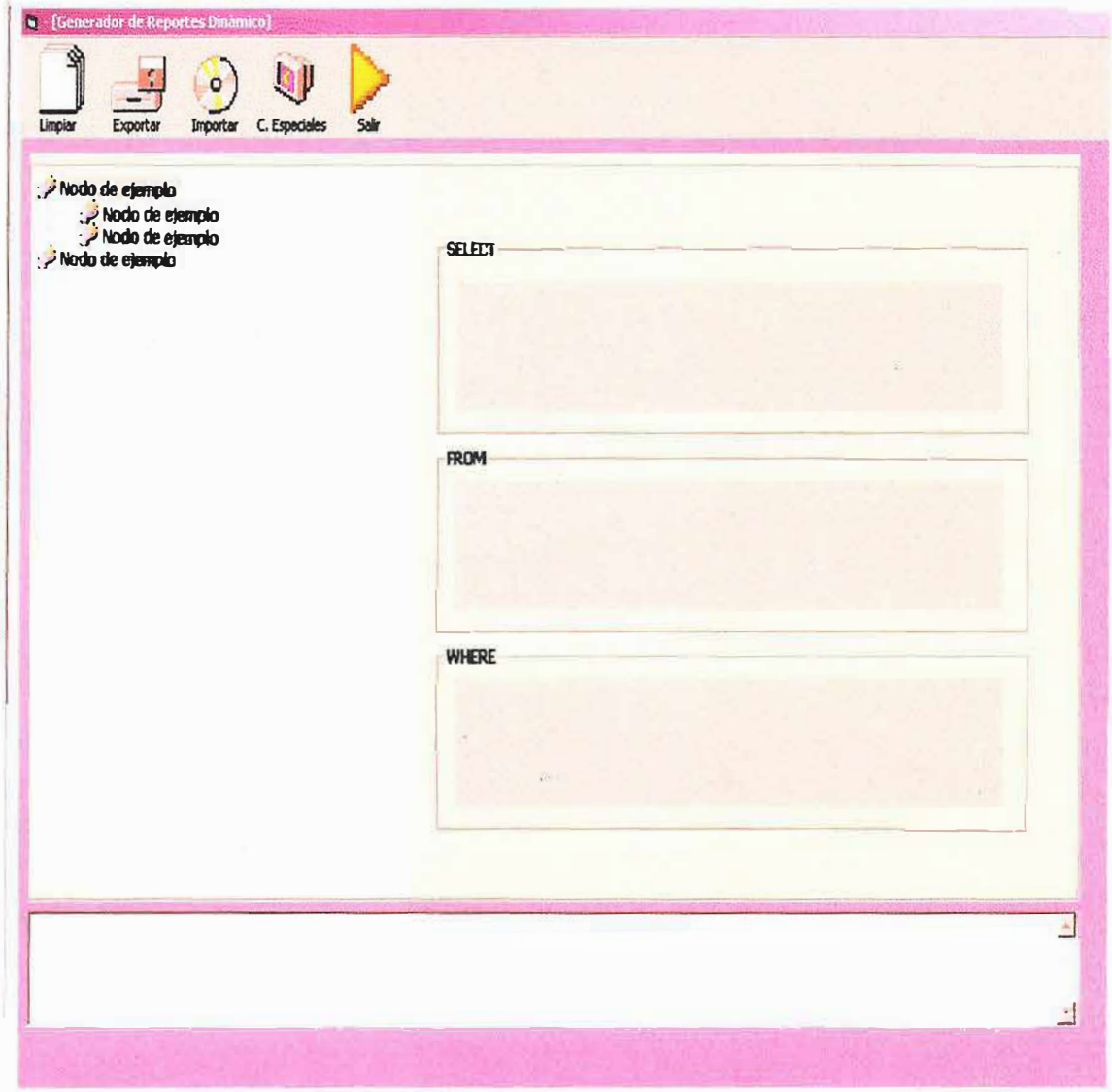


En este formulario se registran todos los accesos y procesos que se realizan en el sistema .

CREACION DE USUARIOS

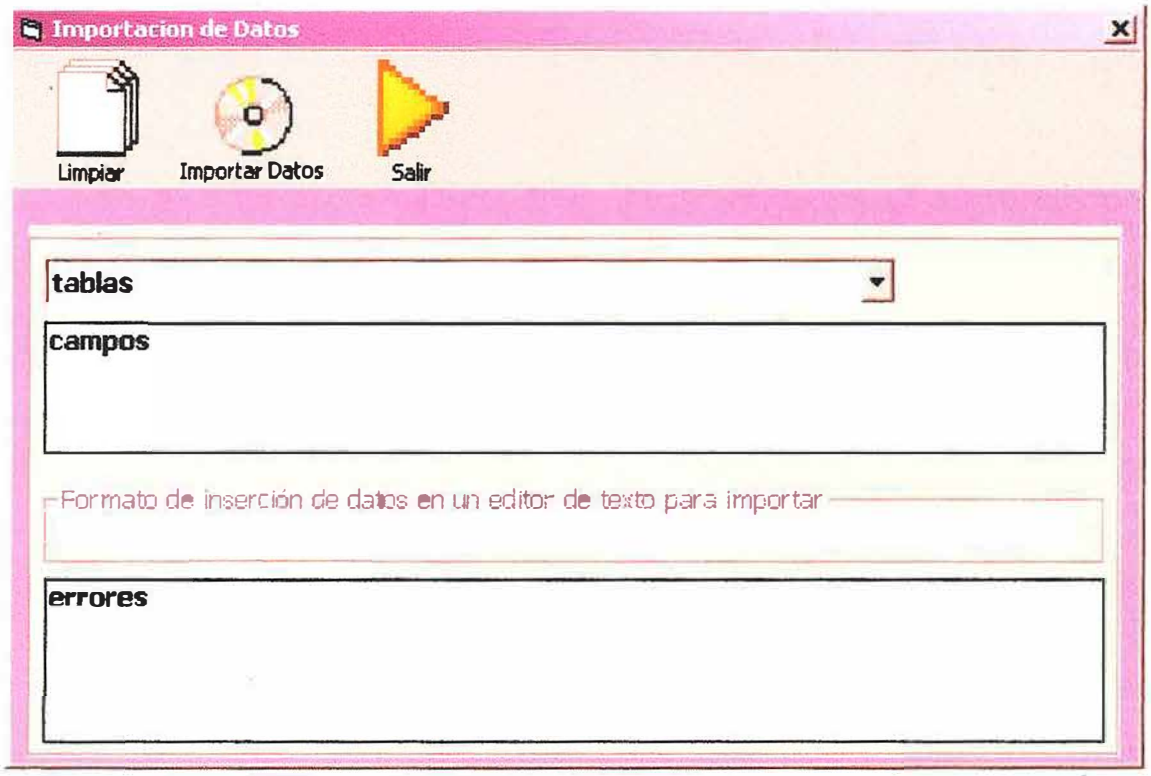
En este formulario se pueden crear los usuarios que van a usar el sistema de tal forma se pueden tambien crear usuarios de dos tipos administradores y usuarios con restricciones.

GENERADOR DE REPORTES DINÁMICOS



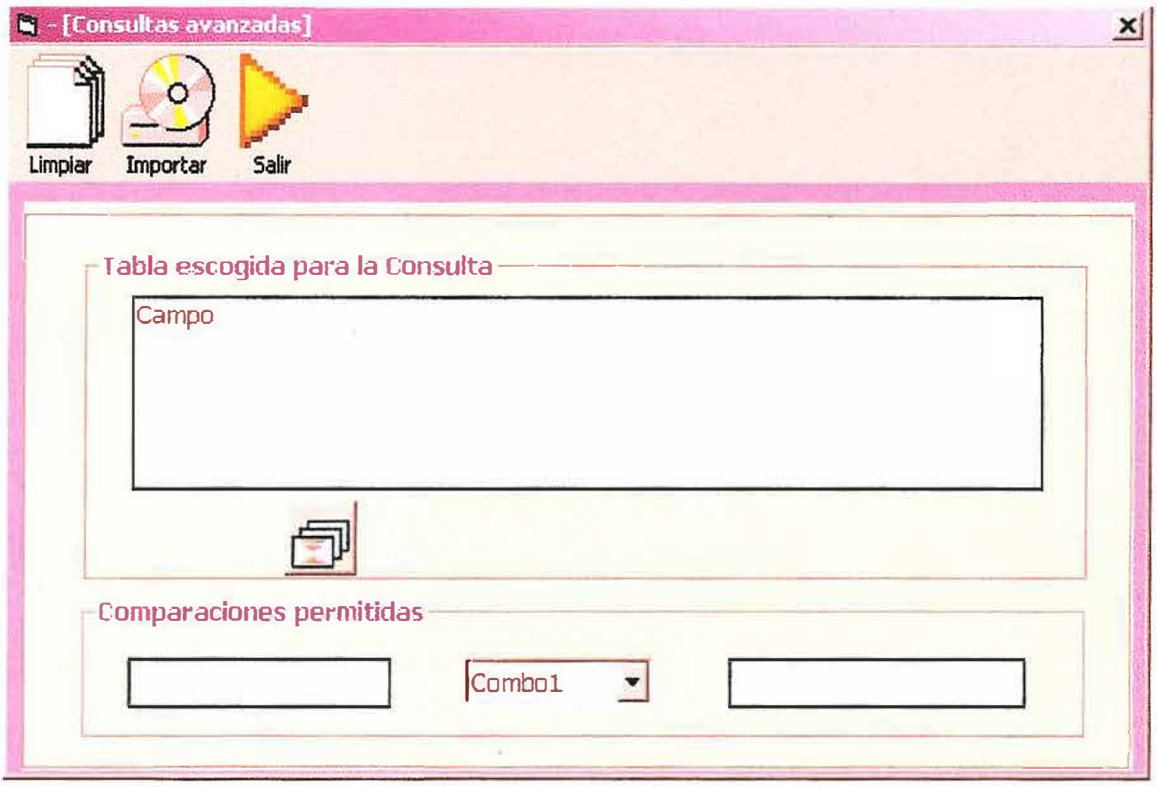
En este formulario se pueden generar los reportes dinámicos, con los campos deseados consultar en el lado izquierdo nos muestra un árbol jerárquico donde se podrán seleccionar las tablas y los campos a consultar.

IMPORTACIÓN DE DATOS



En este formulario se podrán importar datos a un editor de texto en donde se podrá leer la información guardada en las tablas y los campos seleccionados.

CONSULTA AVANZADA SQL

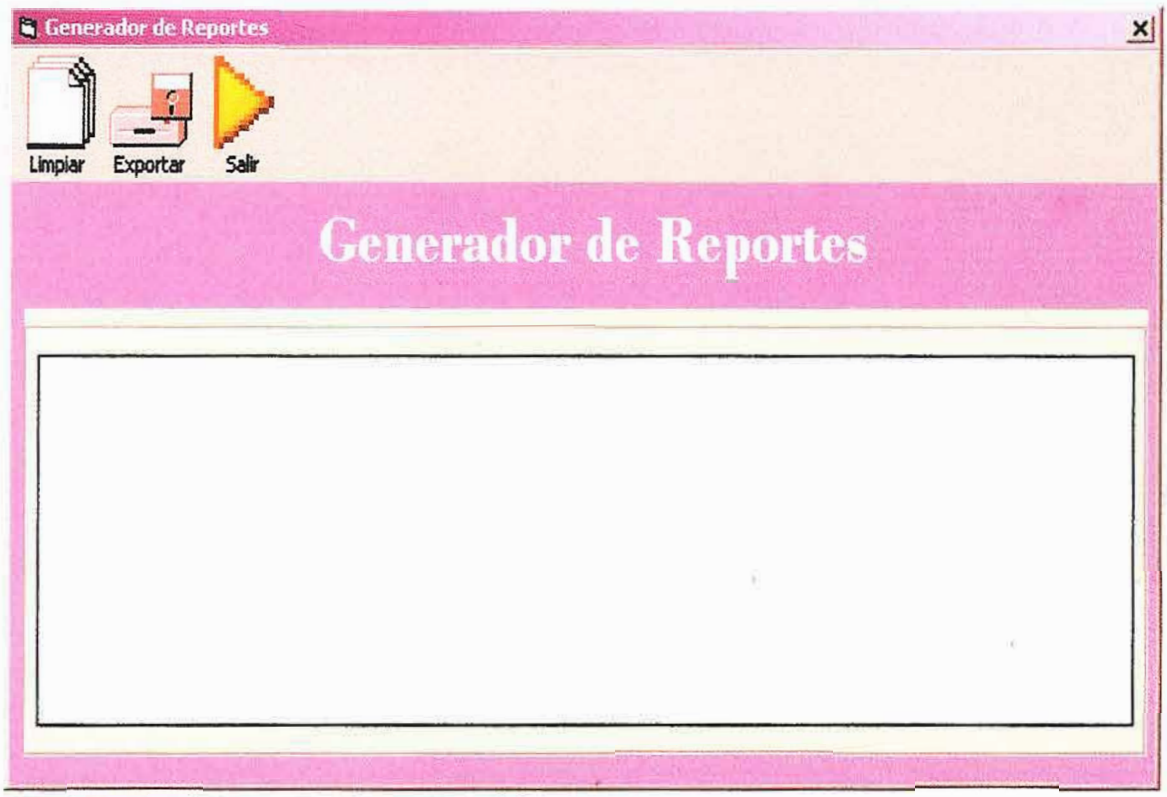


UNIVERSIDAD SIMON RODRIGUEZ
HEMEROTECA

REGISTROS DE LOGROS

En este formulario se registran los logros asignándolos respectivamente a su asignatura, curso y grupo.

GENERADOR DE REPORTES



13. DESARROLLO Y PRUEBA DEL SISTEMA.

CODIGO FUENTE DOCUMENTADO.

1. CODIGO FUENTE DEL MODULO INGRESO DE USUARIO

```
Private Sub Command1_Click()

pws = Encryptacion("keykeykeyk", Text2.text, ENCRYPT)
Set rs = Nothing
cmd.CommandText = "select * from usuarios where login ='" +
Trim(Text1.text) + "' and  pasword = '" + Trim(pws) + "' and estado = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Module1. auditoria ("Inicio de Sesion"), ("Usuarios")
Menu.Show
Else
MsgBox "USUARIO NO EXISTE"
End If
End Sub

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X
As Single, Y As Single)
   Command1.FontBold = True
   Command1.SetFocus
   Command2.FontBold = False
   Command1.BackColor = &HFFFFFF
   Command2.BackColor = &H8000000F
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X
As Single, Y As Single)
Command2.FontBold = True
```

```
    Command1.SetFocus
    Command1.FontBold = False
    Command2.BackColor = &HFFFFFF
    Command1.BackColor = &H8000000F
End Sub


Private Sub Form_Load()
usu = Text1.text
conex
End Sub
```

2. CODIGO FUENTE DEL MODULO DE MENU DE OPCIONES.

```
Private Sub Areas_Click()
Area.Show 1
End Sub

Private Sub Asi_asin_Click()
Asig_Pen.Show 1
End Sub

Private Sub Asignaturas_Click()
Asignatura.Show 1
End Sub

Private Sub auditoria_Click()
auditoria
End Sub

Private Sub Auditorias_Click()
auditoria.Show 1
End Sub
Private Sub Cursos_Click()
Curso.Show 1
End Sub

Private Sub Docentes_Click()
```

```
Docente.Show 1
End Sub

Private Sub E_I_Click()
GENERAR_REPORTES.Show 1
End Sub

Private Sub Estudiantes_Click()
Estudiante.Show 1
End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub Grupos_Click()
Grupo.Show 1
End Sub

Private Sub Logros_Click()
Logro.Show 1
End Sub

Private Sub Matriculas_Click()
Matricula.Show 1
End Sub

Private Sub Notas_Click()
Registro_Logros.Show 1
End Sub

Private Sub Pensum1_Click()
Pensun.Show 1
End Sub

Private Sub Profesiones_Click()
Profesion.Show 1
End Sub
```

```
Private Sub salirsistema_Click()
End
End Sub

Private Sub Sesion_Click()
Unload Me
Ingreso.Visible = True
End Sub
```

3. CODIGO FUENTE DEL MODULO DE REGISTRO DE ESTUDIANTES Y ACUDIENTES.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' limpiar todos los estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarEst()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text7.text = ""
Text8.text = ""
Text1.SetFocus
VsEst.Clear
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' limpiar todos los Acudientes
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarAcu()
```

```
Text9.text = ""
Text10.text = ""
Text11.text = ""
Text12.text = ""
Text13.text = ""
Text14.text = ""
Text15.text = ""
Text16.text = ""
Text9.SetFocus
VsAcu.Clear
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos los estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarEst()
Set rs = Nothing
cmd.CommandText = "select * from estudiantes where codigoest = '" +
Trim(Text1.text) + "' and estadoest = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Estudiantes values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "'," & _
"'" + Trim(Text3.text) + "','" + Trim(Text4.text) + "','" + Trim(Text5.text) +
"'," & _
"'" + Trim(Text6.text) + "','" + Trim(Text7.text) + "','" + Trim(Text8.text) +
"'," & _
"'" + Trim(Combo1.text) + "','" + Trim(DTPicker1.Value) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Estudiante Ingresado Satisfactoriamente...", vbExclamation,
"Estudiantes"
Module1.auditoria ("Inserto"), ("Estudiantes")
Else
MsgBox "El Estudiante ya existe...", vbInformation, "Estudiantes"
End If
End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos los acudientes
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarAcu()
Set rs = Nothing
cmd.CommandText = "select * from Estudiantes where CodigoEst = '" +
Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Acudientes values ('" + Trim(Text1.text) +
"','" + Trim(Text9.text) + "','" + Trim(Text10.text) + "'," & _
"'" + Trim(Text11.text) + "','" + Trim(Text12.text) + "','" + Trim(Text13.text)
+ "'," & _
"'" + Trim(Text14.text) + "','" + Trim(Text15.text) + "','" + Trim(Text16.text)
+ "'," & _
"'" + Trim(Combo2.text) + "','" + Trim(DTPicker2.Value) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Acudiente Ingresado Satisfactoriamente...", vbExclamation,
"Acudiente"
Module1.auditoria ("Inserto"), ("Acudientes")
Else
MsgBox "El Acudiente No Tiene Estudiante Asociado...", vbInformation,
"Acudiente"
End If
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos los estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarEst()
Set rs = Nothing
```

```
cmd.CommandText = "select * from Estudiantes where CodigoEst = '" +
Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!CedulaEst
Text3.text = rs!ExpedidaEst
Text4.text = rs!NombresEst
Text5.text = rs!ApellidosEst
Text6.text = rs!EdadEst
Text7.text = rs!DireccionEst
Text8.text = rs!TelefonoEst
Combo1.text = rs!SexoEst
DTPicker1.Value = rs!FechaNEst
Module1.auditoria ("Consulto"), ("Estudiantes")
Else
MsgBox "El Estudiante no existe...", vbInformation, "Estudiantes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos los Acudientes
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarAcu()
Set rs = Nothing
cmd.CommandText = "select * from Acudientes where CodigoAcu = '" +
Trim(Text9.text) + "' and EstadoAcu = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text10.text = rs!CedulaAcu
Text11.text = rs!ExpedidaAcu
Text12.text = rs!NombresAcu
Text13.text = rs!ApellidosAcu
Text14.text = rs!EdadAcu
Text15.text = rs!DireccionAcu
Text16.text = rs!TelefonoAcu
Combo2.text = rs!SexoAcu
DTPicker2.Value = rs!FechaNAcu
Module1.auditoria ("Consulto"), ("Acudientes")
```

```
Else
MsgBox "El Acudiente no existe...", vbInformation, "Acudiente"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''' Modificar todos los Estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarEst()
Set rs = Nothing
cmd.CommandText = "select * from Estudiantes where CodigoEst = '" +
Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Estudiantes set CedulaEst = '" +
Trim(Text2.text) + "', ExpedidaEst = '" + Trim(Text3.text) + "'," & _
" NombresEst = '" + Trim(Text4.text) + "', ApellidosEst = '" +
Trim(Text5.text) + "', EdadEst = '" + Trim(Text6.text) + "'," & _
" DireccionEst = '" + Trim(Text7.text) + "', TelefonoEst = '" +
Trim(Text8.text) + "', SexoEst ='" + Trim(Combo1.text) + "'," & _
" FechaNEst = '" + Trim(DTPicker1.Value) + "' where CodigoEst = '" +
Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Estudiante Modificado Satisfactoriamente...", vbExclamation,
"Estudiantes"
Module1.auditoria ("Modifico"), ("Estudiantes")
Else
MsgBox "El Estudiante no existe...", vbInformation, "Estudiantes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''' Modificar todos los Acudientes
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarAcu()
Set rs = Nothing
```

```
cmd.CommandText = "select * from Acudientes where CodigoAcu = '" +
Trim(Text9.text) + "' and EstadoAcu = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Acudientes set CedulaAcu = '" +
Trim(Text10.text) + "', ExpedidaAcu = '" + Trim(Text11.text) + "'," & _
" NombresAcu = '" + Trim(Text12.text) + "', ApellidosAcu = '" +
Trim(Text13.text) + "', EdadAcu = '" + Trim(Text14.text) + "'," & _
" DireccionAcu = '" + Trim(Text15.text) + "', TelefonoAcu = '" +
Trim(Text16.text) + "', SexoAcu ='" + Trim(Combo2.text) + "'," & _
" FechaNAcu = '" + Trim(DTPicker2.Value) + "' where CodigoAcu = '" +
Trim(Text9.text) + "' and EstadoAcu = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Acudiente Modificado Satisfactoriamente...", vbExclamation,
"Acudientes"
Module1.auditoria ("Modifico"), ("Acudientes")
Else
MsgBox "El Acudiente no existe...", vbInformation, "Acudientes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Eliminar todos los Estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarEst()
Set rs = Nothing
cmd.CommandText = "select * from Estudiantes where CodigoEst = '" +
Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar El Estudiante...", vbYesNo, "Estudiante")
If res = vbYes Then
cmd.CommandText = "update Estudiantes set EstadoEst = '0' where
CodigoEst = '" + Trim(Text1.text) + "' and EstadoEst = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
MsgBox "Estudiante Eliminado Satisfactoriamente...", vbExclamation,
"Estudiantes"
Module1.auditoria ("Elimino"), ("Estudiantes")
End If
Else
MsgBox "El Estudiante no existe...", vbInformation, "Estudiantes"
End If
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Eliminar todos los Acudientes
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarAcu()
Set rs = Nothing
cmd.CommandText = "select * from Acudientes where CodigoAcu = '" +
Trim(Text9.text) + "' and EstadoAcu = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar El Acudiente...", vbYesNo, "Acudientes")
If res = vbYes Then
cmd.CommandText = "update Acudientes set EstadoAcu = '0' where
CodigoAcu = '" + Trim(Text9.text) + "' and EstadoAcu = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Acudiente Eliminado Satisfactoriamente...", vbExclamation,
"Acudientes"
Module1.auditoria ("Elimino"), ("Acudientes")
End If
Else
MsgBox "El Acudiente no existe...", vbInformation, "Acudientes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar en la Grilla todos los Estudiantes
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConGrillaEst()
presEst
Set rs = Nothing
```

```
cmd.CommandText = "select * from Estudiantes"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
i = 1
cont = cont + 1
While Not rs.EOF
VsEst.TextMatrix(i, 0) = cont
VsEst.TextMatrix(i, 1) = rs!CodigoEst
VsEst.TextMatrix(i, 2) = rs!CedulaEst
VsEst.TextMatrix(i, 3) = rs!ExpedidaEst
VsEst.TextMatrix(i, 4) = rs!NombresEst
VsEst.TextMatrix(i, 5) = rs!ApellidosEst
VsEst.AutoSize (0)
VsEst.AutoSize (1)
VsEst.AutoSize (2)
VsEst.AutoSize (3)
VsEst.AutoSize (4)
VsEst.AutoSize (5)
i = i + 1
cont = cont + 1
VsEst.Rows = VsEst.Rows + 1
rs.MoveNext
Wend
End Sub
```

```
''''''''''''''''''''''''''''''''''''''''
'''' Consultar en la Grilla todos los Acudientes
''''''''''''''''''''''''''''''''''''''''
Private Sub ConGrillaAcu()
presAcu
Set rs = Nothing
cmd.CommandText = "select * from Acudientes"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
i = 1
cont = cont + 1
While Not rs.EOF
VsAcu.TextMatrix(i, 0) = cont
VsAcu.TextMatrix(i, 1) = rs!CodigoAcu
VsAcu.TextMatrix(i, 2) = rs!CodigoAcu
VsAcu.TextMatrix(i, 3) = rs!CedulaAcu
VsAcu.TextMatrix(i, 4) = rs!ExpedidaAcu
VsAcu.TextMatrix(i, 5) = rs!NombresAcu
VsAcu.TextMatrix(i, 6) = rs!ApellidosAcu
VsAcu.AutoSize (0)
VsAcu.AutoSize (1)
VsAcu.AutoSize (2)
VsAcu.AutoSize (3)
VsAcu.AutoSize (4)
VsAcu.AutoSize (5)
VsAcu.AutoSize (6)
i = i + 1
cont = cont + 1
VsAcu.Rows = VsAcu.Rows + 1
rs.MoveNext
Wend
End Sub

Private Sub presEst()
VsEst.TextMatrix(0, 0) = "No"
VsEst.TextMatrix(0, 1) = "Codigo"
VsEst.TextMatrix(0, 2) = "Cedula"
VsEst.TextMatrix(0, 3) = "Expedida"
VsEst.TextMatrix(0, 4) = "Nombres"
VsEst.TextMatrix(0, 5) = "Apellidos"
End Sub

Private Sub presAcu()
VsAcu.TextMatrix(0, 0) = "No"
VsAcu.TextMatrix(0, 1) = "Codigo Estudiante"
VsAcu.TextMatrix(0, 2) = "Codigo"
VsAcu.TextMatrix(0, 3) = "Cedula"
```

```
VsAcu.TextMatrix(0, 4) = "Expedida"
VsAcu.TextMatrix(0, 5) = "Nombres"
VsAcu.TextMatrix(0, 6) = "Apellidos"
End Sub

Private Sub Form_Load()
presEst
presAcu
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarEst
Case 2
GuardarEst
ConGrillaEst
Case 3
ConsultarEst
ConGrillaEst
Case 4
ModificarEst
ConGrillaEst

Case 5
EliminarEst
ConGrillaEst
Case 6
reportesest
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub

Private Sub reportesest()
```

```
CrystalReport1.WindowState = crptMaximized
CrystalReport1.ReportFileName = "D:\CODIGOS VISUAL\Proyectos
U\Isabel Lopez\Reportes\estudiantes.rpt"
CrystalReport1.Action = 1
End Sub

Private Sub reportesacu()
CrystalReport2.WindowState = crptMaximized
CrystalReport2.ReportFileName = "D:\CODIGOS VISUAL\Proyectos
U\Isabel Lopez\Reportes\acudiantes.rpt"
CrystalReport2.Action = 1
End Sub

Private Sub Toolbar2_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarAcu
Case 2
GuardarAcu
ConGrillaAcu
Case 3
ConsultarAcu
ConGrillaAcu
Case 4
ModificarAcu
ConGrillaAcu
Case 5
EliminarAcu
ConGrillaAcu
Case 6
reportesacu
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

4. CODIGO FUENTE DEL MODULO REGISTRO DE DOCENTES.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Limpia todos los Docentes
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarDoc()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text7.text = ""
Text1.SetFocus
End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Adicionar todos los Docentes
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarDoc()
Set rs = Nothing
cmd.CommandText = "select * from Docentes where CedulaDoc = '" +
Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Docentes values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "','" & _
```

```
"'" + Trim(Text3.text) + "','" + Trim(Text4.text) + "','" + Trim(Text5.text) +
"'," & _
"'" + Trim(Text6.text) + "','" + Trim(Text7.text) + "','" + Trim(Combo1.text) +
"'," & _
"'" + Trim(DTPicker1.Value) + "','" + Trim(Combo2.text) + "','" +
Trim(Combo3.text) + "'," & _
"'" + Trim(Combo4.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Docente Ingresado Satisfactoriamente...", vbExclamation,
"Docentes"
Module1.auditoria ("Inserto"), ("Docentes")
Else
MsgBox "El Docente ya existe...", vbInformation, "Docentes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos los Docentes
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarDoc()
Set rs = Nothing
cmd.CommandText = "select * from Docentes where CedulaDoc = '" +
Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombresDoc
Text3.text = rs!ApellidosDoc
Text4.text = rs!DireccionDoc
Text5.text = rs!TelefonoDoc
Text6.text = rs!ExpedidaDoc
Text7.text = rs!EdadDoc
Combo1.text = rs!SexoDoc
DTPicker1.Value = rs!FechaNDoc
Module1.auditoria ("Consulto"), ("Docentes")
Else
MsgBox "El Docente no existe...", vbInformation, "Docentes"
End If
End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''
'''' Modificar todos los Docentes
''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarDoc()
Set rs = Nothing
cmd.CommandText = "select * from Docentes where CedulaDoc = '" +
Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Docentes set NombresDoc = '" +
Trim(Text2.text) + "', ApellidosDoc = '" + Trim(Text3.text) + "'," & _
"DireccionDoc = '" + Trim(Text4.text) + "', TelefonoDoc = '" +
Trim(Text5.text) + "', ExpedidaDoc = '" + Trim(Text6.text) + "'," & _
"EdadDoc = '" + Trim(Text7.text) + "', SexoDoc = '" + Trim(Combo1.text) +
"', FechaNDoc ='" + Trim(DTPicker1.Value) + "'," & _
"AsignaturaDoc ='" + Trim(Combo2.text) + "', CursoDoc='" +
Trim(Combo3.text) + "',GradoDoc='" + Trim(Combo4.text) + "' " & _
"where CedulaDoc = '" + Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Docente Modificado Satisfactoriamente...", vbExclamation,
"Docentes"
Module1.auditoria ("Modifico"), ("Docentes")
Else
MsgBox "El Docente no existe...", vbInformation, "Docentes"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''
'''' Eliminar todos los Docentes
''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarDoc()
Set rs = Nothing
cmd.CommandText = "select * from Docentes where CedulaDoc = '" +
Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
```

UNIVERSIDAD SIMON
HEMEROTECA

```
Set rs = Nothing
res = MsgBox("Desea Eliminar el Docente ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Docentes set EstadoDoc = '0' where CedulaDoc
= '" + Trim(Text1.text) + "' and EstadoDoc = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Docente Eliminado Satisfactoriamente...", vbExclamation,
"Docentes"
Module1.auditoria ("Elimino"), ("Docentes")
End If
Else
MsgBox "El Docente no existe...", vbInformation, "Docentes"
End If
End Sub

Private Sub cargAsig()
Set rs = Nothing
cmd.CommandText = "select NombreAsig from Asignaturas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Las Asignaturas...", vbInformation, "Asignaturas del
Pensum"
End If
End Sub

Private Sub cargCur()
Set rs = Nothing
cmd.CommandText = "select NombreCur from Cursos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo3.AddItem rs.Fields(0)
rs.MoveNext
```

```
Wend
Else
MsgBox "Debe Cargar Los Cursos...", vbInformation, "Asignaturas del
Pensum"
End If
End Sub


Private Sub cargGru()
Set rs = Nothing
cmd.CommandText = "select NombreGru from grupos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo4.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Los Grupos...", vbInformation, "Asignaturas del
Pensum"
End If
End Sub

Private Sub Form_Load()
cargAsig
cargGru
cargCur
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarDoc
Case 2
GuardarDoc
Case 3
ConsultarDoc
Case 4
```

```
ModificarDoc
Case 5
EliminarDoc
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

5. CODIGO FUENTE DEL MODULO REGISTRO DEL CURSO.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Limpia todos las Cursos
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarCur()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos las Area
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarCur()
Set rs = Nothing
```

```
cmd.CommandText = "select * from Cursos where CodigoCur = '" +
Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Cursos values ('" + Trim(Text1.text) + "','"
+ Trim(Text2.text) + "'," & _
"'" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Cursos Ingresada Satisfactoriamente...", vbExclamation, "Cursos"
Module1.auditoria ("Inserto"), ("Cursos")
Else
MsgBox "El Cursos ya existe...", vbInformation, "Cursos"
End If
End Sub




''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos las Cursos
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarCur()
Set rs = Nothing
cmd.CommandText = "select * from Cursos where CodigoCur = '" +
Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombreCur
Combo1.text = rs!EstadoCur
Module1.auditoria ("Consulto"), ("Cursos")
Else
MsgBox "La Cursos no existe...", vbInformation, "Cursos"
End If
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
'""" Modificar todos las Cursos
''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarCur()
Set rs = Nothing
cmd.CommandText = "select * from Cursos where CodigoCur = '" +
Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Cursos set NombreCur = '" + Trim(Text2.text)
+ "', EstadoCur = '" + Trim(Combo1.text) + "' where CodigoCur = '" +
Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Cursos Modificada Satisfactoriamente...", vbExclamation, "Cursos"
Module1.auditoria ("Modifico"), ("Cursos")
Else
MsgBox "La Cursos no existe...", vbInformation, "Cursos"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''
'""" Eliminar todos las Cursos
''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarCur()
Set rs = Nothing
cmd.CommandText = "select * from Cursos where CodigoCur = '" +
Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar Las Cursos ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Cursos set EstadoEliCur = '0' where CodigoCur
= '" + Trim(Text1.text) + "' and EstadoEliCur = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Cursos Eliminada Satisfactoriamente...", vbExclamation, "Cursos"
Module1.auditoria ("Elimino"), ("Cursos")
End If
```

```
Else
MsgBox "La Cursos no existe...", vbInformation, "Cursos"
End If
End Sub


Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarCur
Case 2
GuardarCur
Case 3
ConsultarCur
Case 4
ModificarCur
Case 5
EliminarCur
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

6. CODIGO FUENTE DEL MODULO REGISTRO DE LAS AREAS.

```
''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Limpia todos las Areas
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarAre()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub
''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Adicionar todos las Area
```

```
''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarAre()
Set rs = Nothing
cmd.CommandText = "select * from areas where Codigoare = '" +
Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into areas values ('" + Trim(Text1.text) + "','" +
Trim(Text2.text) + "','" & _
"'" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Area Ingresada Satisfactoriamente...", vbExclamation, "Areas"
Module1.auditoria ("Inserto"), ("areas")
Else
MsgBox "El Area ya existe...", vbInformation, "Areas"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos las Areas
''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarAre()
Set rs = Nothing
cmd.CommandText = "select * from Areas where Codigoare = '" +
Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombreAre
Combo1.text = rs!EstadoAre
Module1.auditoria ("Consulto"), ("areas")
Else
MsgBox "La Area no existe...", vbInformation, "Areas"
End If
End Sub
''''''''''''''''''''''''''''''''''''''''
'''' Modificar todos las Areas
''''''''''''''''''''''''''''''''''''''''
```

```
Private Sub ModificarAre()
Set rs = Nothing
cmd.CommandText = "select * from Areas where CodigoAre = '" +
Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Areas set Nombreare = '" + Trim(Text2.text) +
"', EstadoAre = '" + Trim(Combo1.text) + "' where CodigoAre = '" +
Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Areas Modificada Satisfactoriamente...", vbExclamation, "Areas"
Module1.auditoria ("Modifico"), ("areas")
Else
MsgBox "La Areas no existe...", vbInformation, "Areas"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Eliminar todos las Areas
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Eliminarare()
Set rs = Nothing
cmd.CommandText = "select * from Areas where CodigoAre = '" +
Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar Las Areas ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Areas set EstadoEliAre = '0' where CodigoAre
= '" + Trim(Text1.text) + "' and EstadoEliAre = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Areas Eliminada Satisfactoriamente...", vbExclamation, "Areas"
Module1.auditoria ("Elimino"), ("areas")
End If
Else
MsgBox "La Areas no existe...", vbInformation, "Areas"
```

```
End If
End Sub

Private Sub Reportes()
CrystalReport1.WindowState = crptMaximized
CrystalReport1.ReportFileName = "D:\CODIGOS VISUAL\Proyectos
U\Isabel Lopez\Reportes\areas.rpt"
CrystalReport1.Action = 1
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarAre
Case 2
GuardarAre
Case 3
ConsultarAre
Case 4
ModificarAre
Case 5
Eliminarare
Case 6
Reportes
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

7. CODIGO FUENTE DEL MODULO REGISTRO DE ASIGNATURA.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Limpia todos las Asignaturas
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
Private Sub LimpiarAsig()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Adicionar todos las Asignaturas
''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Asignaturas where CodigoAsig = '" +
Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Asignaturas values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "'," & _
"'" + Trim(Combo2.text) + "','" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Asignatura Ingresada Satisfactoriamente...", vbExclamation,
"Asignaturas"
Module1.auditoria ("Inserto"), ("asignaturas")
Else
MsgBox "La Asignatura ya existe...", vbInformation, "Asignaturas"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Consultar todos las Asignaturas
''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Asignaturas where CodigoAsig = '" +
Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombreAsig
```

```
Combo2.text = rs!AreaAsig
Combo1.text = rs!EstadoAsig
Module1.auditoria ("Consulto"), ("asignaturas")
Else
MsgBox "La Asignatura no existe...", vbInformation, "Asignaturas"
End If
End Sub
```

```
''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Modificar todos las Asignaturas
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Asignaturas where CodigoAsig = '" +
Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Asignaturas set NombreAsig = '" +
Trim(Text2.text) + "', AreaAsig = '" + Trim(Combo2.text) + "'," & _
"EstadoAsig = '" + Trim(Combo1.text) + "' where CodigoAsig = '" +
Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Asignatura Modificada Satisfactoriamente...", vbExclamation,
"Asignaturas"
Module1.auditoria ("Modifico"), ("asignaturas")
Else
MsgBox "La Asignatura no existe...", vbInformation, "Asignaturas"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
```

```
"''' Eliminar todos las Asignaturas
''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Asignaturas where CodigoAsig = '" +
Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar La Asignatura ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Asignaturas set EstadoEliAsig = '0' where
CodigoAsig = '" + Trim(Text1.text) + "' and EstadoEliAsig = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Asignatura Eliminada Satisfactoriamente...", vbExclamation,
"Asignaturas"
Module1.auditoria ("Elimino"), ("asignaturas")
End If
Else
MsgBox "La Asignatura no existe...", vbInformation, "Asignaturas"
End If
End Sub

Private Sub cargarAreas()
Set rs = Nothing
cmd.CommandText = "select nombreare from areas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
End Sub


Private Sub Form_Load()
cargarAreas
End Sub
```

```
Private Sub Reportes()
CrystalReport1.WindowState = crptMaximized
CrystalReport1.ReportFileName = "D:\CODIGOS VISUAL\Proyectos
U\Isabel Lopez\Reportes\asignaturas.rpt"
CrystalReport1.Action = 1
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarAsig
Case 2
GuardarAsig
Case 3
ConsultarAsig
Case 4
ModificarAsig
Case 5
EliminarAsig
Case 6
Reportes
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

8. COFIDO FUENTE DEL MODULO REGISTRO DE ASIGNATURAS AL PENSUM.

```
Private Sub cargAsig()
Set rs = Nothing
cmd.CommandText = "select NombreAsig from Asignaturas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
```

```
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Las Asignaturas...", vbInformation, "Asignaturas del
Pensum"
End If
End Sub

Private Sub pres()
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Codigo"
vs.TextMatrix(0, 2) = "Nombre"
End Sub

Private Sub Command1_Click()
Consultar_Asig.Show 1
End Sub

Private Sub Command2_Click()
i = vs.Rows
Set rs = Nothing
cmd.CommandText = "select * from asignaturas where NombreAsig = '" +
Trim(Combo2.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
vs.TextMatrix(i - 1, 1) = rs!CodigoAsig
vs.TextMatrix(i - 1, 2) = rs!NombreAsig
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
End If
vs.Rows = vs.Rows + 1
End Sub

Private Sub Guardar()
A = 0
```

```
i = 1
While A <> vs.Rows - 2
Set rs = Nothing
cmd.CommandText = "insert into asignaturas_pensum values ('" +
Trim(Text1.text) + "','" + Trim(Text2.text) + "'," & _
"'" + Trim(vs.TextMatrix(i, 1)) + "','" + Trim(vs.TextMatrix(i, 2)) + "')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
A = A + 1
i = i + 1
Wend
MsgBox "Asignaturas del Pensum Adicionadas...", vbExclamation,
"Asignatura de Pensum"
Module1.auditoria ("Inserto"), ("asignaturas_pensum")
End Sub

Private Sub limpiar()
Text1.text = ""
Text2.text = ""
vs.Clear
pres
Text1.SetFocus
End Sub

Private Sub Buscar()
Set rs = Nothing
cmd.CommandText = "select * from asignaturas_pensum where codigopen =
'" + Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombrePen
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CodigoAsig
vs.TextMatrix(i, 2) = rs!NombreAsig
i = i + 1
cont = cont + 1
```

```
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Module1.auditoria ("Consulto"), ("asignaturas_pensum")
Else
MsgBox "Las Asignaturas no han sido agrrgadas al Pensum", vbInformation,
"Asignatura de Pensum"
End If
End Sub

Private Sub Eliminar()
Set rs = Nothing
cmd.CommandText = "select * from asignaturas_pensum where codigopen =
"' + Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
res = MsgBox("Desea Eliminar Las asignaturas del Pensum..?", vbYesNo,
"Asignatura de Pensum")
If res = vbYes Then
Set rs = Nothing
cmd.CommandText = "delete from asignaturas_pensum where codigopen = '"
+ Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Asignaturas del Pensum Adicionadas...", vbExclamation,
"Asignatura de Pensum"
Module1.auditoria ("Elimino"), ("asignaturas_pensum")
End If
Else
MsgBox "Las Asignaturas no han sido agrrgadas al Pensum", vbInformation,
"Asignatura de Pensum"
End If
End Sub

Private Sub Form_Load()
tip = 1
cargAsig
pres
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
limpiar
Case 2
Guardar
Case 3
Buscar
Case 4
Eliminar
Case 6
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

9. CODIGO FUENTE DEL MODULO CREACION DE USUARIO.

```
Private Sub Guardar()
pws = Encryptacion("keykeykeyk", Text8.text, ENCRYPT)
Set rs = Nothing
cmd.CommandText = "select * from Usuarios where Cedula = '" +
Trim(Text1.text) + "' and estado = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Usuarios Values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "'," & _
"'" + Trim(Text3.text) + "','" + Trim(Text4.text) + "','" + Trim(Text5.text) +
"'," & _
"'" + Trim(Text6.text) + "','" + Trim(Text7.text) + "','" + Trim(pws) + "'," & _
"'" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
MsgBox "Usuario Adicionado Satisfactoriamente", vbExclamation, "Creacion
de Usuarios"
Module1.auditoria ("Inserto"), ("Usuarios")
Else
MsgBox "El Usuario ya existe..", vbInformation, "Creacion de Usuarios"
End If
End Sub

Private Sub Consultar()
Set rs = Nothing
cmd.CommandText = "select * from usuarios where Cedula = '" +
Trim(Text1.text) + "' and estado = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!Nombre
Text3.text = rs!Apellidos
Text4.text = rs!Edad
Text5.text = rs!Direccion
Text6.text = rs!Telefono
Text7.text = rs!Login
Text8.text = rs!Pasword
Combo1.text = rs!tipo_usuario
Module1.auditoria ("Consulto"), ("Usuarios")
Else
MsgBox "El Usuario ya existe..", vbInformation, "Creacion de Usuarios"
End If
End Sub

Private Sub CambiarCon()
pws = Encryptacion("keykeykeyk", Text8.text, ENCRYPT)
Set rs = Nothing
cmd.CommandText = "select * from usuarios where Cedula = '" +
Trim(Text1.text) + "' and estado = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
  If rs!tipo_usuario = Combo1.text Then
      If Not rs.EOF Then
      Set rs = Nothing
```

```
        cmd.CommandText = "update Usuarios set Pasword = '" + Trim(pws) +
"' where Cedula = '" + Trim(Text1.text) + "' and estado = '1'"
        rs.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
        MsgBox "Contraseña Cambiada Satisfactoriamente", vbExclamation,
"Creacion de Usuarios"
        Module1.auditoria ("Modifico"), ("Usuarios")
        Else
        MsgBox "El Usuario ya existe..", vbInformation, "Creacion de Usuarios"
        End If
   Else
   MsgBox "El Tipo de Usuario debe ser Igual...", vbInformation, "Creacion de
Usuarios"
   End If
End Sub


Private Sub Eliminar()
Set rs = Nothing
cmd.CommandText = "select * from usuarios where Cedula = '" +
Trim(Text1.text) + "' and estado = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
 If Combo1.text <> "Administrador" Then
        If Not rs.EOF Then
        Set rs = Nothing
        cmd.CommandText = "update usuarios set estado = '0' " ' where Cedula =
'" + Trim(Text1.text) + "' and estado = '1'"
        rs.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
        MsgBox "Usuario Eliminado Satisfactoriamente", vbExclamation,
"Creacion de Usuarios"
        Module1.auditoria ("Elimino"), ("Usuarios")
        Else
        MsgBox "El Usuario ya existe..", vbInformation, "Creacion de Usuarios"
        End If
   Else
```

```
  MsgBox "El Usuario no puede ser Eliminado por ser Administrador...",
vbInformation, "Creacion de Usuarios"
  End If
End Sub

Private Sub limpiar()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text7.text = ""
Text8.text = ""
Combo1.text = ""
Text1.SetFocus
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
Limpíar
Case 2
Guardar
Case 3
Consultar
Case 4
CambiarCon
Case 5
Case Eliminar
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

10. CODIGO FUENTE DEL MODULO CONSULTAR ASIGNATURAS.

```
Private Sub cargAsig()
Set rs = Nothing
cmd.CommandText = "select * from Pensum"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CodigoPen
vs.TextMatrix(i, 2) = rs!NombrePen
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
i = i + 1
cont = cont + 1
rs.MoveNext
vs.Rows = vs.Rows + 1
Wend
End Sub

Private Sub Form_Load()
cargAsig
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Codigo"
vs.TextMatrix(0, 2) = "Nombre"
End Sub

Private Sub vs_Click()
If tip = 1 Then
d = vs.Row
Asig_Pen.Text1.text = vs.TextMatrix(d, 1)
Asig_Pen.Text2.text = vs.TextMatrix(d, 2)
Unload Me
Asig_Pen.Visible = True
```

```
End If

If tip = 2 Then
d = vs.Row
Logro.Text1.text = vs.TextMatrix(d, 1)
Logro.Text2.text = vs.TextMatrix(d, 2)
Unload Me
Logro.Visible = True
End If
End Sub
```

11. CODIGO FUENTE DEL MODULO REGISTRO DE AUDITORIA.

```
 Private Sub cargarusuarios()
Set rs = Nothing
cmd.CommandText = "select login from usuarios"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
Combo1.AddItem rs.Fields(0)
rs.MoveNext
Wend
End Sub

Private Sub cargartransacciones()
Set rs = Nothing
cmd.CommandText = "select distinct descripcion from auditoria"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
End Sub

Private Sub cargartablas()
Set rs = Nothing
cmd.CommandText = "SHOW TABLES FROM `Isabel_lopez`"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
If rs.Fields(0) <> "auditoria" Then
```

```
Combo3.AddItem rs.Fields(0)
End If
rs.MoveNext
Wend
End Sub

Private Sub Combo1_Click()
Set rs = Nothing
cmd.CommandText = "select * from auditoria where usuario = '" +
Trim(Combo1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!descripcion
vs.TextMatrix(i, 2) = rs!usuario
vs.TextMatrix(i, 3) = rs!tabla
vs.TextMatrix(i, 4) = rs!fecha
vs.TextMatrix(i, 5) = rs!hora
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No existen Registros...!", vbInformation, "Buscando"
End If
End Sub

Private Sub Combo2_Click()
```

```
Set rs = Nothing
cmd.CommandText = "select * from auditoria where descripcion = '" +
Trim(Combo2.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!descripcion
vs.TextMatrix(i, 2) = rs!usuario
vs.TextMatrix(i, 3) = rs!tabla
vs.TextMatrix(i, 4) = rs!fecha
vs.TextMatrix(i, 5) = rs!hora
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No existen Registros...!", vbInformation, "Buscando"
End If

End Sub

Private Sub Command6_Click()
Frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = False
vs.Clear
```

```
pre
vs.Rows = 2
End Sub

Private Sub Command7_Click()
Unload Me
End Sub

Private Sub Form_Load()
cargartransacciones
cargartablas
cargarusuarios
pre
Frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = False
End Sub

Private Sub pre()
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Descripcion"
vs.TextMatrix(0, 2) = "Tablas"
vs.TextMatrix(0, 3) = "Usuarios"
vs.TextMatrix(0, 4) = "Fecha"
vs.TextMatrix(0, 5) = "Hora"
End Sub

Private Sub Combo3_Click()
Set rs = Nothing
cmd.CommandText = "select * from auditoria where tabla = '" +
Trim(Combo3.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
```

```
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!descripcion
vs.TextMatrix(i, 2) = rs!usuario
vs.TextMatrix(i, 3) = rs!tabla
vs.TextMatrix(i, 4) = rs!fecha
vs.TextMatrix(i, 5) = rs!hora
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No existen Registros...!", vbInformation, "Buscando"
End If
End Sub

Private Sub Command1_Click()
Frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = False
Set rs = Nothing
cmd.CommandText = "select * from auditoria"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!descripcion
vs.TextMatrix(i, 2) = rs!usuario
```

```
vs.TextMatrix(i, 3) = rs!tabla
vs.TextMatrix(i, 4) = rs!fecha
vs.TextMatrix(i, 5) = rs!hora
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No existen Registros...!", vbInformation, "Buscando"
End If
End Sub

Private Sub Command2_Click()
Frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = True
End Sub

Private Sub Command3_Click()
Frame2.Visible = True
Frame3.Visible = False
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = False
End Sub

Private Sub Command4_Click()
Frame2.Visible = False
Frame3.Visible = True
```

```
Frame4.Visible = False
Frame5.Visible = False
Frame6.Visible = False
End Sub

Private Sub Command5_Click()
Frame2.Visible = False
Frame3.Visible = False
Frame4.Visible = True
Frame5.Visible = True
Frame6.Visible = False
Set rs = Nothing
cmd.CommandText = "select * from auditoria where fecha >= '" +
Trim(DTPicker1.Value) + "' and fecha <= '" + Trim(DTPicker1.Value) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!descripcion
vs.TextMatrix(i, 2) = rs!usuario
vs.TextMatrix(i, 3) = rs!tabla
vs.TextMatrix(i, 4) = rs!fecha
vs.TextMatrix(i, 5) = rs!hora
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No existen Registros...!", vbInformation, "Buscando"
```

```
End If

End Sub
```

12. CODIGO FUENTE DEL MODULOGENERADOR DE REPORTES(Exportacion).

```
Option Explicit
Private Declare Function ShellExecute Lib "shell32.dll" Alias _
"ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation _
As String, ByVal lpFile As String, ByVal lpParameters _
As String, ByVal lpDirectory As String, ByVal nShowCmd _
As Long) As Long
Public cont As Integer
Public conexion As String

Private Sub limpiar()
resul.Clear
resul.Rows = 2
End Sub

Private Sub mare()
Dim titulo As String
Dim i
  Dim fname As String
  Conarchivo1.DialogTitle = "Seleccione el nombre del archivo Reporte"
  Conarchivo1.InitDir = App.Path
  Conarchivo1.Filter = "Archivos de texto (*.txt)|*.txt|Todos los
archivos(*.*)|*.*"
  Conarchivo1.Filter = "Archivos de Texto (*.txt)|*.txt"
  Conarchivo1.DefaultExt = "*.txt"
  Conarchivo1.ShowSave
  fname = Conarchivo1.FileName
  If fname <> vbNullString Then
   Call MEXPORTACION.exprepor(fname)
  End If
 '//////////////////////////
```

```
End Sub

Private Sub Salir_Click()
Unload Me
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
 limpiar
Case 2
 mare
Case 3
 Unload Me
End Select
End Sub
```

13. CODIGO FURNTE DEL MODULO GENERADOR DE REPORTES DINAMICOS.

```
Dim tablas(50) As String
Dim campos(500) As String
Dim llaves(100) As String
Dim h  As Integer
Dim p  As Integer
Dim j  As Integer
Dim vector_codigo(100) As String
Public nomtab1 As String
Public nomtab2 As String
Dim i As Integer

Private Sub ayu001_Click()
 With awd
  .HelpFile = App.Path & "\e\ayuda.hlp"
  .HelpKey = "GENERADOR DE REPORTES"
  .HelpCommand = cdlHelpKey
  .ShowHelp
```

```
 End With
End Sub

Private Sub Check1_Click()
 SQL_Avanzado.tablas = lbfrom.Caption
 If Check1.Value = 1 Then
  If Trim(SQL_Avanzado.tablas) <> "" Then
    SQL_Avanzado.Show vbModal
 End If
 End If
End Sub

Private Sub limpiar()
lbSelect.Caption = ""
lbfrom.Caption = ""
lbwhere.Caption = ""
campostablas = ""
i = 0
While tablas(i) <> "" Or campos(i) <> "" Or llaves(i) <> ""
   tablas(i) = ""
   campos(i) = ""
   llaves(i) = ""
   i = i + 1
Wend
CountCampos = 0
End Sub

Private Sub generar()
If LTrim(lbSelect) = "" Then
   MsgBox "Defina una consulta por favor!!!", vbInformation, "Mensaje"
Else
   If Trim(lbwhere) = "" Then
      query = "Select " + LTrim(lbSelect) + " From " + Trim(lbfrom)
   Else
      query = "SELECT " + LTrim(lbSelect) + " FROM " + Trim(lbfrom) + "
+ " WHERE " + Trim(lbwhere)
   End If
```

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

```
    con.text = query
    cmd.CommandText = query
   reg.Open cmd.CommandText, conn, 2, 4
      ' reg.Open query, conexion, 2, 4
    EXPORTAR_2.resul.Cols = reg.Fields.Count + 1
    EXPORTAR_2.resul.TextMatrix(0, 0) = "N°": cont = 0
    While Not reg.EOF
       For i = 1 To EXPORTAR_2.resul.Cols - 1
          If reg(i - 1) <> "" Then
             If Not IsNull(reg(i - 1)) Then
                EXPORTAR_2.resul.TextMatrix(EXPORTAR_2.resul.Rows - 1,
i) = reg(i - 1)
                j = 0
                EXPORTAR_2.resul.AutoSize (i)
             End If
          End If
       Next i
       cont = cont + 1
      ' EXPORTAR_2.resul.TextMatrix(cont, 0) = cont
        EXPORTAR_2.resul.AutoSize (0)
       EXPORTAR_2.resul.Rows = EXPORTAR_2.resul.Rows + 1
       reg.MoveNext
    Wend
    EXPORTAR_2.resul.Rows = EXPORTAR_2.resul.Rows - 1
    reg.Close

If tab_1 = False Then
    Set reg = Nothing
    cmd.CommandText = "select nomtab1, nomtab2 from referencia"
    reg.Open cmd.CommandText, conn, 2, 4
    i = 1
    cont = 0
    sw = 1
   While Not reg.EOF
    If (nomtab1 = reg!nomtab1 And nomtab2 = reg!nomtab2) Or (nomtab1 =
reg!nomtab2 And nomtab2 = reg!nomtab1) Then
       If EXPORTAR_2.resul.Rows <> 1 Then
        EXPORTAR_2.Show 1
```

```
        sw = 0
       Else
        MsgBox "La consulta Realizada no contiene registros...", vbQuestion,
"Registro no existentes"
        sw = 0
      End If
   End If
    reg.MoveNext
    Wend
    If sw = 1 Then
       MsgBox "La consulta no es permitida por que las tablas no tienen
relacion..", vbQuestion, "Tablas sin relacion"
    End If
   reg.Close
'   regs.Close
Else
If EXPORTAR_2.resul.Rows <> 1 Then
EXPORTAR_2.Show 1
Else
MsgBox "La consulta Realizada no contiene registros...", vbQuestion,
"Registro no existentes"
End If
End If
End If

limpiar
End Sub

Private Sub Command1_Click()
lbSelect.Caption = ""
lbfrom.Caption = ""
lbwhere.Caption = ""
con.text = ""
End Sub

Private Sub Form_Load()
Set conn = Nothing
```

```
conexion = "Provider=MSDASQL.1;Password=123;Persist Security
Info=True;User ID=root;Data Source=Isabel_lopez"
conn.Open conexion
Call carg_tablas_campos
End Sub

Private Sub TreeView1_Collapse(ByVal Node As MSComctlLib.Node)
 TreeView1.Nodes.Item(Node.Index).Image = 1
End Sub
Private Sub TreeView1_Expand(ByVal Node As MSComctlLib.Node)
 TreeView1.Nodes.Item(Node.Index).Image = 1
End Sub
'Al presionar clikc en el treeview se va armando la consulta
Private Sub TreeView1_NodeClick(ByVal Node As MSComctlLib.Node)
If Node.text <> "TABLAS" Then
   If Node.Parent <> "TABLAS" Then
      If lbSelect.Caption = "" Then
         lbSelect.Caption = Node.Parent.Tag & "." & Node.Tag
         campos(0) = Node.Parent.Tag & "." & Node.Tag
         campostablas = campostablas & campos(0)
      Else
         i = 0: sw = 0
         While campos(i) <> "" And sw = 0
            If campos(i) = Node.Parent.Tag & "." & Node.Tag Then
               sw = 1
            End If
            i = i + 1
         Wend
         If sw = 0 Then
            lbSelect.Caption = lbSelect.Caption & " , " & Node.Parent.Tag &
"." & Node.Tag
            campos(i) = Node.Parent.Tag & "." & Node.Tag
            campostablas = campostablas & " " & campos(i)
         End If
      End If
      CountCampos = CountCampos + 1
      EXPORTAR_2.resul.Cols = EXPORTAR_2.resul.Cols + 1
```

```
    EXPORTAR_2.resul.TextMatrix(0, EXPORTAR_2.resul.Cols - 1) =
Trim(Node.text)
    EXPORTAR_2.resul.Cell(flexcpForeColor, 0, EXPORTAR_2.resul.Cols
- 1) = vbBlack
    EXPORTAR_2.resul.Cell(flexcpFontBold, 0, EXPORTAR_2.resul.Cols
- 1) = True
    If lbfrom.Caption = "" Then
      lbfrom.Caption = Node.Parent.Tag
      tab_1 = True
      nomtab1 = lbfrom.Caption
      codtabla(0) = Node.Parent.Tag
      tablas(0) = Node.Parent.Tag
    Else
      i = 0: sw = 0
      While tablas(i) <> "" And sw = 0
        If tablas(i) = Node.Parent.Tag Then
          sw = 1
        End If
        i = i + 1
      Wend
      If sw = 0 Then
        lbfrom.Caption = lbfrom.Caption & " , " & Node.Parent.Tag
        tab_2 = True
        tab_1 = False
        nomtab2 = Node.Parent.Tag
        codtabla(i) = Node.Parent.Tag
        tablas(i) = Node.Parent.Tag
      End If
    End If
    If tablas(0) <> "" Then
      i = 0: sw = 0
      condicion = ""
      While tablas(i) <> "" And sw = 0
        cmd.CommandText = "select * from referencia where (nomtab1='"
+ Node.Parent.Tag + "' and nomtab2='" + tablas(i) + "')" + _
        " or (nomtab2='" + Node.Parent.Tag + "' and nomtab1='" + tablas(i)
+ "')"
```

```
            'cons = "select * from referencia where (nomtab1='" +
Node.Parent.Tag + "' and nomtab2='" + tablas(i) + "')" + _
            "' or (nomtab2='" + Node.Parent.Tag + "' and nomtab1='" + tablas(i)
+ "')"
            reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
            'reg.Open cons, conexion, 2, 4
            If Not reg.EOF Then
               sw = 1
               condicion = reg!nomtab1 + "." + reg!campo1 + "=" +
reg!nomtab2 + "." + reg!campo2
            End If
            reg.Close
            i = i + 1
         Wend
         If condicion <> "" Then
            i = 0: sw = 0
            While llaves(i) <> "" And sw = 0
               If llaves(i) = condicion Then
                  sw = 1
               End If
               i = i + 1
            Wend
            If sw = 0 Then
               If lbwhere.Caption = "" Then
                  llaves(0) = condicion
                  lbwhere.Caption = condicion
               Else
                  llaves(i) = condicion
                  lbwhere.Caption = lbwhere.Caption & " and " & condicion
               End If
            End If
         End If
      End If
   End If
End If
End Sub
Private Sub carg_tablas_campos()
```

```
Dim regcamtab As New ADODB.Recordset
Set Nd = TreeView1.Nodes.Add(, , "R", "TABLAS", 1)
Nd.Expanded = True
CountCampos = 0
p = 1
Set reg = Nothing
'Se consulta las tablas que hay en la base de datos
cmd.CommandText = "select * from aliastablas"
reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
While Not reg.EOF
   Set Nd = TreeView1.Nodes.Add("R", 4, "R" & p, reg!nomtab)
   Nd.Tag = reg!codtab
   h = 1
   'Consultan los campos de la Tabla
   cmd.CommandText = "select * from aliascampos where codtab ='" +
Trim(reg!codtab) + "'"
   regcamtab.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
     While Not regcamtab.EOF
      Set Nd = TreeView1.Nodes.Add("R" & p, 4, "R" & p & "_" & h, "" & ""
& regcamtab!nomcam & "")
       Nd.Tag = regcamtab!codcam
       h = h + 1
       regcamtab.MoveNext
    Wend
   p = p + 1
   regcamtab.Close
   reg.MoveNext
   Wend
reg.Close
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
lbSelect.Caption = ""
lbfrom.Caption = ""
```

```
lbwhere.Caption = ""
con.text = ""
limpiar
Case 2
generar
Case 3
IMPORTAR.Show 1
Case 4
SQL_Avanzado.Show 1
Case 5
res = MsgBox("Desea Salir del Generador de Reportes ...?", vbYesNo,
"Generador de Reportes")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

14. CODIGO FUENTE DEL MODULO REGISTRO DE GRUPOS.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Limpia todos las Grupos
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarGru()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos las Grupos
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarGru()
Set rs = Nothing
cmd.CommandText = "select * from Grupos where CodigoGru = '" +
Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Grupos values ('" + Trim(Text1.text) + "','" 
+ Trim(Text2.text) + "'," & _
"'" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Grupos Ingresada Satisfactoriamente...", vbExclamation, "Grupos"
Module1.auditoria ("Inserto"), ("Grupos")
Else
MsgBox "El Grupos ya existe...", vbInformation, "Grupos"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''
'''' Consultar todos las Grupos
''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarGru()
Set rs = Nothing
cmd.CommandText = "select * from Grupos where CodigoGru = '" +
Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombreGru
Combo1.text = rs!EstadoGru
Module1.auditoria ("Consulto"), ("Grupos")
Else
MsgBox "La Grupos no existe...", vbInformation, "Grupos"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''
'''' Modificar todos las Grupos
''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarGru()
Set rs = Nothing
cmd.CommandText = "select * from Grupos where CodigoGru = '" +
Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Grupos set NombreGru = '" + Trim(Text2.text)
+ "', EstadoGru = '" + Trim(Combo1.text) + "' where CodigoGru = '" +
Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Grupos Modificada Satisfactoriamente...", vbExclamation, "Grupos"
Module1.auditoria ("Modifico"), ("Grupos")
Else
MsgBox "La Grupos no existe...", vbInformation, "Grupos"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Eliminar todos las Grupos
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarGru()
Set rs = Nothing
cmd.CommandText = "select * from Grupos where CodigoGru = '" +
Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar Las Grupos ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Grupos set EstadoEliGru = '0' where
CodigoGru = '" + Trim(Text1.text) + "' and EstadoEliGru = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Grupos Eliminada Satisfactoriamente...", vbExclamation, "Grupos"
Module1.auditoria ("Elimino"), ("Grupos")
End If
Else
MsgBox "La Grupos no existe...", vbInformation, "Grupos"
End If
End Sub
```

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarGru
Case 2
GuardarGru
Case 3
ConsultarGru
Case 4
ModificarGru
Case 5
EliminarGru
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

15. CODIGO FUENTE DEL MODULO DE IMPORTACION DE DATOS.

```
Dim listtablas(50) As String
Dim tablainsert As String
Private Sub ayuda001_Click()
 With awd
  .HelpFile = App.Path & "\e\ayuda.hlp"
  .HelpKey = "Importar"
  .HelpCommand = cdlHelpKey
  .ShowHelp
 End With
End Sub

Private Sub Option1_Click()
If Option1 = True Then
   frm.Visible = True
Else
```

```
   frm.Visible = False
End If
End Sub

Private Sub tablas_Click()
If tablas.text <> Empty Then
   cmd.CommandText = "select * from  aliascampos where codtab='" +
Trim(listtablas(tablas.ListIndex)) + "'"
   reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
   ejemplo.Caption = ""
   tablainsert = reg!codtab
   While Not reg.EOF
      campos.AddItem reg!nomcam + " " + "Tipo de Dato= Varchar "
      ejemplo.Caption = ejemplo.Caption + "," + "12345..."
      reg.MoveNext
   Wend
   reg.Close
End If
End Sub

Private Sub import()
Dim txtreg As String
Dim outfile As Long
Dim fname As String
If tablas.text <> Empty Then
   errores.Clear
   With siarchdia
      .DialogTitle = "Seleccione el archivo a cargar"
      .InitDir = App.Path
      .Filter = "Archivos de texto (*.txt)|*.txt|Todos los archivos(*.*)|*.*"
      .DefaultExt = "*.csv"
      .ShowOpen
      fname = .FileName


       If fname <> vbNullString Then
         Call cargar(fname)
```

```
        If errores.ListCount > 0 Then
           MsgBox "Se presentaron los siguientes errores ", vbInformation,
"Importacion de Datos"
        Else
           MsgBox "Importacion realizada con exito", vbInformation,
"Importacion de Datos"
        End If
      End If
   End With
Else
   MsgBox "Seleccione primero una Tabla de la Lista", vbInformation,
"Importacion de Datos"
End If
End Sub

Private Sub Form_Load()
'ctseguridad.Skin.ApplySkin Me.hwnd
cmd.CommandText = "select * from aliastablas"
reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
 i = 0
 While Not reg.EOF
  listtablas(i) = Trim(reg!codtab)
  tablas.AddItem Trim(reg!nomtab)
  reg.MoveNext: i = i + 1
 Wend
 reg.Close
End Sub

Private Sub tablas_KeyPress(KeyAscii As Integer)
tablas.SetFocus
End Sub

Public Sub cargar(ByVal fname As String, Optional ByVal registro As
ADODB.Recordset)
 Dim cad(1000), vec(1000), cadenaF, cadres As String
 Dim cont, lon, j, i, sw As Long
```

```
 Dim infile As Long
 Dim M As Long
 Dim k As Long
 Dim datos1() As String
 Dim txtDatos() As String
 Dim txtDatos1() As String
 Dim txtLinea, aux As String
 On Error GoTo errTrans
 infile = FreeFile()
 Open fname For Input As #infile
  Line Input #infile, txtLinea
If verificarcampos(txtLinea) = False Then
   MsgBox "Los Campos" & Chr(10) & txtLinea & Chr(10) & "No
Pertenecen a la Tabla " & tablainsert, vbInformation, "Importacion de Datos"
Else
 Line Input #infile, txtLinea
 Do While Not EOF(infile)
   Line Input #infile, txtLinea
   k = k + 1
   txtDatos = Split(txtLinea, "\n")
   For M = 0 To UBound(txtDatos)
'////////////////////////////////////////
'////////////////////////////////////////
      For j = 1 To lon '+ (cont * 2)
         cad(j) = ""
      Next j
      aux = ""
      cadres = ""
      cadenaF = ""
      cadenaF = Trim(txtDatos(M))
      lon = Len(cadenaF)
      For j = 1 To lon
         vec(j) = Mid(cadenaF, j, 1)
         If (vec(j) = " ") Then
            cont = cont + 1
         End If
      Next j
      i = 1
```

```
        For j = 1 To lon
           If (vec(j) <> " ") Then
              cad(i) = vec(j)
              i = i + 1
              sw = 0
           ElseIf (vec(j) = " " And cad(i - 1) <> """) And (vec(j + 1) <> " " And
vec(j + 1) <> "æ") Then
              cad(i) = vec(j)
              i = i + 1
              sw = 0
           End If
           If (vec(j) = "æ") And j < lon And (sw = 0) Then
              cad(i) = """
              cad(i + 1) = ","
              cad(i + 2) = """
              i = i + 3
              sw = 1
           End If
        Next j
        For j = 1 To lon '+ (cont * 2)
           If cad(j) <> "æ" Then
              cadres = cadres & cad(j)
           End If
           If cad(j) <> "æ" And cad(j) <> """ Then
              aux = aux & cad(j)
           End If
        Next j
        cadres = """ & cadres & """
'////////////////////////////////////////
'////////////////////////////////////////
'       Set reg = Nothing
'       cmd.CommandText = "desc " & tablainsert
'       'cmd.CommandText = "SHOW FIELDS FROM " & tablainsert
'       reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
'       contcam = 0
'       While Not reg.EOF
'          contcam = contcam + 1
```

```
'         reg.MoveNext
'       Wend
'       reg.Close
'
       cmd.CommandText = "select * from " & tablainsert
       reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
       sw = 0: datos1 = Split(aux, ","): sw1 = 0
       If reg.EOF Then
          sw = 1: sw1 = 1
       End If
       While Not reg.EOF And sw1 = 0
          sw = 0
         For i = 0 To contcam - 1
             If reg.Fields(i) <> datos1(i) Then
                sw = 1
             End If
          Next i
          If sw = 0 Then
             sw1 = 1
          End If
          reg.MoveNext
       Wend
       reg.Close
       sw = 1
       If sw = 1 Then
          cadena = "INSERT INTO " + tablainsert + " VALUES (" + cadres +
")"
          conn.Execute (cadena)
       Else
          IMPORTAR.errores.AddItem "Linea " & k & "-> " & cadres
          IMPORTAR.errores.AddItem "        Los datos son parecidos a los de
la tabla"
        End If
    Next M
Loop
End If
Close infile
```

```
Exit Sub
errTrans:
MsgBox txtLinea
MsgBox Err.Description
IMPORTAR.errores.AddItem "Linea " & k & "-> " & cadres
IMPORTAR.errores.AddItem "        " & Err.Description
Resume Next
End Sub

Function verificarcampos(campos) As Boolean
Dim Campo() As String
Campo = Split(campos, " ")
cmd.CommandText = "select * from aliascampos where codtab='" +
tablainsert + "'"
reg.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
i = 0: sw = 0: verificarcampos = True
While Not reg.EOF And i <= UBound(Campo) And sw = 0
   If Campo(i) <> reg!codtab & "." & reg.Fields(1) Then
      sw = 1: verificarcampos = False
   End If
   i = i + 1
   reg.MoveNext
Wend
reg.Close
End Function
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
  Case 1
   tablas.ListIndex = -1
   campos.Clear
   errores.Clear
   ejemplo.Caption = Empty
  Case 2
  'If perfiles(1) = 1 Then
'    If (Option1.Value = True) Then
       import
'    End If
```

```
 ' Else
  '  MsgBox "El Usuario " + Trim(nom) + " no tiene Perfil para Importar
Datos, Por favor consulte al ADMINISTRADOR DE SISTEMA ",
vbInformation, "ADMINISTRADOR DE SISTEMA"
  'End If
  Case 3
   Unload Me
 End Select
End Sub



'////////////////////////////////////
'////////////////////////////////////
Public Sub cargar_consulta(ByVal fname As String, Optional ByVal registro
As ADODB.Recordset)
Dim sw As Integer
Dim infile, k, j As Long
Dim txtDatos() As String, txtDatos1() As String
Dim txtLinea As String
   infile = FreeFile()
   Open fname For Input As #infile
      Line Input #infile, txtLinea
      txtDatos = Split(txtLinea, "\n")
      cadena1 = txtDatos(0)
      Line Input #infile, txtLinea
      txtDatos = Split(txtLinea, "\n")
      cadena = txtDatos(0)
   Close infile
'////////////////////////////////////
   txtDatos1 = Split(cadena1, "-")
sw = 0
On Error GoTo errTrans
   conex.Execute (cadena)
If sw = 0 Then
   reg.Open cadena, conex, adOpenDynamic, adLockBatchOptimistic
   EXPORTAR_2.resul.Cols = reg.Fields.Count + 1
   For i = 0 To EXPORTAR_2.resul.Cols - 1
```

```
      EXPORTAR_2.resul.TextMatrix(0, i) = txtDatos1(i)
   Next
   j = 1
   While Not (reg.EOF)
      For k = 1 To EXPORTAR_2.resul.Cols - 1
         EXPORTAR_2.resul.TextMatrix(j, k) = reg.Fields.Item(k - 1)
         EXPORTAR_2.resul.AutoSize (k)
      Next k
      reg.MoveNext
      EXPORTAR_2.resul.TextMatrix(j, 0) = j
      EXPORTAR_2.resul.AutoSize (0)
      j = j + 1
      EXPORTAR_2.resul.Rows = EXPORTAR_2.resul.Rows + 1
   Wend
   reg.Close
'////////////////////////////////////
   EXPORTAR_2.Show vbModal
End If
Exit Sub
errTrans:
sw = 1
'char(10) Salta la linea en msgbox
MsgBox "Error en la consulta " & Chr(10) & cadena & Chr(10) &
Err.Description, vbInformation, "Importacion de Datos"

Resume Next
End Sub

Private Sub import_con()
Dim txtreg As String
Dim outfile As Long
Dim fname As String
errores.Clear
With siarchdia
   .DialogTitle = "Seleccione el archivo a cargar"
   .InitDir = App.Path
   .Filter = "Archivos de Consulta (*.con)|*.con|Todos los Archivos(*.*)|*.*"
   .DefaultExt = "*.con"
```

```
    .ShowOpen
    fname = .FileName
    If fname <> vbNullString Then
       Call cargar_consulta(fname)
    End If
End With
End Sub
```

16. CODIGO FUENTE DEL MODULO REGISTRO DE LOGROS.

```
Private Sub cargAsig()
Set rs = Nothing
cmd.CommandText = "select NombreAsig from Asignaturas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo1.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Las Asignaturas...", vbInformation, "Logros"
End If
End Sub

Private Sub cargCursos()
Set rs = Nothing
cmd.CommandText = "select NombreCur from Cursos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Los Cursos...", vbInformation, "Logros"
End If
End Sub
```

```
Private Sub cargGru()
Set rs = Nothing
cmd.CommandText = "select NombreGru from Grupos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo3.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Los Grupos...", vbInformation, "Logros"
End If
End Sub

Private Sub Guardar()
Set rs = Nothing
cmd.CommandText = "select * from Logros where codigoLog = '" +
Trim(Text3.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "insert into Logros values ('" + Trim(Text1.text) + "','"
+ Trim(Text2.text) + "'," & _
"'" + Trim(Text3.text) + "','" + Trim(Text4.text) + "','" + Trim(Combo1.text) +
"','" + Trim(Combo2.text) + "'," & _
"'" + Trim(Combo3.text) + "')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Logros Adicionados Satisfactoriamente...", vbExclamation,
"Logros"
Module1.auditoria ("Inserto"), ("Logros")
Else
MsgBox "El Logro ya ha sido ingresado...", vbInformation, "Logros"
End If
End Sub

Private Sub Modificar()
Set rs = Nothing
```

```
cmd.CommandText = "select * from Logros where codigopen = '" +
Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Logros set Nombrepen = '" + Trim(Text2.text)
+ "'," & _
" Codigolog = '" + Trim(Text3.text) + "', NombreLog = '" + Trim(Text4.text)
+ "'," & _
" NombreAsig = '" + Trim(Combo1.text) + "',NombreCur = '" +
Trim(Combo2.text) + "'," & _
" NombreGru = '" + Trim(Combo3.text) + "' where codigopen = '" +
Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Logros Modificados Satisfactoriamente...", vbExclamation,
"Logros"
Module1.auditoria ("Modifico"), ("Logros")
Else
MsgBox "El Logro ya ha sido ingresado...", vbInformation, "Logros"
End If
End Sub

Private Sub Eliminar()
Set rs = Nothing
cmd.CommandText = "select * from Logros where codigoLog = '" +
Trim(Text3.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "delete from Logros where codigoLog = '" +
Trim(Text3.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Logros Eliminado Satisfactoriamente...", vbExclamation, "Logros"
Module1.auditoria ("Elimino"), ("Logros")
Else
MsgBox "El Logro ya ha sido ingresado...", vbInformation, "Logros"
End If
End Sub
```

```
Private Sub Command1_Click()
Consultar_Asig.Show 1
End Sub

Private Sub Form_Load()
tip = 2
cargAsig
cargCursos
```

17. CODIGO FUENTE DEL MODULO DE REGISTRO DE MATRICULA.

```
Private Sub cargCursos()
Set rs = Nothing
cmd.CommandText = "select NombreCur from Cursos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
While Not rs.EOF
Combo2.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Los Cursos...", vbInformation, "Logros"
End If
End Sub

Private Sub cargGru()
Set rs = Nothing
cmd.CommandText = "select NombreGru from Grupos"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
```

```
While Not rs.EOF
Combo3.AddItem rs.Fields(0)
rs.MoveNext
Wend
Else
MsgBox "Debe Cargar Los Grupos...", vbInformation, "Logros"
End If
End Sub

Private Sub Guardar()
Set rs = Nothing
cmd.CommandText = "select * from Matriculas where NoMatricula = '" +
Trim(Text1.text) + "' and estado = 'Activo'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "insert into Matriculas values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "'," & _
"'" + Trim(Text3.text) + "','" + Trim(Text4.text) + "','" + Trim(Text5.text) +
"','" + Trim(Text6.text) + "'," & _
"'" + Trim(Combo1.text) + "','" + Trim(Combo2.text) + "','" +
Trim(Combo3.text) + "'," & _
"'" + Trim(Combo4.text) + "','" + Trim(DTPicker1.Value) + "','" +
Trim(DTPicker2.Value) + "','" + Trim(Combo5.text) + "')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Matricula Adicionada Satisfactoriamente...", vbExclamation,
"Matricula"
Module1.auditoria ("Inserto"), ("Matriculas")
Else
MsgBox "La Matricula ya ha sido ingresado...", vbInformation, "Matricula"
End If
End Sub

Private Sub Modificar()
Set rs = Nothing
cmd.CommandText = "select * from Matriculas where NoMatricula = '" +
Trim(Text1.text) + "' and estado = 'Activo'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Matriculas set identificacion = '" +
Trim(Text2.text) + "'," & _
" nombres = '" + Trim(Text3.text) + "', Apellidos = '" + Trim(Text4.text) + "',
Direccion = '" + Trim(Text5.text) + "', Telefono = '" + Trim(Text6.text) + "',"
& _
" Sexo = '" + Trim(Combo1.text) + "', Curso = '" + Trim(Combo2.text) + "',
Grupo = '" + Trim(Combo3.text) + "'," & _
" Jornada = '" + Trim(Combo4.text) + "', FechaN = '" +
Trim(DTPicker1.Value) + "', FechaM = '" + Trim(DTPicker2.Value) + "'" & _
" where NoMatricula = '" + Trim(Text1.text) + "' and estado = 'Activo'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Matricula Modificadas Satisfactoriamente...", vbExclamation,
"Matricula"
Module1.auditoria ("Modifico"), ("Matriculas")
Else
MsgBox "La Matricula ya ha sido ingresado...", vbInformation, "Matricula"
End If
End Sub

Private Sub Eliminar()
Set rs = Nothing
cmd.CommandText = "select * from Matriculas where NoMatricula = '" +
Trim(Text1.text) + "' and estado = 'Activo'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "delete from Matriculas where NoMatricula = '" +
Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Matricula Eliminada Satisfactoriamente...", vbExclamation,
"Matricula"
Module1.auditoria ("Elimino"), ("Matriculas")
Else
MsgBox "La Matricula ya ha sido ingresado...", vbInformation, "Matricula"
End If
End Sub
```

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA

```
Private Sub Command1_Click()
Consultar_Est.Show 1
End Sub

Private Sub Form_Load()
cargCursos
cargGru
End Sub

Private Sub limpiar()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Text6.text = ""
Text1.SetFocus
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
limpiar
Case 2
Guardar
Case 3
Modificar
Case 4
Eliminar
Case 6
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

18. CODIGO FUENTE DEL MODULO REGISTRO DE PENDUM.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Limpia todos las Pensum
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarAsig()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos las Pensum
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Pensum where CodigoPen = '" +
Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into Pensum values ('" + Trim(Text1.text) + "','"
+ Trim(Text2.text) + "'," & _
"'" + Trim(Combo2.text) + "','" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Pensum Ingresada Satisfactoriamente...", vbExclamation, "Pensum"
Module1.auditoria ("Inserto"), ("Pensum")
```

```
Else
MsgBox "La Pensum ya existe...", vbInformation, "Pensum"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Consultar todos las Pensum
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Pensum where CodigoPen = '" +
Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombrePen
Combo2.text = rs!JornadaPen
Combo1.text = rs!EstadoPen
Module1.auditoria ("Consulto"), ("Pensum")
Else
MsgBox "La Pensum no existe...", vbInformation, "Pensum"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Modificar todos las Pensum
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Pensum where CodigoPen = '" +
Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
cmd.CommandText = "update Pensum set NombrePen = '" + Trim(Text2.text)
+ "', JornadaPen = '" + Trim(Combo2.text) + "'," & _
"EstadoPen = '" + Trim(Combo1.text) + "' where CodigoPen = '" +
Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
```

```
MsgBox "Pensum Modificada Satisfactoriamente...", vbExclamation,
"Pensum"
Module1.auditoria ("Modifico"), ("Pensum")
Else
MsgBox "La Pensum no existe...", vbInformation, "Pensum"
End If
End Sub


''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Eliminar todos las Pensum
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarAsig()
Set rs = Nothing
cmd.CommandText = "select * from Pensum where CodigoPen = '" +
Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar La Pensum ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update Pensum set EstadoEliPen = '0' where
CodigoPen = '" + Trim(Text1.text) + "' and EstadoEliPen = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "Pensum Eliminada Satisfactoriamente...", vbExclamation, "Pensum"
Module1.auditoria ("Elimino"), ("Pensum")
End If
Else
MsgBox "La Pensum no existe...", vbInformation, "Pensum"
End If
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarAsig
Case 2
GuardarAsig
```

```
Case 3
ConsultarAsig
Case 4
ModificarAsig
Case 5
EliminarAsig
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

19. CODIGO FUENTE DEL MODULO REGISTRO DE PROFESIONES.

```
''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Limpia todos las profesion
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub LimpiarPro()
Text1.text = ""
Text2.text = ""
Text1.SetFocus
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Adicionar todos las profesion
''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub GuardarPro()
Set rs = Nothing
cmd.CommandText = "select * from profesion where CodigoPro = '" +
Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If rs.EOF Then
Set rs = Nothing
cmd.CommandText = "Insert into profesion values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "','" & _
```

```
"'" + Trim(Combo1.text) + "','1')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "profesion Ingresada Satisfactoriamente...", vbExclamation,
"profesion"
Module1.auditoria ("Inserto"), ("Profesion")
Else
MsgBox "El profesion ya existe...", vbInformation, "profesion"
End If
End Sub



''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Consultar todos las profesion
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ConsultarPro()
Set rs = Nothing
cmd.CommandText = "select * from profesion where CodigoPro = '" +
Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombrePro
Combo1.text = rs!EstadoPro
Module1.auditoria ("Consulto"), ("Profesion")
Else
MsgBox "La profesion no existe...", vbInformation, "profesion"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''
'''''' Modificar todos las profesion
''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub ModificarPro()
Set rs = Nothing
cmd.CommandText = "select * from profesion where CodigoPro = '" +
Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
```

```
Set rs = Nothing
cmd.CommandText = "update profesion set NombrePro = '" +
Trim(Text2.text) + "', EstadoPro = '" + Trim(Combo1.text) + "' where
CodigoPro = '" + Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "profesion Modificada Satisfactoriamente...", vbExclamation,
"profesion"
Module1.auditoria ("Modifico"), ("Profesion")
Else
MsgBox "La profesion no existe...", vbInformation, "profesion"
End If
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'''' Eliminar todos las profesion
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub EliminarPro()
Set rs = Nothing
cmd.CommandText = "select * from profesion where CodigoPro = '" +
Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Set rs = Nothing
res = MsgBox("Desea Eliminar Las profesion ..?", vbYesNo, "Eliminar")
If res = vbYes Then
cmd.CommandText = "update profesion set EstadoEliPro = '0' where
CodigoPro = '" + Trim(Text1.text) + "' and EstadoEliPro = '1'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
MsgBox "profesion Eliminada Satisfactoriamente...", vbExclamation,
"profesion"
Module1.auditoria ("Elimino"), ("Profesion")
End If
Else
MsgBox "La profesion no existe...", vbInformation, "profesion"
End If
End Sub
```

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
LimpiarPro
Case 2
GuardarPro
Case 3
ConsultarPro
Case 4
ModificarPro
Case 5
EliminarPro
Case 7
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

20. CODIGO FUENTE DEL MODULO REGISTRO DE NOTAS.

```
Private Sub Command1_Click()
tip = 3
Consultar_Est_Prof.Show 1
End Sub

Private Sub Command2_Click()
tip = 4
Consultar_Est_Prof.Show 1
End Sub

Private Sub pres()
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Codigo Logro"
vs.TextMatrix(0, 2) = "Nombre del Logro"
```

```
End Sub

Private Sub Form_Load()
cargarmaterias
pres
vs.ColWidth(2) = 5500
End Sub

Private Sub cargarmaterias()
Set rs = Nothing
cmd.CommandText = "select NombreAsig from Asignaturas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF
Combo4.AddItem rs.Fields(0)
rs.MoveNext
Wend
End Sub

Private Sub cargarLogrosdeMaterias()
Set rs = Nothing
cmd.CommandText = "select CodigoLog, NombreLog from Logros where
NombreAsig = '" + Trim(Combo4.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CodigoLog
vs.TextMatrix(i, 2) = rs!NombreLog
i = i + 1
con = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "No se Han Cargado Los Logros de la Asignatura...", vbInformation,
"Registro de Notas"
```

```
End If
End Sub

Private Sub Combo4_Click()
cargarLogrosdeMaterias
End Sub


Private Sub limpiar()
Text1.text = ""
Text2.text = ""
Text3.text = ""
Text4.text = ""
Text5.text = ""
Combo1.text = ""
Combo2.text = ""
Combo3.text = ""
Combo4.text = ""
vs.Clear
pres
Text1.SetFocus

End Sub

Private Sub Guardar()
A = 0
i = 1
While A <> vs.Rows - 2
Set rs = Nothing
cmd.CommandText = "insert into boletines values ('" + Trim(Text1.text) +
"','" + Trim(Text2.text) + "'," & _
"'" + Trim(Text3.text) + "','" + Trim(Combo1.text) + "','" +
Trim(Combo2.text) + "','" + Trim(Combo3.text) + "'," & _
"'" + Trim(Text4.text) + "','" + Trim(Text5.text) + "','" + Trim(Combo4.text) +
"','" + Trim(vs.TextMatrix(i, 1)) + "'," & _
"'" + Trim(vs.TextMatrix(i, 2)) + "')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
A = A + 1
```

```
i = i + 1
Wend
MsgBox "Boletines de Logros Adicionados...", vbExclamation, "boletines de
Logros"
Module1.auditoria ("Inserto"), ("Boletines")
End Sub
Private Sub Consultar()
Set rs = Nothing
cmd.CommandText = "select * from Boletines where CodigoEst = '" +
Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
Text2.text = rs!NombreEst
Text3.text = rs!ApellidosEst
Text4.text = rs!CodigoDoc
Text5.text = rs!NombreDoc
Combo1.text = rs!Curso
Combo2.text = rs!Grupo
Combo3.text = rs!Periodo
Combo4.text = rs!NombreAsig
Module1.auditoria ("Consulto"), ("Boletines")
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CodigoLog
vs.TextMatrix(i, 2) = rs!NombreLog
i = i + 1
cont = cont + 1
vs.Rows = vs.Rows + 1
rs.MoveNext
Wend
Else
MsgBox "El Boletin del Estudiante no ha sido Ingresado...", vbInformation,
"Registro de Notas"
End If
End Sub
Private Sub Eliminar()
```

```
Set rs = Nothing
cmd.CommandText = "select * from Boletines where CodigoEst = '" +
Trim(Text1.text) + "'"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
res = MsgBox("Desea Eliminar el Logro del Estudiante...?", vbYesNo,
"Registro de Notas")
   If res = vbYes Then
   Set rs = Nothing
   cmd.CommandText = "delete from Boletines where CodigoEst = '" +
Trim(Text1.text) + "'"
   rs.Open cmd.CommandText, conn, adOpenDynamic,
adLockBatchOptimistic
   MsgBox "Boletines de Logros Eliminados...", vbExclamation, "boletines de
Logros"
   Module1.auditoria ("Elimino"), ("Boletines")
   End If
Else
MsgBox "El Boletin del Estudiante no ha sido Ingresado...", vbInformation,
"Registro de Notas"
End If
End Sub

Private Sub Reportes()
CrystalReport1.WindowState = crptMaximized
CrystalReport1.SQLQuery = "SELECT * From `boletines` boletines1 where
`boletines1`.`codigoest` = '" + Trim(Text1.text) + "'"
CrystalReport1.ReportFileName = "D:\CODIGOS VISUAL\Proyectos
U\Isabel Lopez\Reportes\asignaturas.rpt"
CrystalReport1.Action = 1
End Sub


Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
limpiar
Case 2
```

```
Guardar
Case 3
Consultar
Case 4
Eliminar
Case 5
Reportes
Case 6
res = MsgBox("Desea Salir del Formulario...", vbYesNo, "Salir")
If res = vbYes Then
Unload Me
End If
End Select
End Sub
```

21. CODIGO FUENTE DEL MODULO DE CONSULTAS AVANZADAS.

```
Public tablas As String
Dim listcampos(200), CS1 As String
Public CS As String
'Private Declare Function SendMessageLong Lib "user32" Alias
"SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal
wParam As Long, ByVal lParam As Long) As Long

'Public conexion As New ADODB.Connection
Private Sub campo_Click()
'ComboBox desplegado
'Dim Resp As Long
'Resp = SendMessageLong(Combo1.hwnd, &H14F, True, 0)
 Dim i As Integer
  For i = 0 To Campo.ListCount - 1
    If Campo.Selected(i) Then
       Exit For
    End If
  Next i
  TX = Campo
```

```
 ' Shape3.Visible = True
  TX.Visible = True
  If Mid(TX.text, 1, 1) = " " Then
    TX.text = ""
  Else
  Set reg = Nothing
  cmd.CommandText = "select codtab from aliascampos where nomcam = '" +
Trim(TX.text) + "'"
  reg.Open cmd.CommandText, conn, 2, 4
     If Not reg.EOF Then
     While Not reg.EOF
      If tabla_concatenar = reg!codtab Then
       CS1 = tabla_concatenar & "." & listcampos(i) & ""
       End If
       If tabla_concatenar1 = reg!codtab Then
       CS1 = tabla_concatenar1 & "." & listcampos(i) & ""
       End If
       reg.MoveNext
    Wend
    End If
    End If
  Set reg = Nothing
End Sub
Private Sub ejecutar()
If TX.text = "" Then
    MsgBox " Falta Seleccionar un Campo en la Lista", vbInformation,
"Consulta Avanzada...."
ElseIf Combo1.text = "" Then
    MsgBox " Falta Seleccionar Operador de Comparacion", vbInformation,
"Consulta Avanzada...."
ElseIf TX2.text = "" Then
    MsgBox " Falta Escribir el dato de " & TX.text, vbInformation, "Consulta
Avanzada...."
Else
' If Combo1.text = "Like" And Combo2.text = "Comienzen" Then
    CS = "(" & "'" & TX2 & "%" & "'" & ")"
    'Else
'   If Combo1.text = "Like" And Combo2.text = "Terminen" Then
```

```
'    CS = "(" & "'" & "%" & TX2 & "'" & ")"
'    Else
'    If Combo1.text = "Like" And Combo2.text = "Entre" Then
'      CS = "(" & "'" & "%" & TX2 & "%" & "'" & ")"
' Else
    CS = "'" + TX2 + "'"
    'End If
'    End If
 End If
 If Trim(GENERAR_REPORTES.lbwhere) <> "" Then
  GENERAR_REPORTES.lbwhere = GENERAR_REPORTES.lbwhere & "
and " & _
  CS1 & " " & Combo1 & " " & CS
  Else
  GENERAR_REPORTES.lbwhere = GENERAR_REPORTES.lbwhere & " "
& _
  CS1 & " " & Combo1 & " " & CS
 End If
 Unload Me
'End If
End Sub

Private Sub Form_Load()
carg_listablas
Combo1.Clear
Combo1.AddItem "="
Combo1.AddItem "<"
Combo1.AddItem ">"
Combo1.AddItem "<>"
Combo1.AddItem "Like"
End Sub
Private Sub carg_listablas()
Dim i As Integer
i = 0: j = 0
sw = 0
While codtabla(i) <> ""
   cmd.CommandText = "select * from aliastablas where codtab= '" +
codtabla(i) + "'"
```

```
    reg.Open cmd.CommandText, conn, 2, 4
    If Not reg.EditMode Then
       SQL_Avanzado.Campo.AddItem "    " & UCase(reg!nomtab)
       listcampos(j) = reg!codtab: j = j + 1
       If sw = 0 Then
       tabla_concatenar = reg!nomtab
       Else
       tabla_concatenar1 = reg!nomtab
       End If
    End If
    reg.Close
    cmd.CommandText = "select * from aliascampos where codtab= '" +
codtabla(i) + "'"
    reg.Open cmd.CommandText, conn, 2, 4
    While Not reg.EOF
       SQL_Avanzado.Campo.AddItem reg!nomcam
       listcampos(j) = reg!codcam: j = j + 1
       reg.MoveNext
    Wend
    reg.Close
    i = i + 1
    sw = 1
Wend
End Sub

Private Sub limpiar()
TX.text = ""
TX2.text = ""
'Shape3.Visible = False
'TX.Visible = False
'Frame2.Visible = False
Campo.SetFocus
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Index
Case 1
limpiar
```

```
Case 2
ejecutar
Case 3
Unload Me
End Select
End Sub
```

22. CODIGO FUENTE DEL MODULO DE CONSULTAR ESTUDIANTES.

```
Private Sub cargEst()
Set rs = Nothing
cmd.CommandText = "select * from Estudiantes"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CedulaEst
vs.TextMatrix(i, 2) = rs!NombresEst
vs.TextMatrix(i, 3) = rs!ApellidosEst
vs.TextMatrix(i, 4) = rs!DireccionEst
vs.TextMatrix(i, 5) = rs!TelefonoEst
vs.TextMatrix(i, 6) = rs!SexoEst
vs.TextMatrix(i, 7) = rs!FechaNEst
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
vs.AutoSize (6)
vs.AutoSize (7)
i = i + 1
cont = cont + 1
rs.MoveNext
```

```
vs.Rows = vs.Rows + 1
Wend
End Sub

Private Sub Form_Load()
cargEst
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Cedula"
vs.TextMatrix(0, 2) = "Nombre"
vs.TextMatrix(0, 3) = "Apellidos"
vs.TextMatrix(0, 4) = "Direccion"
vs.TextMatrix(0, 5) = "Telefono"
vs.TextMatrix(0, 6) = "Sexo"
vs.TextMatrix(0, 7) = "Fecha de Nacimiento"
End Sub

Private Sub vs_Click()
d = vs.Row
Matricula.Text2.text = vs.TextMatrix(d, 1)
Matricula.Text3.text = vs.TextMatrix(d, 2)
Matricula.Text4.text = vs.TextMatrix(d, 3)
Matricula.Text5.text = vs.TextMatrix(d, 4)
Matricula.Text6.text = vs.TextMatrix(d, 5)
Matricula.Combo1.text = vs.TextMatrix(d, 6)
Matricula.DTPicker1.Value = vs.TextMatrix(d, 7)
Unload Me
Matricula.Visible = True
End Sub
```

23. CODIGO FUENTE DEL MODULO CONSULTAS(Est-Prof).

```
Private Sub cargEst()
If tip = 3 Then
vs.Cols = 6
Set rs = Nothing
```

```
cmd.CommandText = "select * from Matriculas"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!identificacion
vs.TextMatrix(i, 2) = rs!Nombres
vs.TextMatrix(i, 3) = rs!Apellidos
vs.TextMatrix(i, 4) = rs!Curso
vs.TextMatrix(i, 5) = rs!Grupo
vs.AutoSize (0)
vs.AutoSize (1)
vs.AutoSize (2)
vs.AutoSize (3)
vs.AutoSize (4)
vs.AutoSize (5)
i = i + 1
cont = cont + 1
rs.MoveNext
vs.Rows = vs.Rows + 1
Wend
End If

If tip = 4 Then
vs.Cols = 4
Set rs = Nothing
cmd.CommandText = "select * from Docentes"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
i = 1
cont = 1
While Not rs.EOF
vs.TextMatrix(i, 0) = cont
vs.TextMatrix(i, 1) = rs!CedulaDoc
vs.TextMatrix(i, 2) = rs!NombresDoc
vs.TextMatrix(i, 3) = rs!ApellidosDoc
vs.AutoSize (0)
vs.AutoSize (1)
```

```
vs.AutoSize (2)
vs.AutoSize (3)
i = i + 1
cont = cont + 1
rs.MoveNext
vs.Rows = vs.Rows + 1
Wend
End If
End Sub

Private Sub Form_Load()
cargEst
If tip = 3 Then
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Cedula"
vs.TextMatrix(0, 2) = "Nombre"
vs.TextMatrix(0, 3) = "Apellidos"
vs.TextMatrix(0, 4) = "Curso"
vs.TextMatrix(0, 5) = "Grupo"
End If

If tip = 4 Then
vs.TextMatrix(0, 0) = "No"
vs.TextMatrix(0, 1) = "Cedula"
vs.TextMatrix(0, 2) = "Nombre"
vs.TextMatrix(0, 3) = "Apellidos"
End If
End Sub

Private Sub vs_Click()
If tip = 3 Then
d = vs.Row
Registro_Logros.Text1.text = vs.TextMatrix(d, 1)
Registro_Logros.Text2.text = vs.TextMatrix(d, 2)
Registro_Logros.Text3.text = vs.TextMatrix(d, 3)
Registro_Logros.Combo1.text = vs.TextMatrix(d, 4)
Registro_Logros.Combo2.text = vs.TextMatrix(d, 5)
Unload Me
```

```
Registro_Logros.Visible = True
End If

If tip = 4 Then
d = vs.Row
Registro_Logros.Text4.text = vs.TextMatrix(d, 1)
Registro_Logros.Text5.text = vs.TextMatrix(d, 2)
Unload Me
Registro_Logros.Visible = True
End If
End Sub
```

24. CODIGO FUENTE DEL MODULO MEXPORTACION.

```
Private Declare Function ShellExecute Lib "shell32.dll" Alias
"ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String, ByVal
lpFile As String, ByVal lpParameters As String, ByVal lpDirectory As String,
ByVal nShowCmd As Long) As Long
Global T
Global tabla_concatenar As String
Global tabla_concatenar1 As String
Global respuesta As String
Global reg As New ADODB.Recordset
Global regs As New ADODB.Recordset
'Global conn As New ADODB.Connection
'Global cmd As New ADODB.Command
Global nombre1 As String
Global tab_1 As Boolean
Global tab_2 As Boolean
Global nombre2 As String
Public codtabla(50)


'Procedimiento para exportar datos a un archivo de texto
Public Sub exprepor(ByVal fname As String)
 Dim v1() As Integer
 Dim cadena As String
```

```
 sFic = fname
 nfic = FreeFile
 'ESTE ES EL MAN
 Open sFic For Output As nfic
 Print #nfic, campostablas
 'aqui redimencionamos un V1 para guardar
 'la mayor longitud por columnas
 ReDim v1(1 To (EXPORTAR_2.resul.Cols - 1))
 For i = 0 To EXPORTAR_2.resul.Rows - 1
  EXPORTAR_2.resul.Row = i
  For n = 1 To EXPORTAR_2.resul.Cols - 1
   EXPORTAR_2.resul.Col = n
   lonmn = Len(EXPORTAR_2.resul.Cell(flexcpText,
EXPORTAR_2.resul.Row, EXPORTAR_2.resul.Col,
EXPORTAR_2.resul.Row, EXPORTAR_2.resul.Col))
   If v1(n) < lonmn Then
    v1(n) = lonmn
   End If
  Next n
 Next i
 'utilazando el valor guardado en el V1 escribimos la
 'informacion a SICM_EXPORTAR de tal manera que las columnas queden
alineadas
 For i = 0 To EXPORTAR_2.resul.Rows - 1
  EXPORTAR_2.resul.Row = i
  cadena = ""
  For n = 1 To EXPORTAR_2.resul.Cols - 1
   EXPORTAR_2.resul.Col = n
   lonmn = Len(EXPORTAR_2.resul.Cell(flexcpText,
EXPORTAR_2.resul.Row, EXPORTAR_2.resul.Col,
EXPORTAR_2.resul.Row, EXPORTAR_2.resul.Col))
   If lonmn < v1(n) Then
    s = v1(n) - lonmn
    Else
     s = 0
   End If
   cadena1 = Space(1 + s)
   If cadena = "" Then
```

```
   cadena = EXPORTAR_2.resul.text & cadena1 & "æ"
   Else
    cadena = cadena & " " & EXPORTAR_2.resul.text & cadena1 & "æ"
  End If
 Next
 Print #nfic, cadena
Next
'aqui abrimos automaticamente el archivo de texto donde se
'guardaron los datos
T = ShellExecute(EXPORTAR_2.hwnd, "OPEN", fname, "", "", 1)
Close
campostablas = ""
End Sub

'Procedimiento para SICM_EXPORTAR datos a Excel
Public Sub exprepor1()
 Dim camp() As String
 Dim i As Long
 Dim n As Long
 On Error Resume Next
 Set objExcel = GetObject(, "Excel.Application")
 If Err.Number Then
  Err.Clear
  Set objExcel = CreateObject("Excel.Application")
  If Err.Number Then
   MsgBox "No se puede abrir Excel"
  End If
 End If
 objExcel.Visible = True
 Set objWorkbook = objExcel.Workbooks.Add
 AppActivate "FlexGrid To Excel Demo"
 camp = Split(campostablas, " ")
 For i = 0 To EXPORTAR_2.resul.Cols - 2
   objWorkbook.ActiveSheet.Cells(1, i + 1).Value = camp(i)
 Next
 For i = 1 To EXPORTAR_2.resul.Rows
  EXPORTAR_2.resul.Row = i - 1
  For n = 1 To EXPORTAR_2.resul.Cols - 1
```

```
  EXPORTAR_2.resul.Col = n
  objWorkbook.ActiveSheet.Cells(i + 1, n).Value = EXPORTAR_2.resul.text
 Next
Next
End Sub

Public Sub EXPORTAR(ByVal txtarchivo As String, ByVal registro As
ADODB.Recordset)
 On Error Resume Next
 Dim outfile As Long, i As Long
 Dim txtfilename As String
 outfile = FreeFile()
 Open txtarchivo For Output As #outfile
 If Not registro.EOF Then
  registro.MoveFirst
  Do While Not registro.EOF
   DoEvents
   txtarchivo = registro.Fields(0).Value
   For i = 1 To registro.Fields.Count - 1
    txtarchivo = txtarchivo & " " & registro.Fields(i).Value
   Next i
   registro.MoveNext
   Print #outfile, txtarchivo
   txtarchivo = ""
  Loop
  MsgBox "El proceso de exportación de datos fue satisfactorio",
vbInformation, "Exportación de  Datos"
 End If
 Close outfile
End Sub
```

25. CODIGO FUENTE DEL MODULO MODULE 1.

```
DefInt A-Z
Option Explicit

Global var As Integer
Global usu As String
Global fecha As Date
```

```
Global rs As New ADODB.Recordset
Global cmd As New ADODB.Command
Global conn As New ADODB.Connection
Global conexion As String
Global tip As Integer
Global hora As Variant
Public Const ENCRYPT = 1, DECRYPT = 2

Sub conex()
conexion = "Provider=MSDASQL.1;Password=123;Persist Security
Info=True;User ID=root;Data Source=Isabel_lopez"
conn.Open conexion
End Sub
Public Function auditoria(desc As String, tabla As String)
Set rs = Nothing
fecha = Date
hora = time
cmd.CommandText = "insert into auditoria
(descripcion,tabla,usuario,fecha,hora) values ('" + Trim(desc) + "','" +
Trim(tabla) + "','" + Trim(usu) + "','" + Trim(Format(fecha,
"DD/MM/YYYY")) + "','" + Trim(Format(hora, "HH:MM:SS")) + "')"
rs.Open cmd.CommandText, conn, adOpenDynamic, adLockBatchOptimistic
End Function
Public Function Encryptacion( _
userkey As String, text As String, Action As Single _
) As String
Dim UserKeyx As String
Dim temp As Integer
Dim time As Integer
Dim i, j, n As Integer
Dim rtn As String

n = Len(userkey)
ReDim UserKeyASCIIS(1 To n)
For i = 1 To n
UserKeyASCIIS(i) = Asc(Mid$(userkey, i, 1))
Next
```

```
ReDim textASCIIS(Len(text)) As Integer
For i = 1 To Len(text)
textASCIIS(i) = Asc(Mid$(text, i, 1))
Next
If Action = ENCRYPT Then
For i = 1 To Len(text)
temp = textASCIIS(i) + UserKeyASCIIS(i)
If temp > 255 Then
temp = temp - 255
End If
rtn = rtn + Chr$(temp)
Next
ElseIf Action = DECRYPT Then
For i = 1 To Len(text)
temp = textASCIIS(i) - UserKeyASCIIS(i)
If temp < 0 Then
temp = temp + 255
End If
rtn = rtn + Chr$(temp)
Next
End If
Encryptacion = rtn
End Function
```

BIBLIOGRAFIA

Tamayo y Tamayo, Mario. Metodología formal de la investigación Científica. Editorial Limusa S.A. Segunda Edición.

Tamayo y Tamayo, Mario. Proceso de la investigación Científica. Editorial Limusa S.A. México 2000.

Senn J.A, Mario. Análisis y diseño de sistemas de información, Editorial Mc Graw Hill. México 1998.

La nueva educación. Editorial Prolibros.

Microsoft Enciclopedia Encarta 2002.

Internet **www.google.com,** Sistemas de Información.

MANUAL DEL SISTEMA

DISEÑO E IMPLEMENTACION DE UN SISTEMA DE INFORMACION ACADEMICO-ADMINISTRATIVO PARA LA ESCUELA ANA BOLIVAR DE CONSUEGRA Y EL COLEGIO JOSE CONSUEGRA HIGGINS EN EL CORREIMIENTO ISABAEL LOPEZ (ATLANTICO) .

JUAN CARLOS CASAS JACOME.

RICARDO CORTES.

OSVALDO VILLALBA.

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO SIMÓN BOLÍVAR
DECIMO SEMESTRE
INGENIERIA DE SISTEMAS
BARANQUILLA
2005

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

TABLA DE CONTENIDO

INTRODUCCION

1. OBJETIVOS

1.1 OBJETIVO GENERAL

1.2 OBJETIVOS ESPECÍFICOS

2. INSTALACIÓN

2.1 HARDWARE

2.2 SOFTWARE

2.3 INICIAR LA INSTALACION

3. DEFINICIÓN DE LA BASE DE DATOS

3.1 TABLAS DEL SISTEMA

3.2 SCRIPT PARA LA CREACIÓN DE LA BASE DE DATOS

4. DESCRIPCION DE LOS FORMULARIOS

5. DIAGRAMA ENTIDAD RELACION

6. DIAGRAMA DE FLUJO DE DATOS

7. DICCIONARIO DE DATOS

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

INTRODUCCION

Este manual del sistema ha sido elaborado con el fin de facilitar especificaciones precisas de cada uno de los procedimientos que contiene esta aplicación.

Esta aplicación se hizo con el propósito de llevar el manejo de las matriculas y el registro de notas en la escuela y en el colegio al implementar un software que maneje el registro de las inscripciones y el reporte de los boletines con el propósito de facilitar el trabajo a la secretaria de la escuela y del colegio.

El documento sirve de guía y soporte para las personas encargadas de realizar mantenimiento al software, en él se explican detalladamente las funciones y los procedimientos, las variables que se utilizaron, las tablas que interactúan y las funciones que realiza cada formulario.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Elaborar un manual del sistema con el fin de proporcionar explicaciones concisas del diseño, el desarrollo del sistemas de información y cada uno de los procedimientos que se realizan en el software y también sirva de guía y soporte a las personas encargada del mantenimiento del sistema.

1.2 OBJETIVOS ESPECIFICOS

- Informar sobre las funciones que cada uno de los formularios .
- Facilitar el mantenimiento del sistema de información.
- Realizar la planificación y análisis del proyecto.
- Desarrollar códigos y presentación del sistema de información.
- Describir el diseño de la base de dato.
- Notificar sobre los estándares de las tablas.
- Implementar pruebas del sistema instalado.

2. INSTALACION

2.1 HARDWARE

Computador con las siguientes características

- Memória Ram de 128 MB, 256 MB o superior
- Unidad CD
- Pentium II, III o superior
- 800GHz o superior

2.2 SOFTWARE

- Mysql
- Visual Basic 6.0
- Windows 95/98/XP
- Crystal report 7.0

3. DEFINICIÓN DE LA BASE DE DATOS

3.1 TABLAS DEL SISTEMA

ACUDIENTES

Esta tabla contiene los campos CedulaEst, codigoacu, cedulaacu, expedidaacu, nombresacu, apellidosacu, edadacu, direccionacu, telefonoacu, sexoacu, fechanacu, estadoacu, en esta tabla se registran los datos del acudiente.

AREAS

Esta tabla contiene los campos codigoare, nombreare, estadoare, en esta tabla se registran los datos de las áreas..

ASIGNATURAS

Esta tabla contiene los campos codigoasig, nombreasig, estadoasig, en esta tabla se registran los datos de las asignaturas.

ASIGNATURAS_PENSUM

Esta tabla contiene los campos codigopen, nombrepen, codigoasig, nombreasig, en esta tabla se registran los datos de las relacion asignaturas_pensum

AUDITORIA

Esta tabla contiene los campos Descripción, tabla, usuario, fecha, hora, en esta tabla se registran los datos de las transacciones que se han hecho en el sistema

BOLETINES

Esta tabla contiene los campos codigoest nombreest, apellidosest, curso, grupo, periodo, codigodoc, nombreasig, codigolog, nombredoc, nombrelog, en esta tabla se registran los datos de el boletín

CURSOS

Esta tabla contiene los campos codigocur, nombrecur, estadocur, en esta tabla se registran los datos de los cursos

DOCENTES

Esta tabla contiene los campos ceduladoc, nombresdoc, apellidosdoc, direcciondoc, telefonodoc, expedidadoc, edaddoc, sexodoc, fechandoc, asignaturadoc, cursodoc, gradodoc, estadodoc, en esta tabla se registran los datos de los docentes.

ESTUDIANTES

Esta tabla contiene los campos codigoest, cedulaest, expedidaest, nombresest, apellidosest, edadest, direccionest, telefonoest, sexoest, fechanest, estadoest, en esta tabla se registran los datos de los estudiantes.

GRUPOS

Esta tabla contiene los campos codigogru, nombregru, estadogru, en esta tabla se registran los datos de los grupos

LOGROS

Esta tabla contiene los campos codigopen, nombrepen, codigolog, nombrelog, nombreasig, nombrecur, nombregru, en esta tabla se registran los datos de los logros.

MATRICULAS

Esta tabla contiene los campos NoMatricula, identificacion, nombres, apellidos, direccion, telefono, sexo, curso, grupo, jornada, fechan, fecham, estado, en esta tabla se registran los datos de las matriculas.

PENSUM

Esta tabla contiene los campos **codigopen, nombrepen, jornadapen, estadopen,** en esta tabla se registran los datos de los pensum.

UNIVERSIDAD SIMON BOLIVAR
BIBLIOTECA
BARRANQUILLA

PROFESION

Esta tabla contiene los campos codigopro, nombrepro, estadopro, en esta tabla se registran los datos de las profesiones de los docentes.

USUARIO

Esta tabla contiene los campos cedula, nombre, apellidos, edad, direccion, telefono, login, pasword, tipo_usuario, estado, en esta tabla se registran los datos de los usuarios del sistema.

3.2 SCRIPT PARA LA CREACIÓN DE LA BASE DE DATOS

```
CREATE TABLE `acudientes` (
  `CedulaEst` varchar(60) default NULL,
  `codigoacu` varchar(60) default NULL,
  `cedulaacu` varchar(60) default NULL,
  `expedidaacu` varchar(60) default NULL,
  `nombresacu` varchar(60) default NULL,
  `apellidosacu` varchar(60) default NULL,
  `edadacu` varchar(60) default NULL,
  `direccionacu` varchar(60) default NULL,
  `telefonoacu` varchar(60) default NULL,
  `sexoacu` varchar(60) default NULL,
  `fechanacu` varchar(60) default NULL,
  `estadoacu` varchar(5) default NULL
) TYPE=MyISAM;
```

```
CREATE TABLE `aliascampos` (
  `codtab` varchar(60) default NULL,
  `codcam` varchar(60) default NULL,
  `nomcam` varchar(60) default NULL
) TYPE=MyISAM;

  CREATE TABLE `aliastablas` (
  `codtab` varchar(60) default NULL,
  `nomtab` varchar(60) default NULL
) TYPE=MyISAM;

CREATE TABLE `areas` (
  `codigoare` varchar(60) default NULL,
  `nombreare` varchar(60) default NULL,
  `estadoare` varchar(10) default NULL,
  `EstadoEliAre` varchar(5) default NULL
) TYPE=MyISAM;
```

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

```
CREATE TABLE `asignaturas` (
 `codigoasig` varchar(60) default NULL,
 `nombreasig` varchar(60) default NULL,
 `estadoasig` varchar(60) default NULL,
 `estadoeliasig` varchar(60) default NULL
) TYPE=MyISAM;

CREATE TABLE `asignaturas_pensum` (
 `codigopen` varchar(60) default NULL,
 `nombrepen` varchar(60) default NULL,
 `codigoasig` varchar(60) default NULL,
 `nombreasig` varchar(60) default NULL
) TYPE=MyISAM;

CREATE TABLE `auditoria` (
 `descripcion` varchar(60) default NULL,
 `tabla` varchar(60) default NULL,
 `usuario` varchar(60) default NULL,
 `fecha` varchar(60) default NULL,
 `hora` varchar(60) default NULL
) TYPE=MyISAM;
```

```
CREATE TABLE `boletines` (
`codigoest` varchar(60) default NULL,
`nombreest` varchar(60) default NULL,
`apellidosest` varchar(60) default NULL,
`curso` varchar(60) default NULL,
`grupo` varchar(60) default NULL,
`periodo` varchar(60) default NULL,
`codigodoc` varchar(60) default NULL,
`nombredoc` varchar(60) default NULL,
`nombreasig` varchar(60) default NULL,
`codigolog` varchar(60) default NULL,
`nombrelog` varchar(60) default NULL
) TYPE=MyISAM;

CREATE TABLE `cursos` (
`codigocur` varchar(60) default NULL,
`nombrecur` varchar(60) default NULL,
`estadocur` varchar(10) default NULL,
`EstadoEliCur` varchar(5) default NULL
) TYPE=MyISAM;
```

```
CREATE TABLE `docentes` (
`ceduladoc` varchar(60) default NULL,
`nombresdoc` varchar(60) default NULL,
`apellidosdoc` varchar(60) default NULL,
`direcciondoc` varchar(60) default NULL,
`telefonodoc` varchar(60) default NULL,
`expedidadoc` varchar(60) default NULL,
`edaddoc` varchar(60) default NULL,
`sexodoc` varchar(60) default NULL,
`fechandoc` varchar(60) default NULL,
`asignaturadoc` varchar(60) default NULL,
`cursodoc` varchar(60) default NULL,
`gradodoc` varchar(60) default NULL,
`estadodoc` varchar(5) default NULL
) TYPE=MyISAM;

CREATE TABLE `estudiantes` (
`codigoest` varchar(60) default NULL,
`cedulaest` varchar(60) default NULL,
`expedidaest` varchar(60) default NULL,
`nombresest` varchar(60) default NULL,
`apellidosest` varchar(60) default NULL,
```

```
  `edadest` varchar(60) default NULL,
  `direccionest` varchar(60) default NULL,
  `telefonoest` varchar(60) default NULL,
  `sexoest` varchar(60) default NULL,
  `fechanest` varchar(60) default NULL,
  `estadoest` varchar(5) default NULL
) TYPE=MyISAM;

 CREATE TABLE `grupos` (
  `codigogru` varchar(60) default NULL,
  `nombregru` varchar(60) default NULL,
  `estadogru` varchar(10) default NULL,
  `EstadoEliGru` varchar(5) default NULL
) TYPE=MyISAM

 CREATE TABLE `logros` (
  `codigopen` varchar(60) default NULL,
  `nombrepen` varchar(60) default NULL,
  `codigolog` varchar(60) default NULL,
  `nombrelog` varchar(60) default NULL,
  `nombreasig` varchar(60) default NULL,
  `nombrecur` varchar(60) default NULL,
```

```
  `nombregru` varchar(60) default NULL
) TYPE=MyISAM;

  CREATE TABLE `matriculas` (
  `NoMatricula` varchar(60) default NULL,
  `identificacion` varchar(60) default NULL,
  `nombres` varchar(60) default NULL,
  `apellidos` varchar(60) default NULL,
  `direccion` varchar(60) default NULL,
  `telefono` varchar(60) default NULL,
  `sexo` varchar(60) default NULL,
  `curso` varchar(60) default NULL,
  `grupo` varchar(60) default NULL,
  `jornada` varchar(60) default NULL,
  `fechan` varchar(60) default NULL,
  `fecham` varchar(60) default NULL,
  `estado` varchar(10) default NULL
) TYPE=MyISAM;

CREATE TABLE `pensum` (
  `codigopen` varchar(60) default NULL,
  `nombrepen` varchar(60) default NULL,
```

```
  `jornadapen` varchar(60) default NULL,
  `estadopen` varchar(60) default NULL,
  `EstadoEliPen` varchar(5) default NULL
) TYPE=MyISAM;

CREATE TABLE `profesion` (
  `codigopro` varchar(60) default NULL,
  `nombrepro` varchar(60) default NULL,
  `estadopro` varchar(60) default NULL,
  `EstadoEliPro` varchar(5) default NULL
) TYPE=MyISAM;

CREATE TABLE `referencia` (
  `nomtab1` varchar(60) default NULL,
  `nomtab2` varchar(60) default NULL,
  `campo1` varchar(60) default NULL,
  `campo2` varchar(60) default NULL
) TYPE=MyISAM;

CREATE TABLE `usuarios` (
  `cedula` varchar(60) default NULL,
  `nombre` varchar(60) default NULL,
```

```
  `apellidos` varchar(60) default NULL,
  `edad` varchar(60) default NULL,
  `direccion` varchar(60) default NULL,
  `telefono` varchar(60) default NULL,
  `login` varchar(60) default NULL,
  `pasword` varchar(60) default NULL,
  `tipo_usuario` varchar(60) default NULL,
  `estado` varchar(5) default NULL
) TYPE=MyISAM;
```

4. DESCRIPCION DEL ESTÁNDAR DE LOS FORMULARIOS

Los nombres de los formularios hacen referencia a la función que cada uno realiza en el sistema de información de una manera sencilla para el mejor entendimiento de la persona encargada de hacer el mantenimiento del software.

4.1 Descripción De Los Formularios

REGIRTRO DE AREAS

Private Sub LimpiarAre() Limpia todos las Areas

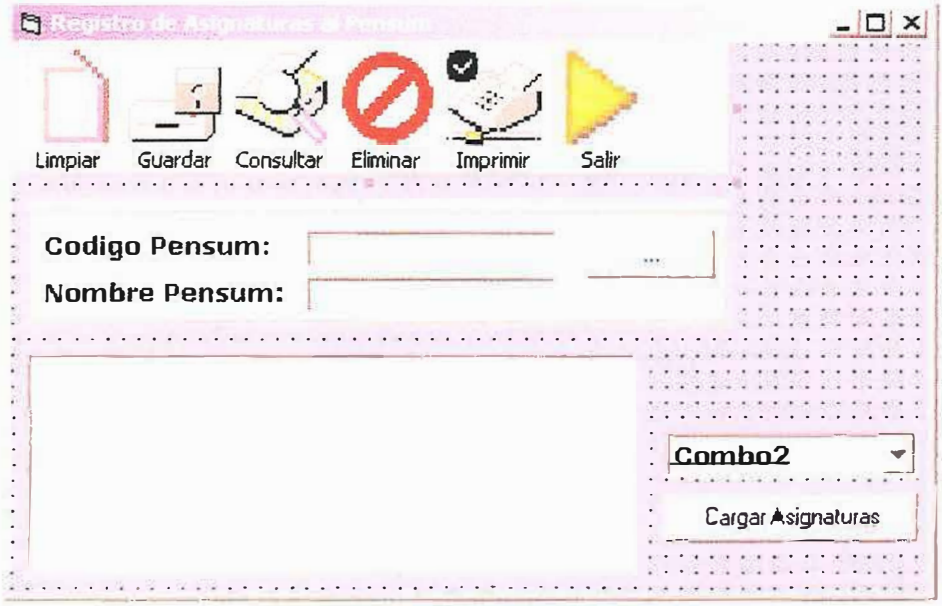
Private Sub GuardarAre() Adicionar todos las Area

Private Sub ConsultarAre() Consultar todos las Areas

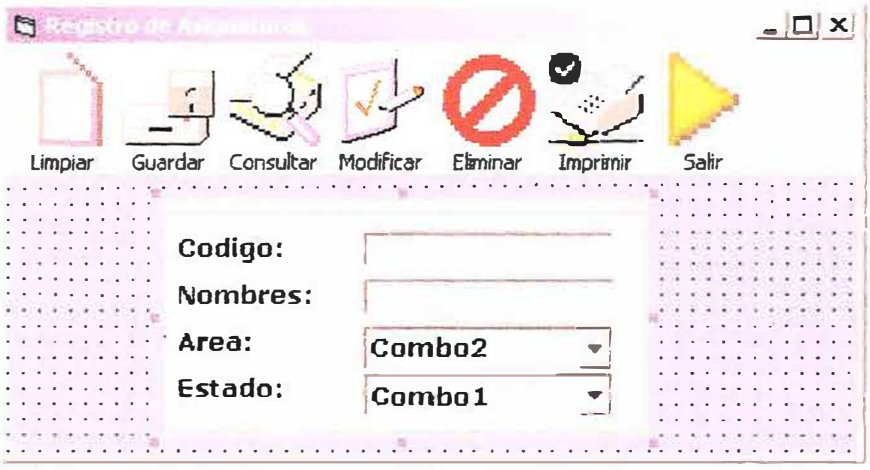
Private Sub ModificarAre() Modificar todos las Areas

Private Sub Eliminarare() Eliminar todos las Areas

REGISTROS DE ASIGNA TURAS AL PENSUM



REGISTRO DE ASIGNATURAS



Private Sub LimpiarAsig() Limpia todos las Asignaturas

Private Sub GuardarAsig() Adicionar todos las Asignaturas

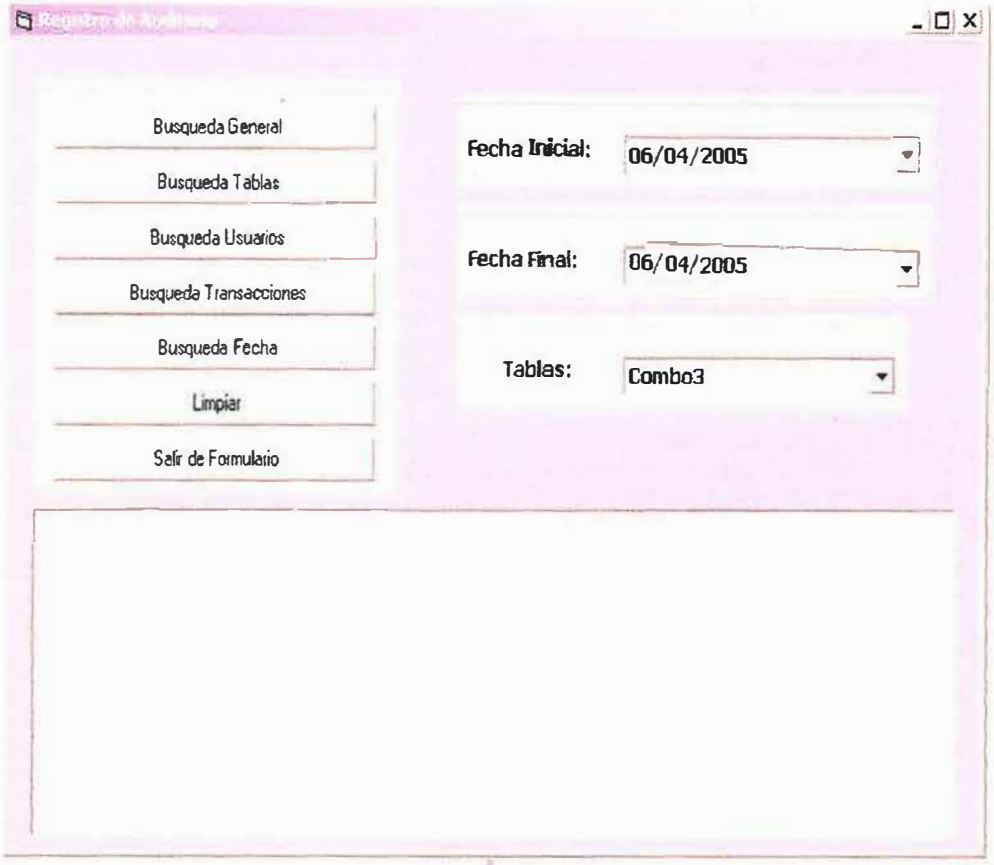
Private Sub ConsultarAsig() Consultar todos las Asignaturas

Private Sub ModificarAsig() Modificar todos las Asignaturas

Private Sub EliminarAsig( ) Eliminar todos las Asignaturas

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

REGISTRO DE AUDITORIA



CREACION DE USUARIOS

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

REGISTRO DE CURSOS

Private Sub LimpiarCur() Limpia todos las Cursos

Private Sub GuardarCur() Adicionar todos las cursos

Private Sub ConsultarCur() Consultar todos las Cursos

Private Sub ModificarCur() Modificar todos las Cursos

Private Sub EliminarCur() Eliminar todos las Cursos

REGISTRO DE DOCENTE

Private Sub LimpiarDoc() Limpia todos los Docentes

Private Sub GuardarDoc() Adicionar todos los Docentes

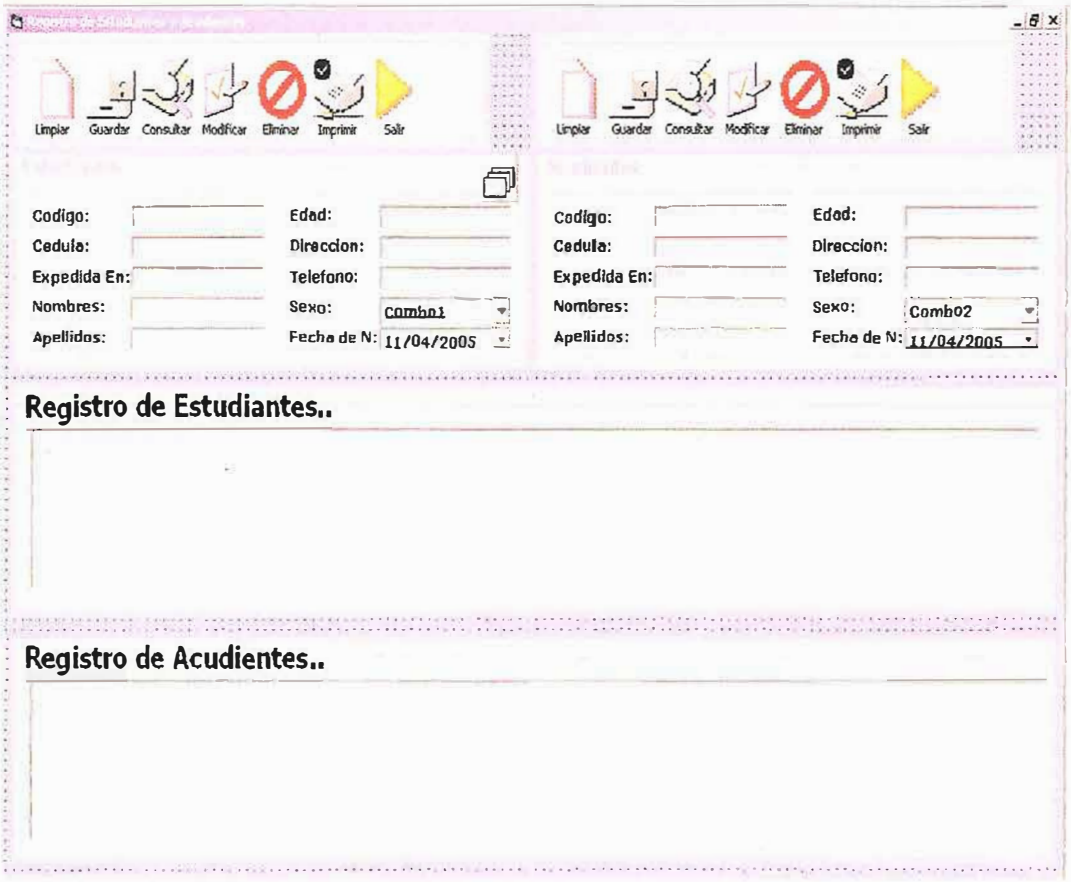
Private Sub ConsultarDoc() Consultar todos los Docentes

Private Sub ModificarDoc() Modificar todos los Docentes

Private Sub EliminarDoc() Eliminar todos los Docentes

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

REGISTRO DE ESTUDIANTES Y ACUDIENTES



Private Sub LimpiarEst() limpiar todos los estudiantes

Private Sub LimpiarAcu() limpiar todos los Acudientes

Private Sub GuardarEst() Adicionar todos los estudiantes

Private Sub GuardarAcu() Adicionar todos los acudientes

Private Sub ConsultarEst() Consultar todos los estudiantes

Private Sub ConsultarAcu() Consultar todos los Acudientes

Private Sub ModificarEst() Modificar todos los Estudiantes

Private Sub ModificarAcu() Modificar todos los Acudientes

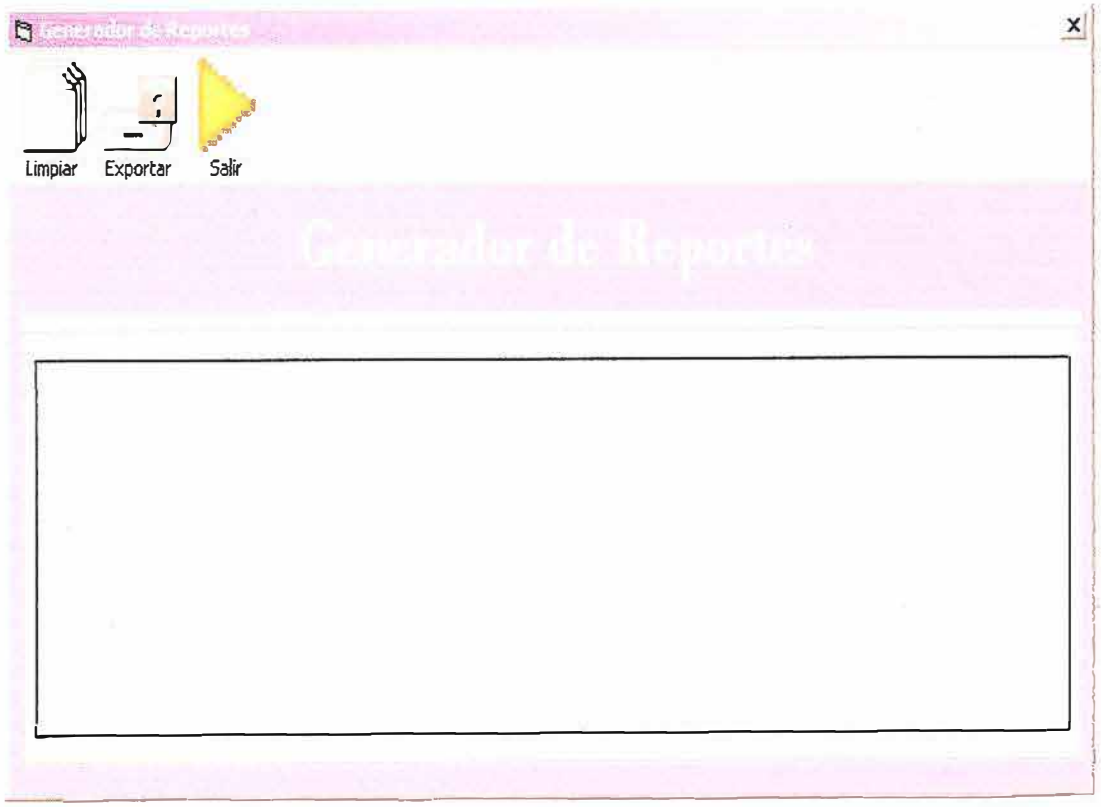
Private Sub EliminarEst() Eliminar todos los Estudiantes

Private Sub EliminarAcu() Eliminar todos los Acudientes

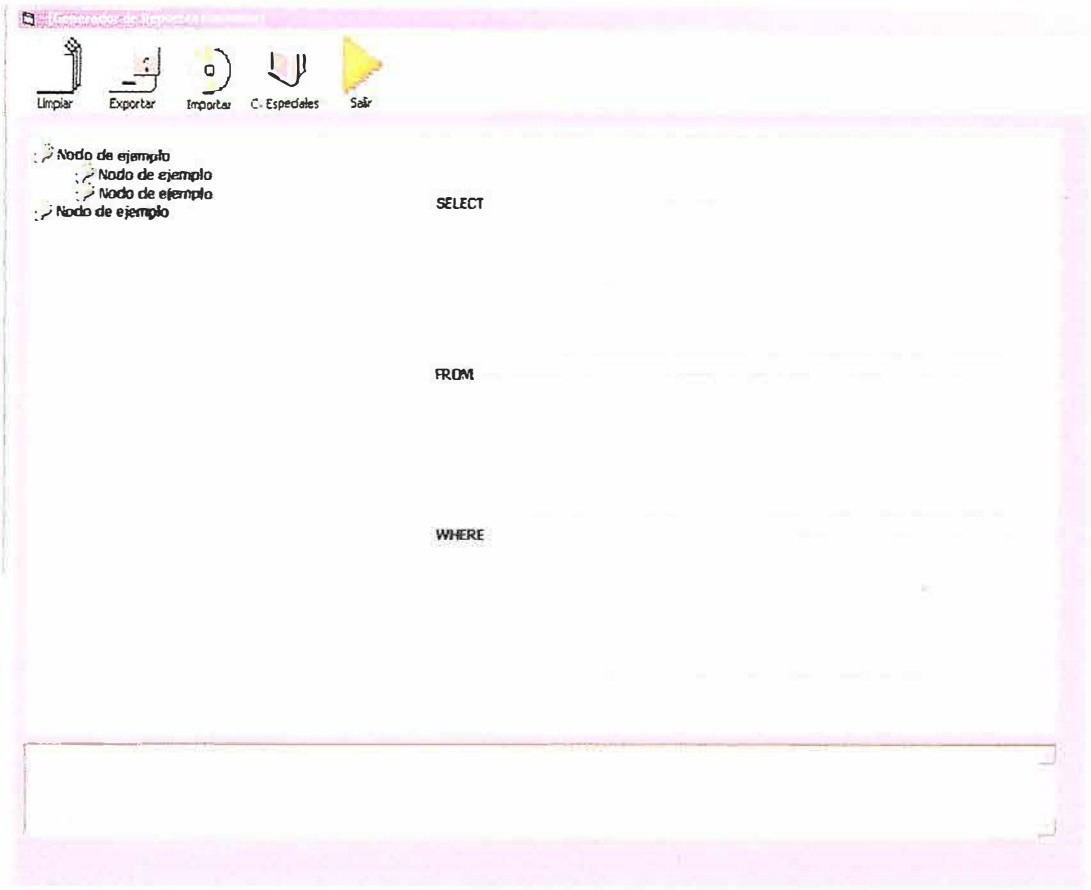
Private Sub ConGrillaEst() Consultar en la Grilla todos los Estudiantes

Private Sub ConGrillaAcu() Consultar en la Grilla todos los Acudientes

GENERADOR DE REPORTES



GENERADOR DE REPORTES DINAMICOS



REGISTRO DE GRUPO

Private Sub LimpiarGru() Limpia todos las Grupos

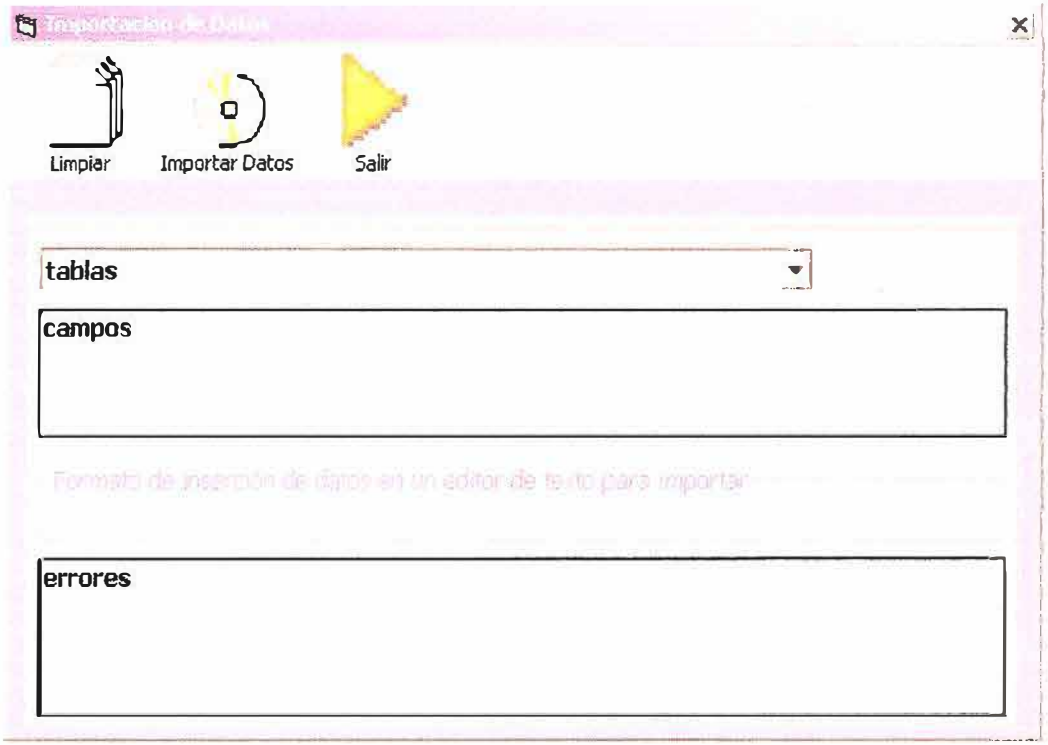
Private Sub GuardarGru() Adicionar todos las Grupos

Private Sub ConsultarGru() Consultar todos las Grupos

Private Sub ModificarGru() Modificar todos las Grupos

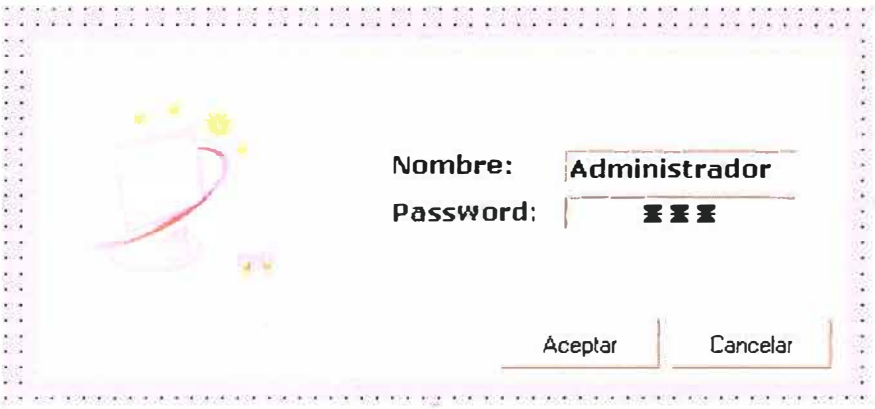
Private Sub EliminarGru() Eliminar todos las Grupos

IMPORTACION DE DATOS



UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

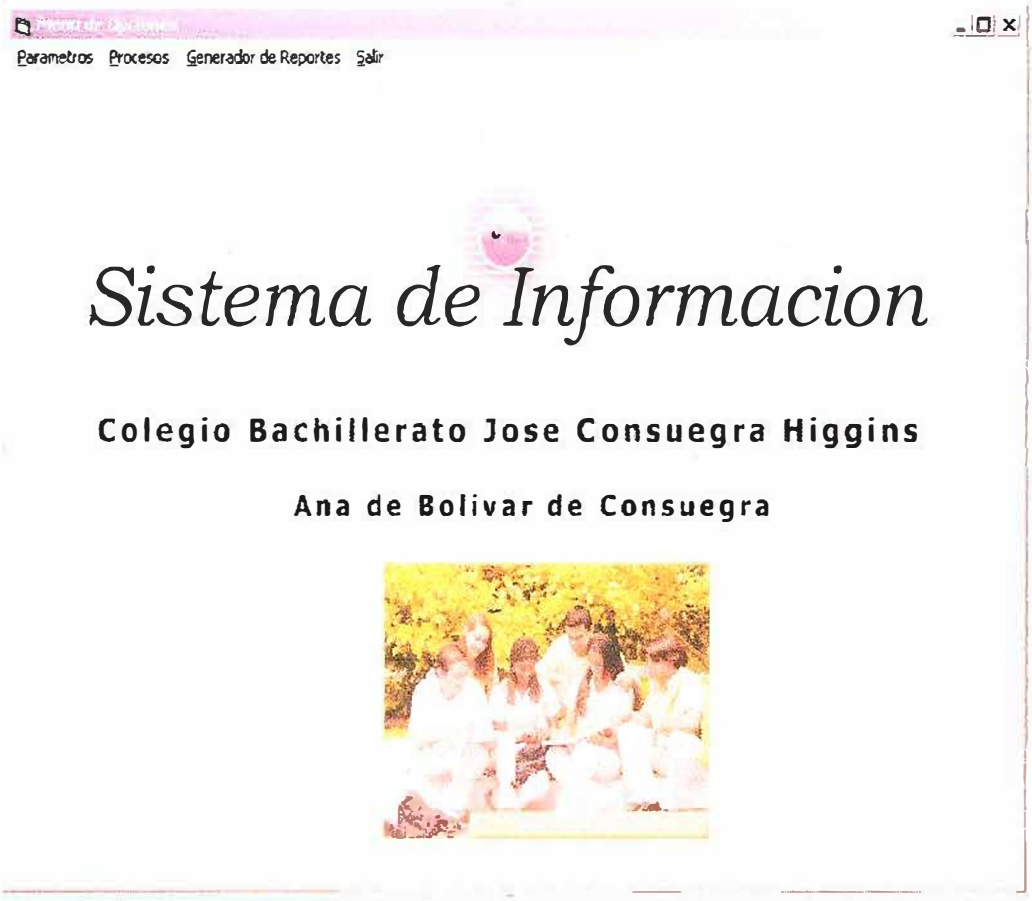
INGRESO



REGISTRO DE LOGROS

REGISTRO DEMATRICULAS

MENU PRINCIPAL



REGISTRO DE PENSUM

Private Sub LimpiarAsig() Limpia todos las Pensum

Private Sub GuardarAsig() Adicionar todos las Pensum

Private Sub ConsultarAsig() Consultar todos las Pensum

Private Sub ModificarAsig() Modificar todos las Pensum

Private Sub EliminarAsig() Eliminar todos las Pensum

REGISTRO DE PROFESION

Private Sub LimpiarPro() Limpia todos las profesion

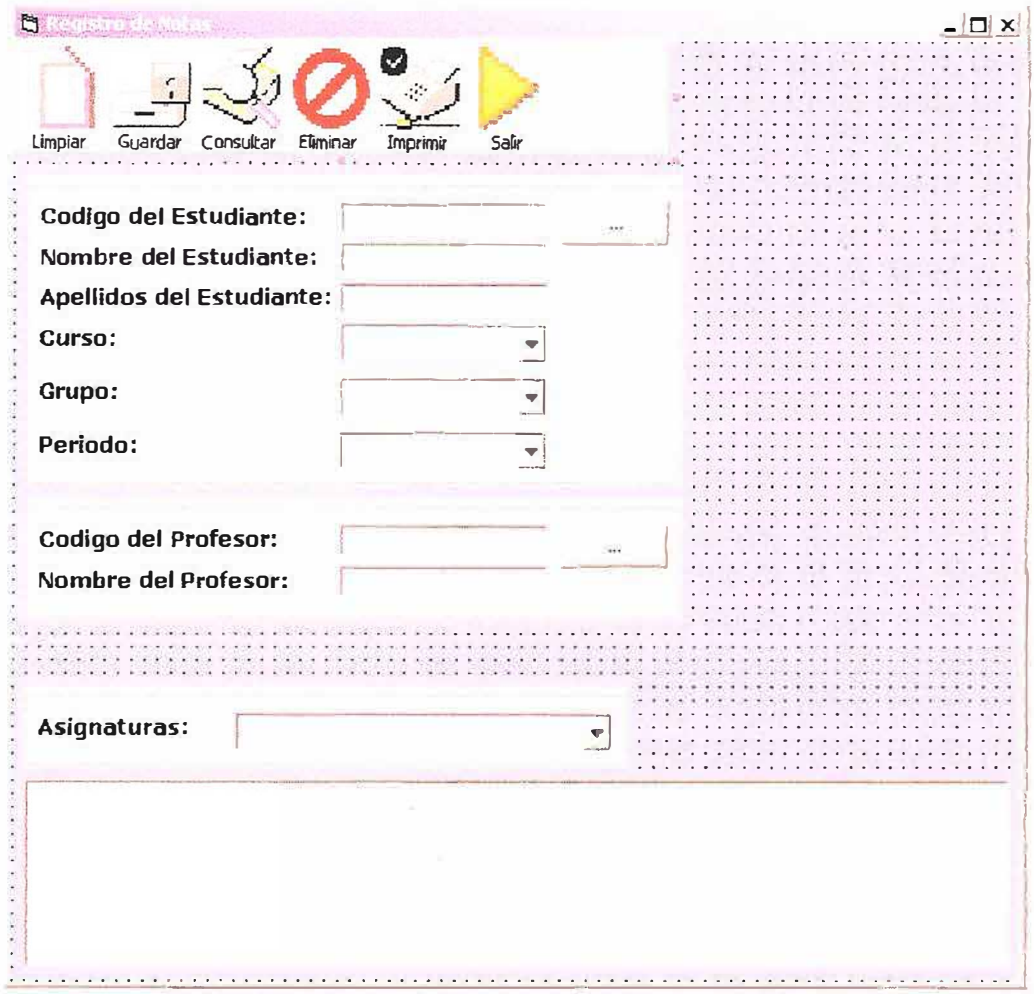
Private Sub GuardarPro() Adicionar todos las profesion

Private Sub ConsultarPro() Consultar todos las profesion

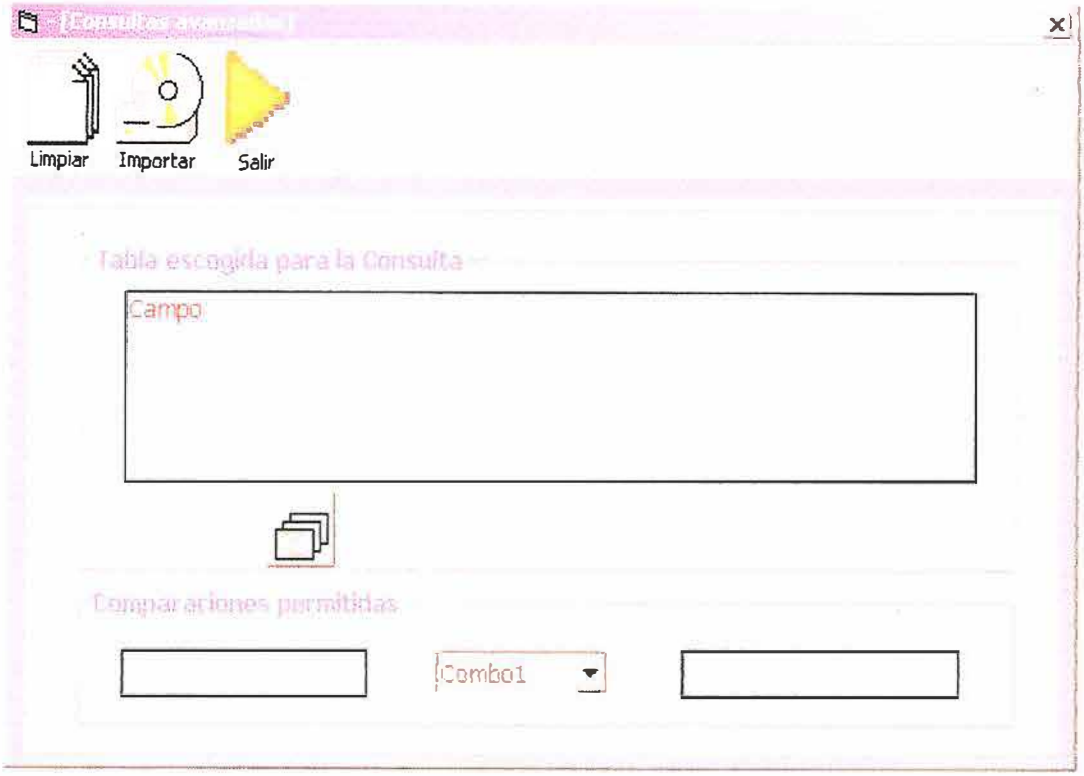
Private Sub ModificarPro() Modificar todos las profesion

Private Sub EliminarPro() Eliminar todos las profesion

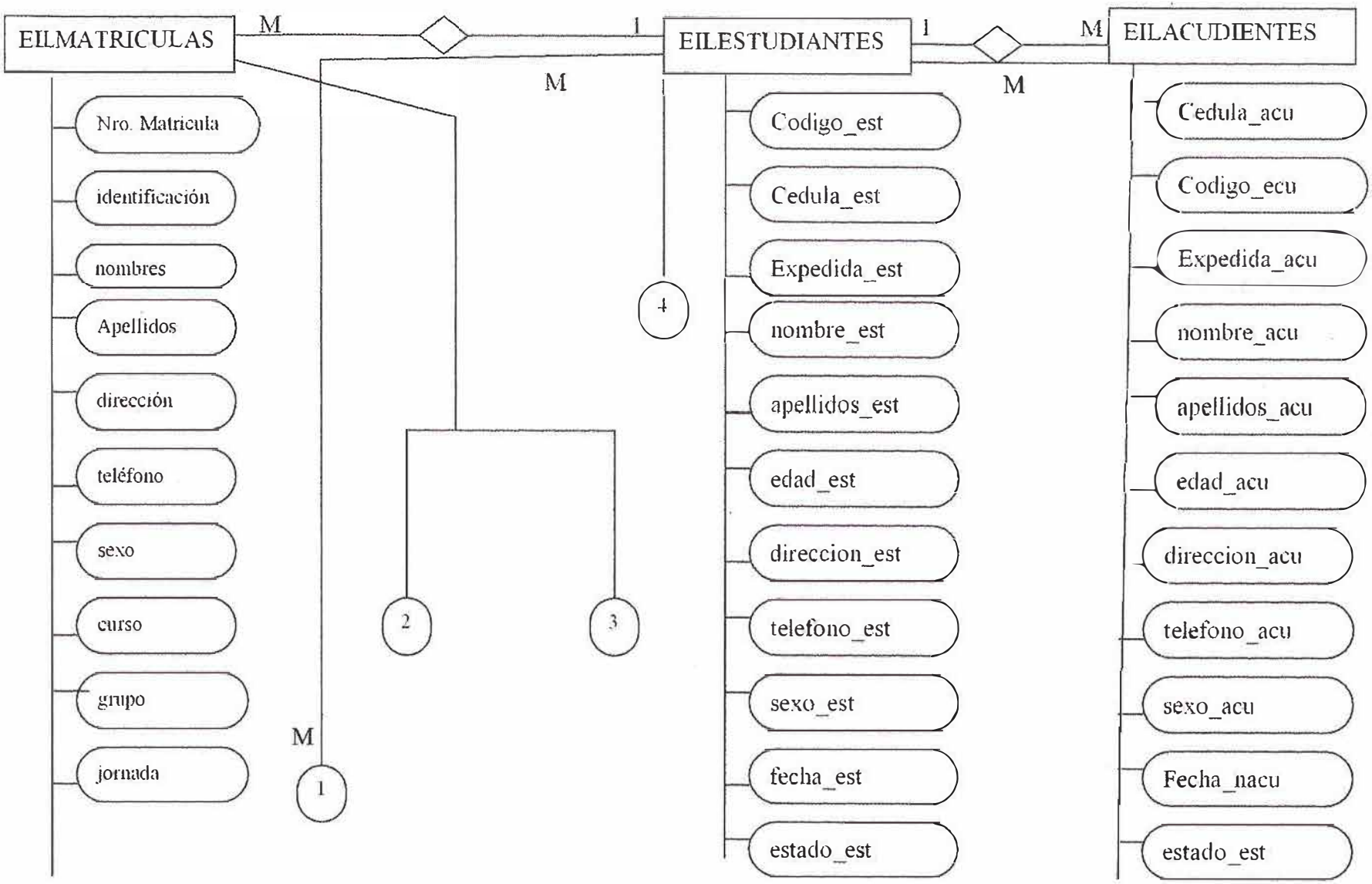
REGISTRO DE NOTAS

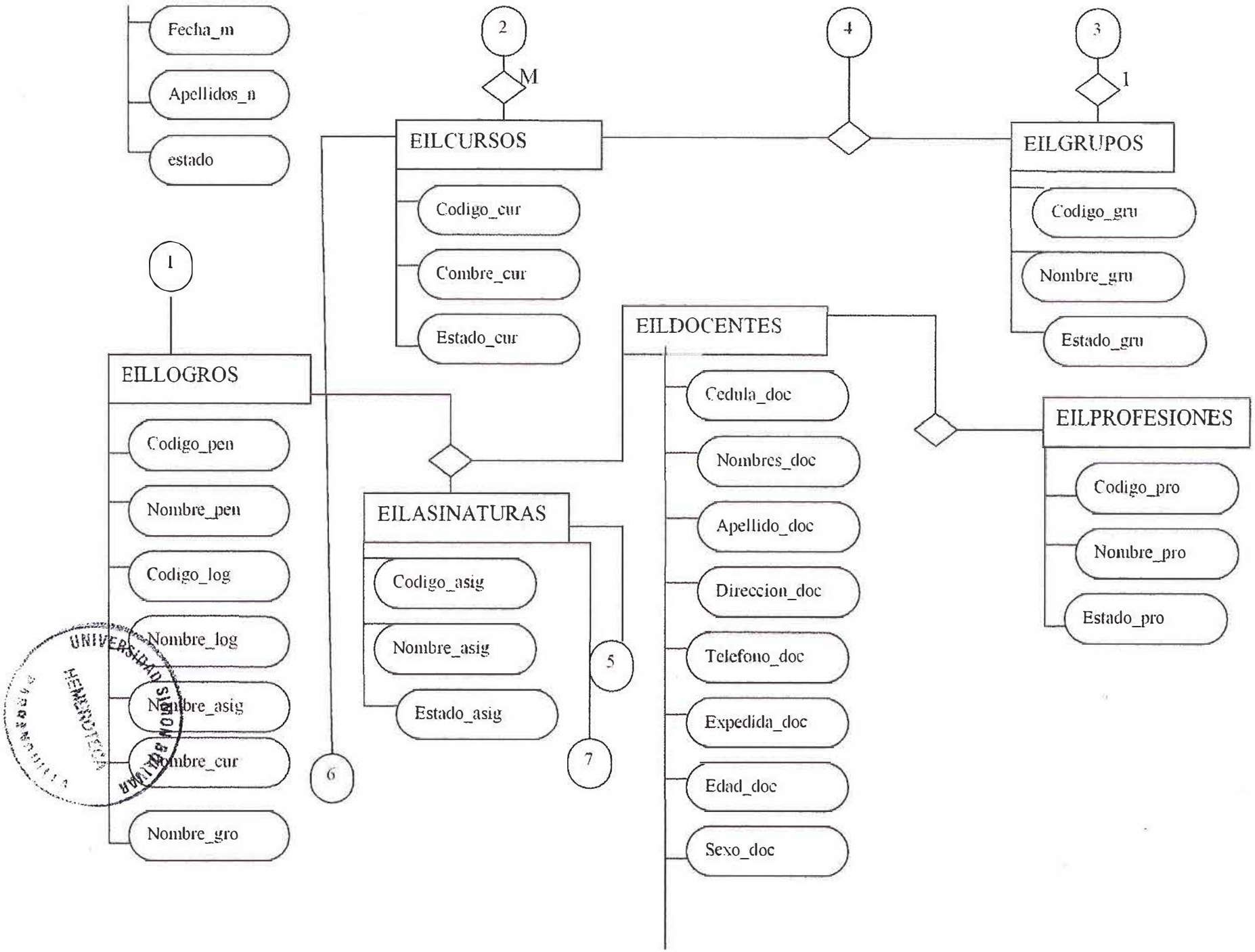


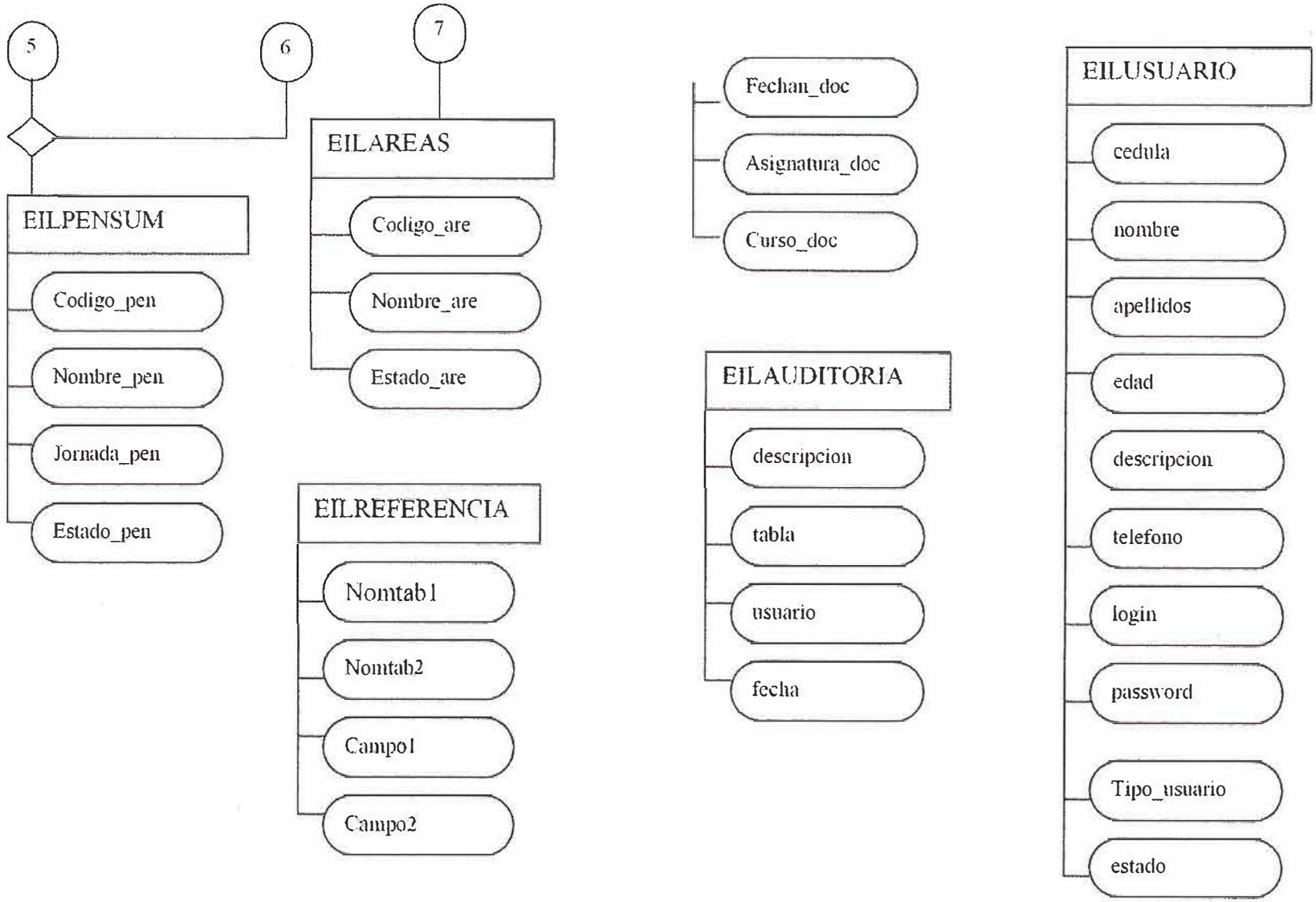
CONSULTAS AVANZADAS



5 MODELO ENTIDAD-RELACION

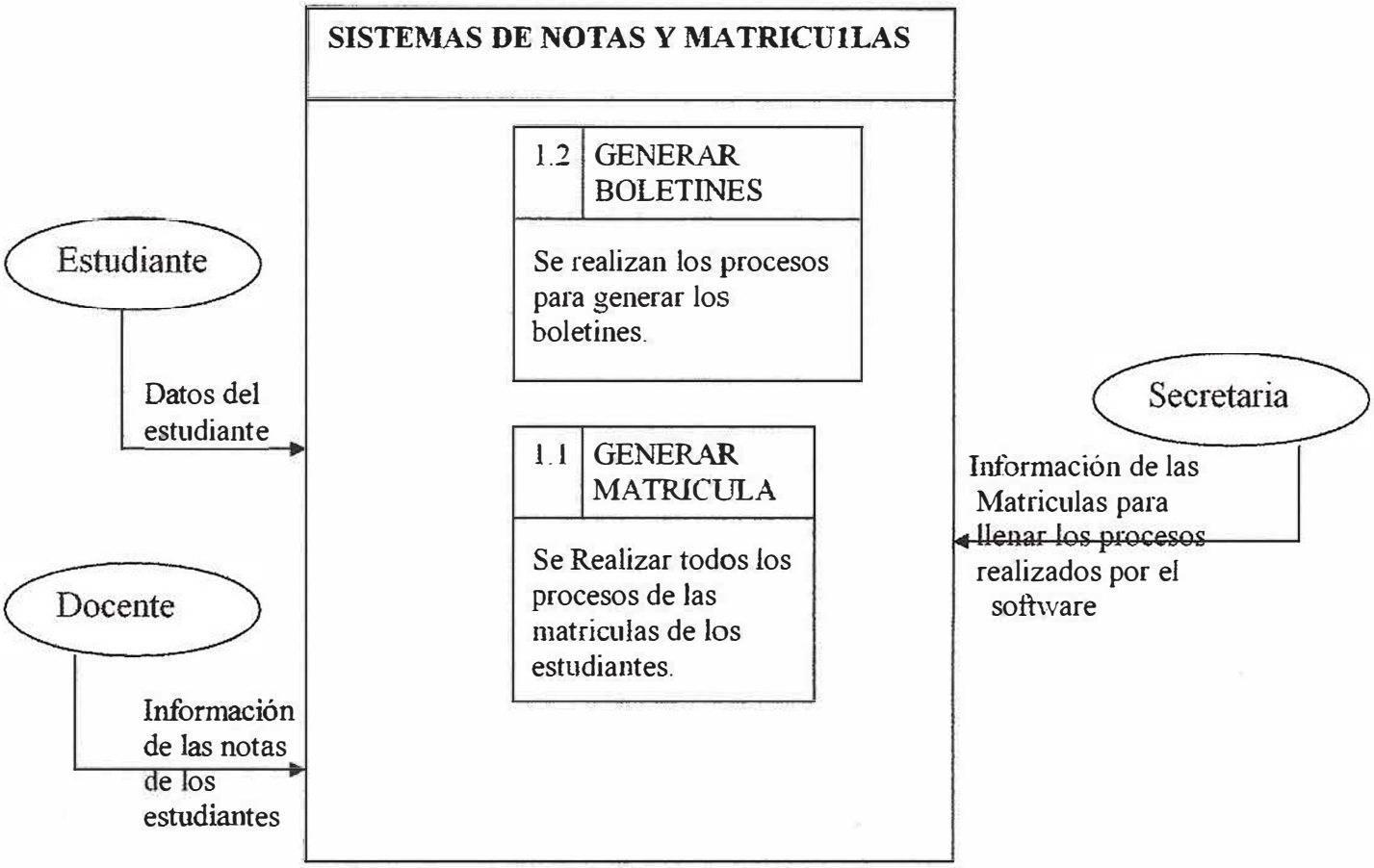



Fecha_m
Apellidos_n
estado
2
M
EILCURSOS
Codigo_cur
Combre_cur
Estado_cur
4
3
1
EILGRUPOS
Codigo_gru
Nombre_gru
Estado_gru
1
EILLOGROS
Codigo_pen
Nombre_pen
Codigo_log
Nombre_log
Nombre_asig
Nombre_cur
Nombre_gro
6
EILASINATURAS
Codigo_asig
Nombre_asig
Estado_asig
5
7
EILDOCENTES
Cedula_doc
Nombres_doc
Apellido_doc
Direccion_doc
Telefono_doc
Expedida_doc
Edad_doc
Sexo_doc
EILPROFESIONES
Codigo_pro
Nombre_pro
Estado_pro

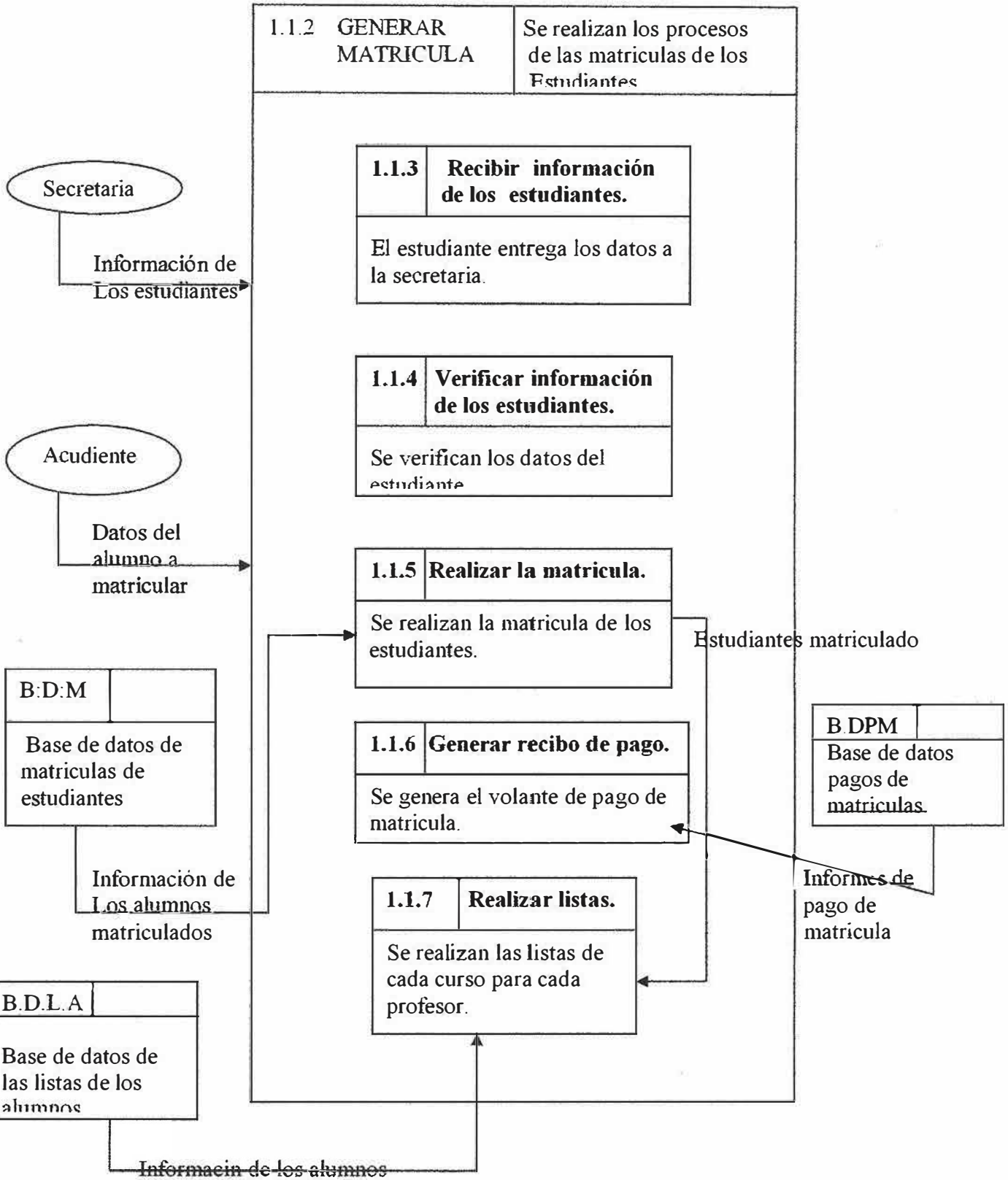

5
6
7
EILPENSUM
Codigo_pen
Nombre_pen
Jornada_pen
Estado_pen
EILAREAS
Codigo_are
Nombre_are
Estado_are
EILREFERENCIA
Nomtab1
Nomtab2
Campo1
Campo2
Fechan_doc
Asignatura_doc
Curso_doc
EILAUDITORIA
descripcion
tabla
usuario
fecha
EILUSUARIO
cedula
nombre
apellidos
edad
descripcion
telefono
login
password
Tipo_usuario
estado

6. DIGARMA DE FLUJO DE DATOS.

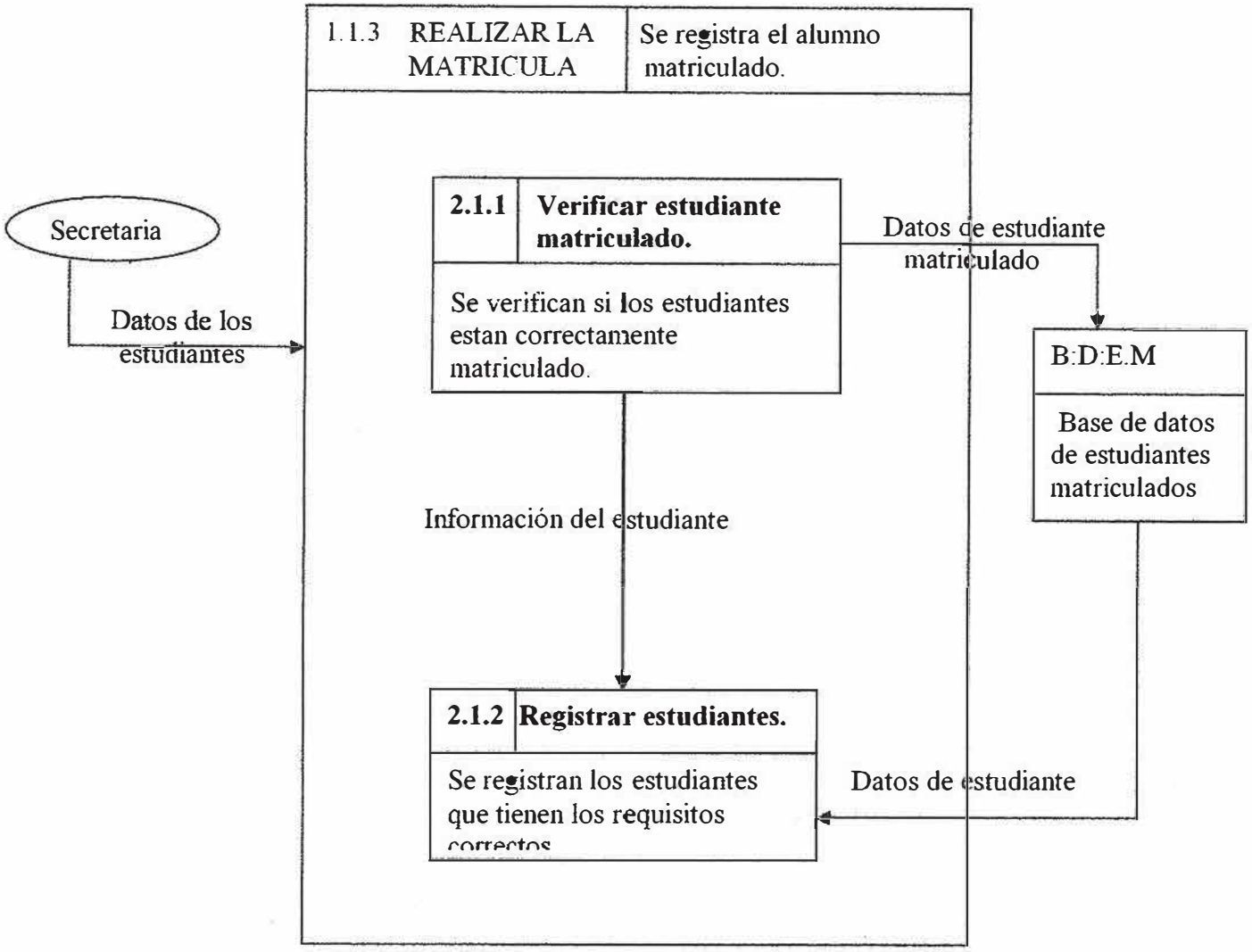
NIVEL 0



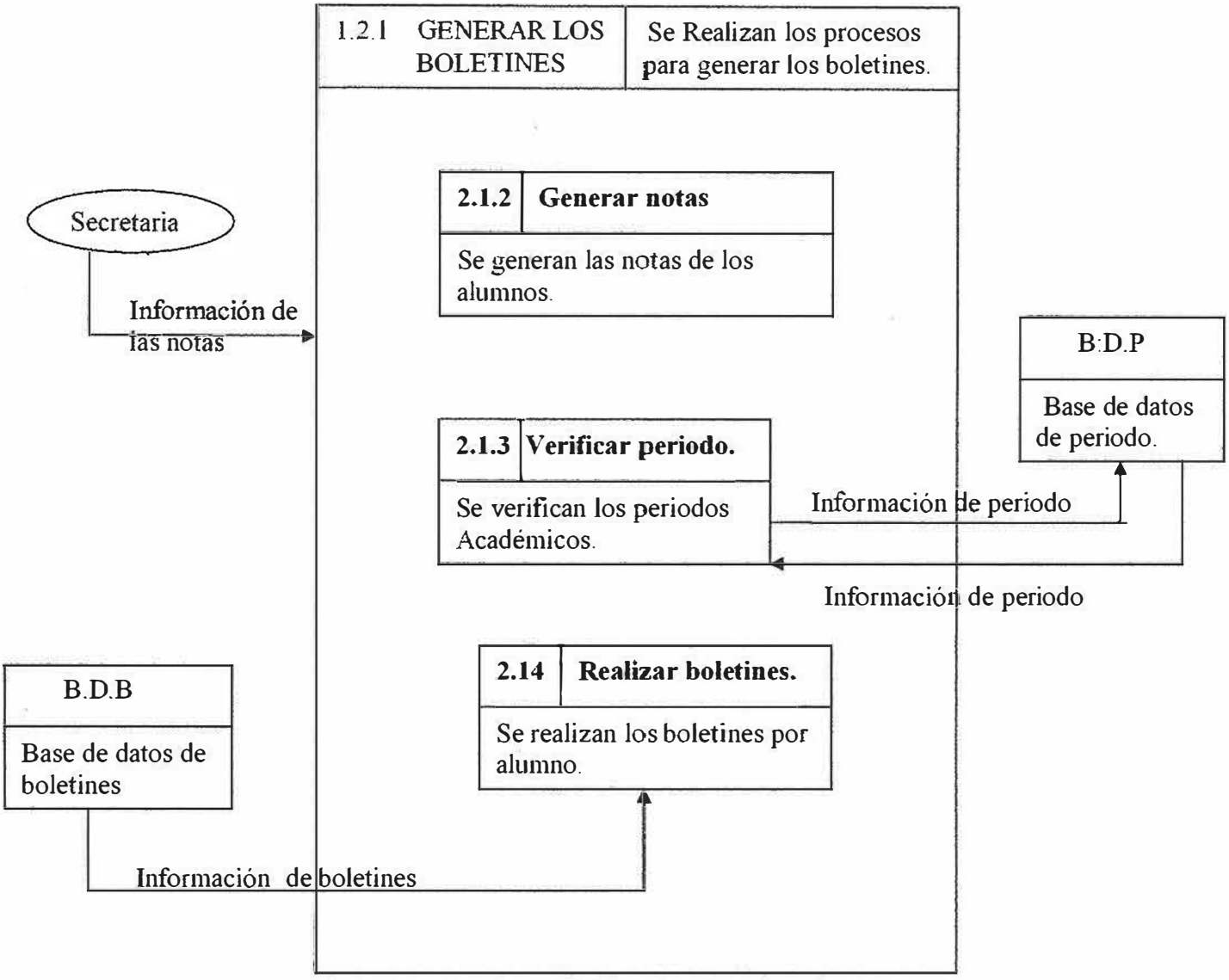
NIVEL 1 MATRICULA



NIVEL 2 MATRICULAS



NIVEL 1 NOTAS



7. DICCIONARIO DE DATOS

Diccionario de datos del diseño e implementación de un sistema de información académico-administrativo para la escuela ana bolívar de consuegra y el colegio José Consuegra Higgins del corregimiento Isabel López (Atlántico).

Flujo de datos	Datos del alumno
Descripción	Los datos de los estudiantes matriculados en la institución.
Proviene de	La entidad: estudiantes
para	El proceso de diseño e implementación de un sistema de información para...

Flujo de datos	Información de notas de estudiantes
Descripción	Es la información sobre las notas de los estudiantes que existen en la institución.
Proviene de	La entidad: Profesor
para	El proceso de diseño e implementación

Flujo de datos	Información de matriculas.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene de	La entidad: Secretaria.
para	El proceso de diseño e implementación.

Proceso	1.1. Generar matricula
Descripción	Genera todos los procesos de matriculas de los estudiantes que estén en la institución.
Entrada	Datos del alumno, información de notas de estudiantes, información de matriculas.
Resumen lógico	Genera los distintos procesos que necesita el sistema para el desarrollo de matriculas.

Proceso	1.2. Generar boletines
Descripción	Genera todos los procesos de notas de los estudiantes que estén en la institución.
Entrada	Datos del alumno, información de notas de estudiantes, información de matriculas.
Resumen lógico	Genera los distintos procesos que necesita el sistema para el desarrollo de boletines.

Flujo de datos	Datos del alumno
Descripción	Los datos de los estudiantes matriculados en la institución.
Proviene de	La entidad. estudiantes
para	El proceso de diseño e implementación de un sistema de información para...

Flujo de datos	Información de notas de estudiantes
Descripción	Es la información sobre las notas de los estudiantes que existen en la institución.
Proviene de	La entidad: Profesor
para	El proceso de diseño e implementación

Flujo de datos	Información de matriculas.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene de	La entidad: Secretaria.
para	El proceso de diseño e implementación.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

Proceso	1.1.2 Recibir información del estudiante.
Descripción	Procesa la información que el estudiante entrega a la secretaria
Entrada	Datos del alumno a matricular, información de estudiantes
Resumen lógico	La información es entregada a la secretaria para continuar con el proceso de matricula.

Proceso	1.1.3 Verificar información del estudiante.
Descripción	Procesa la información de verificación de documentos que el estudiante entrega a la secretaria
Entrada	Datos del alumno a matricular, información de estudiantes.
Resumen lógico	La información es verificada por la secretaria para continuar con el proceso de matricula.

Proceso	1.1.4 Realizar la matricula.
Descripción	Procesa la información del estudiante después de ser matriculado.
Entrada	Datos del alumno a matricular, información de estudiantes.
salida	Información de los alumnos matriculados.
Resumen lógico	La información es verificada por la secretaria para continuar con el proceso de matricula.

Proceso	1.1.5 Generar recibo de pago.
Descripción	Procesa la información para la generación de recibos después que el alumno sea matriculado.
Entrada	Datos del alumno a matricular, información de estudiantes.
salida	Información de los pagos de matricula.
Resumen lógico	Se genera los recibos de pagos después de que el alumnos se haya cancelado el valor de la matricula.

Proceso	1.1.6 Realizar lista de estudiantes.
Descripción	Procesa la información para la realización de listas de los estudiantes matriculados.
Entrada	Datos del alumno a matricular, información de estudiantes. información de estudiantes matriculados.
salida	Información de los alumnos
Resumen lógico	Se generan las lista después de que el alumnos haya cancelado y matriculado.

Almacén de datos	Estudiantes matriculados
Descripción	almacena la información del estudiante después de ser matriculado.
Flujos de datos recibidos	Información de los alumnos matriculados.
Flujos de datos proporcionado	ninguno

Almacén de datos	Pagos de matriculas
Descripción	almacena la información de los pagos de los estudiantes.
Flujos de datos recibidos	Información de los pagos de matricula.
Flujos de datos proporcionado	ninguno

Almacén de datos	Pagos de matriculas
Descripción	almacena la información de las listas de estudiantes matriculados en la institución.
Flujos de datos recibidos	Información de los alumnos.
Flujos de datos proporcionado	ninguno

Flujo de datos	Información de alumnos matriculados.
Descripción	Es la información sobre las matriculas de los estudiantes que existen en la institución.
Proviene del	proceso: realizar matriculas.
para	Base de datos de estudiantes matriculados.

Flujo de datos	Información de los pagos de matricula.
Descripción	Es la información sobre los pagos de matriculas de los estudiantes que existen en la institución.
Proviene del	proceso: generar recibos de pagos.
para	Base de datos de pagos de matricula.

Flujo de datos	Información de alumnos.
Descripción	Es la información sobre los estudiantes que existen en la institución.
Proviene del	proceso: realizar listas.
para	Base de datos de listas de estudiantes matriculados.

Proceso	1.2.1 Verificar estudiante matriculado.
Descripción	Se verifica la información para la realización de matricula.
Entrada	Datos del alumno, datos del estudiante.
salida	Información de los alumnos.
Resumen lógico	Se verifica la información de los documentos de los estudiantes para la elaboración de la matricula.

Proceso	1.2.2 Registrar estudiantes.
Descripción	Se registran los estudiantes que tengan los requisitos al dia
Entrada	Información de los alumnos.
salida	Datos de estudiantes matriculados.
Resumen lógico	Después de ser verificados los documentos se procede a registrar a los estudiantes.

Flujo de datos	Datos de estudiante matriculado.
Descripción	Es la informacion sobre los estudiantes registrados
Proviene del	proceso: registrar estudiantes.
para	Base de datos de estudiantes matriculados.

Flujo de datos	Información de estudiante.
Descripción	Es la información sobre los estudiantes que fueron verificados por la institución.
Proviene del	proceso: verificar estudiantes matriculados.
para	Registrar estudiantes.

Flujo de datos	Datos de estudiantes.
Descripción	Es la información sobre los estudiantes que existen en la institución.
Proviene del	Base de datos estudiantes matriculados.
para	Verificar estudiantes matriculados.

Almacén de datos	Estudiantes matriculados.
Descripción	almacena la información de los estudiantes matriculados en la institución.
Flujos de datos recibidos	Datos de estudiantes matriculados
Flujos de datos proporcionado	Datos de estudiante.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

Proceso	1.2.3 Generar notas.
Descripción	Se generan las notas de los estudiantes para la generación de los boletines.
Entrada	Información de notas.
salida	ninguno.
Resumen lógico	Se genera las diferentes notas por materia y alumnos para la elaboración de boletines.

Flujo de datos	Información de notas.
Descripción	Es la información sobre las notas de los estudiantes
Proviene del	La entidad: secretaria.
para	Generar notas.

Proceso	1.2.4 Verificar periodo
Descripción	se verifica el periodo de los estudiantes para la elaboración de los boletines.
Entrada	Información de periodo.
salida	Información de periodo.
Resumen lógico	Se verifica el periodo del estudiante.

Flujo de datos	Información de periodo.
Descripción	Es la información sobre los periodos de cada uno de los estudiantes en la institución.
Proviene del	Proceso: verificar periodo.
para	Base de datos de periodo académico.

Flujo de datos	Información de periodo.
Descripción	Es la información sobre los periodos de cada uno de los estudiantes en la institución.
Proviene de la	Base de datos de periodo académico.
para	verificar periodo.

Proceso	1.2.5 Realizar boletines
Descripción	Se realizan los boletines de cada estudiantes con sus respectivas notas.
Entrada	Ninguno.
salida	Información de boletines.
Resumen lógico	Se realizan los boletines de los estudiantes.

Flujo de datos	Información de boletines.
Descripción	Es la información sobre los boletines de cada uno de los estudiantes en la institución.
Proviene del	Proceso: realizar boletines.
para	Base de datos de boletines.

Almacén de datos	Boletines.
Descripción	almacena la información de los boletines de cada estudiante matriculado en la institución.
Flujos de datos recibidos	Información de boletines.
Flujos de datos proporcionados	Ninguno.

Almacén de datos	Periodo académico.
Descripción	Almacena la información de los periodos académicos de los estudiantes matriculados en la institución.
Flujos de datos recibidos	Información de periodo.
Flujos de datos proporcionado	Información de periodo.

DISEÑO E IMPLEMENTACION DE UN SISTEMA DE INFORMACION ACADEMICO-ADMINISTRATIVO PARA LA ESCUELA ANA BOLIVAR DE CONSUEGRA Y EL COLEGIO JOSE CONSUEGRA HIGGINS DEL CORREGIMIENTO ISABEL LOPEZ (Atlántico).

MANUAL DEL USUARIO

Juan Carlos Casas

Ricardo Cortes

Osvaldo Villalba

CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO

SIMÓN BOLÍVAR

FACULTAD DE INGENIERIA DE SISTEMAS

AREA INVESTIGACIÓN FORMATIVA IV

BARRANQUILLA

2005

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

TABLA DE CONTENIDO

		Pág.
1	Introducción.	
2	Antes de Comenzar.	
2.1	Convenciones Generales.	
3	Para que sirve el sistema de información Académico-Administrativo.	
4	Términos usados en este manual.	
5	Entrar al sistema de información Académico-Administrativo.	
6	Qué hay en la pantalla del sistema información Académico-Administrativo.	
7	Salir del sistema de información. Académico-Administrativo	

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

ANTES DE COMENZAR

Convenciones generales.

El sistema de información Académico-Administrativo para la escuela y colegio de Isabel López tiene como convenciones generales los siguientes puntos:

- Primero que todo, el cliente tiene que tener un equipo de primera, que tenga una gran capacidad de almacenamiento, una buena Memoria, un procesador de alta velocidad para realizar todos los procesos de una manera rápida y eficaz.

- Después de tener el equipo el cliente debe tener instalado en el equipo el programa Microsoft Visual Estudio 6.0 con la herramienta Microsoft Visual Basic para el debido funcionamiento del software.

- También tiene que instalar el programa de la base de datos que es MySql para poder realizar todos los procesos que exige el software tales como: insertar, modificar, eliminar, etc.

- Además el cliente tiene que tener instalado el programa Crystal Report 7.0 para poder realizar los reportes del sistema.

PARA QUE SIRVE EL SISTEMA DE INFORMACION ACADEMICO-ADMINISTRATIVO PARA LA ESCUELA Y COLEGIO ISABEL LOPEZ

El sistema de información Académico-Administrativo para la escuela y colegio Isabel López es un software para el manejo de la información que se maneja en la institución.

Este software se realizo con el fin de mejorar la forma de realización de los procesos que se manejan en la institución, ya que estos anteriormente se realizaban manualmente y esto ocasionaba retrasos y perdidas en los documentos.

Con la implementación de este software al institución no tendrá que volver a manejar la información de esa forma tan obsoleta como lo es manualmente. Con este software el cliente puede realizar todos sus procesos con mayor rapidez, puede insertar y modificar estudiantes, acudientes, profesores en solo segundos. Además este software permitirá a la institución tener un mejor control y visualización de todos sus procesos.

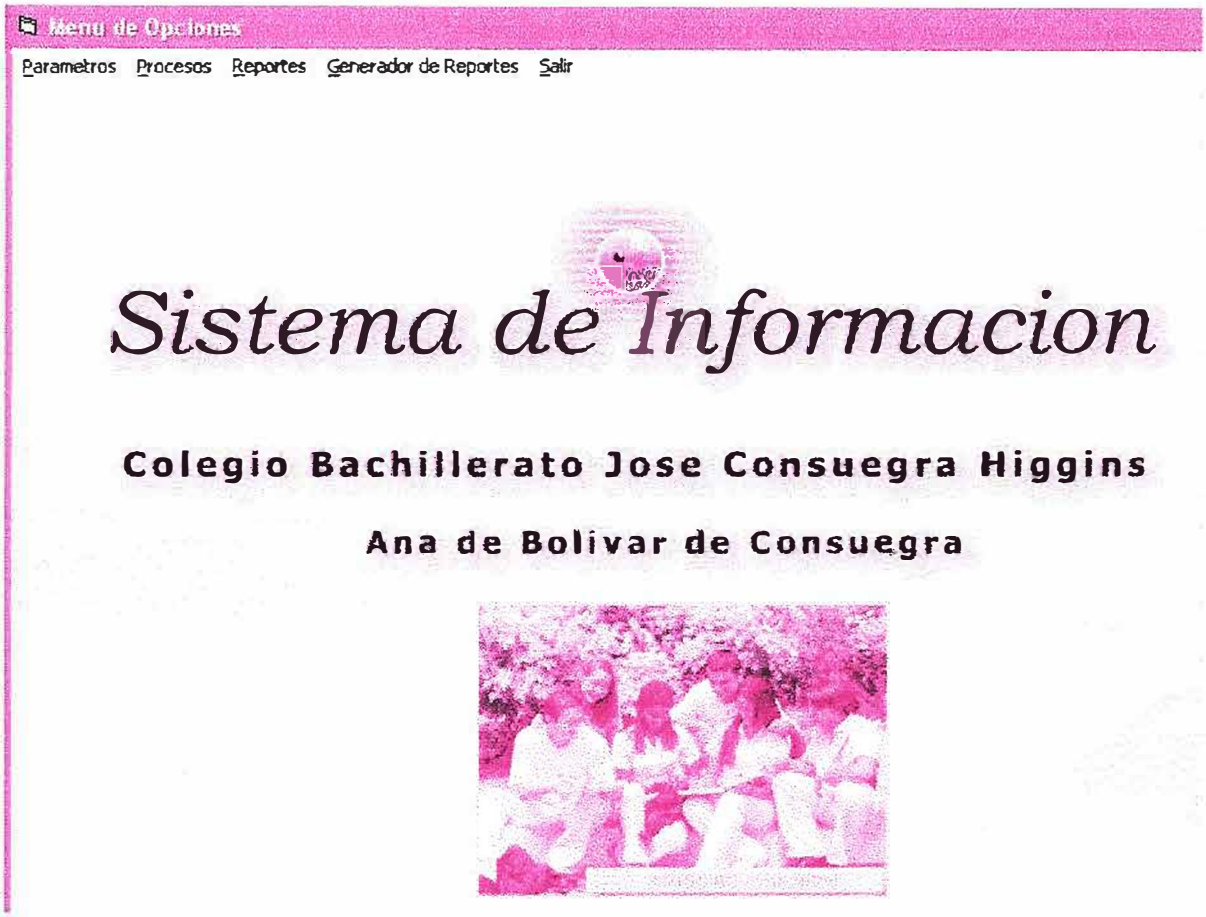
FORMULARIO DE INGRESO AL SISTEMA

Este es unos de los formularios más importantes por medio del cual podemos ingresar al sistema utilizando un nombre y un password.

Este formulario contiene des botones, los cuales son:

Botón Aceptar: Permite el ingreso del usuario al sistema.

Botón Cancelar: Este botón cierra la ventana.



FORMULARIO MENU DE OPCIONES.

En esta ventana se encuentran todas las acciones que realiza el sistema en general, es decir que desde aquí se manejan todos los procesos que realiza este software, este Menú contiene las siguientes opciones: Parámetros, procesos, reportes, generador de

reportes y salir, cada una de estas opciones contiene sus propias subopciones.

MENU DE OPCIONES "OPCION PARAMETROS"

En esta opción de parámetros se encuentra un submenú de opciones que contiene todos los registros que maneja el sistema, entre estos tenemos: el registro de estudiantes, de docentes, de asignaturas, de cursos, de grupos ,de áreas, de profesiones, de pensum y de logros.

FORMULARIO DE REGISTRO DE ESTUDIANTES Y ACUDIENTES

Este formulario permite al usuario insertar todos los datos personales de los estudiantes y acudientes que desean ingresar a la institución. Las dos pantallas que aparecen en la parte inferior del formulario muestran todos los acudientes y estudiantes que ya han sido ingresados a la base de datos de la institución.
Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO REGISTRO DE DOCENTES.

En este formulario se ingresan los datos personales del docente y las asignaturas que el dicta y en los cursos y grados en que da las clases, además contiene los siguientes botones:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para guardar todos los datos del Docente en la base de datos.

CONSULTAR: Este botón permite consultar tos los docentes que Existen en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente del docente en ala base de datos.

ELIMINAR: Este botón permite eliminar un docente de la base de datos.

IMPRIMIR: Este botón permite imprimir los docentes que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO DE REGISTRO DE ASIGNATURAS

En este formulario el usuario puede registrar todas las asignaturas que se dictan en el colegio, pero antes de realizar el registro de asignaturas se debe realizar el registro de las áreas, porque cada asignatura pertenece a un área especifica. En este formulario se coloca el código de la asignatura, su nombre, su área a la que pertenece y su estado.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

HEMEROTECA

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

FORMULARIO REGISTRO DE CURSOS

Este formulario permite al usuario insertar todos los cursos que existen en el colegio, a estos cursos se le asigna un código se coloca el nombre del curso y su estado que puede ser activo o inactivo.
Este formulario tiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO REGISTRO DE GRUPOS

Este formulario permite al usuario ingresar todos los grupos manejados en la institución, colocando su código, nombre y estado. Este formulario contiene los siguientes botones que son :

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO REGISTRO DE AREAS

Este formulario permite al administrador ingresar todas las áreas que se manejen en el colegio ingresando el código del área, su nombre y el estado de esta, este estado puede ser activo o inactivo.
Este formulario contiene de varios botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO DE REGISTRO DE PROFESIONES

Este formulario permite al usuario llevar el registro de todas las profesiones de cada docente. A estas profesiones se le asigna un código, se coloca el nombre de la profesión y su estado ya sea activo o inactivo.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

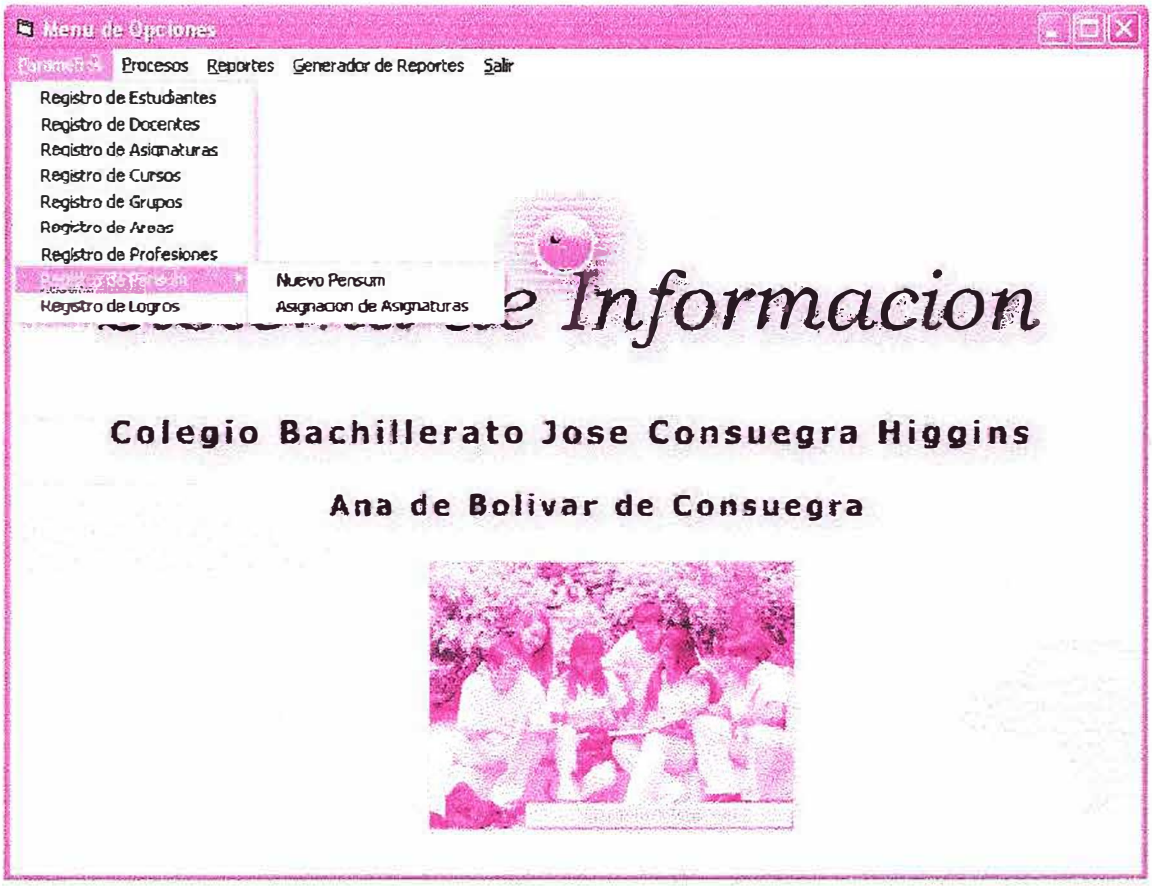
MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA



MENU DE OPCIONES "SUBMENU REGISTRO DEL PENSUM".

En este menú se encuentra el submenú del registro del pensum y asignación de asignaturas.

FORMULARIO DE REGISTRO DE PENSUM

Este formulario permite al usuario insertar en la base de datos los pensum correspondientes. el pensum se registra con un código, su nombre, la jornada a la que corresponda y su estado ya sea activo o inactivo.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

FORMULARIO DE REGISTRO DE ASIGNATURAS AL PENSUM

Este formulario permite al usuario ingresar todas las materias al pensum que corresponda.

El usuario ingresa el código y el nombre del pensum y luego oprime el botón cargar asignaturas y aromáticamente se carga el combo con las asignaturas, selecciona la asignatura que el usuario desee y luego puede guardar.

Este formulario consta de una serie de botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

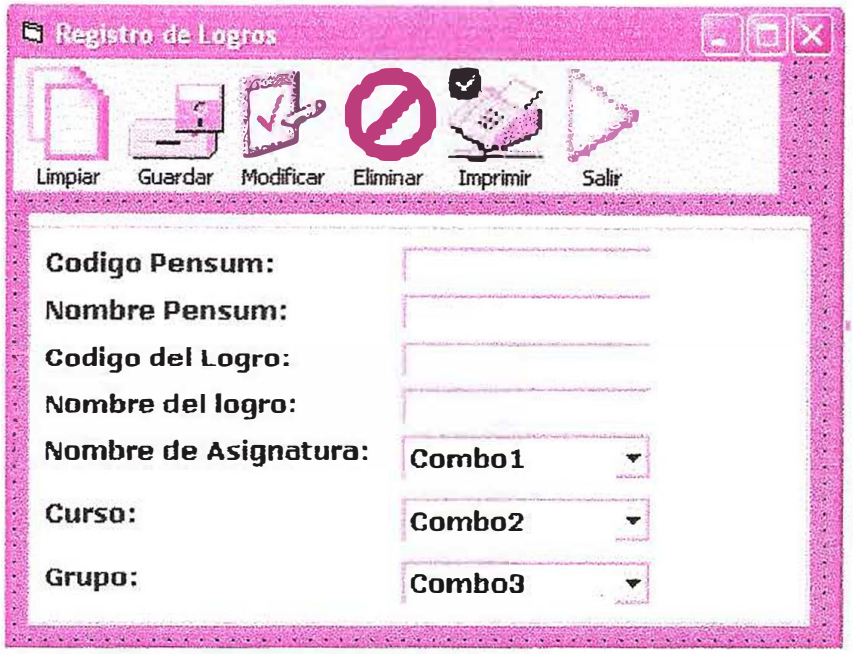
MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.

UNIVERSIDAD SIMÓN BOLÍVAR
HEMEROTECA
BARRANQUILLA



FORMULARIO DE REGISTRO DE LOGROS

Este formulario permite al usuario registrar los logros correspondientes a las asignaturas. Las asignaturas se registran con un código y un nombre, este nombre viene siendo la descripción de este logro. En este formulario colocamos el código del pensum y su nombren estoe s para saber a que pensum pertenece este logro, además tenemos el nombre de la asignatura , el curso y el grupo todo esto es para que al momento de realizar los boletines el usuario sepa de que curso, grupo y asignatura pertenece ese logro.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

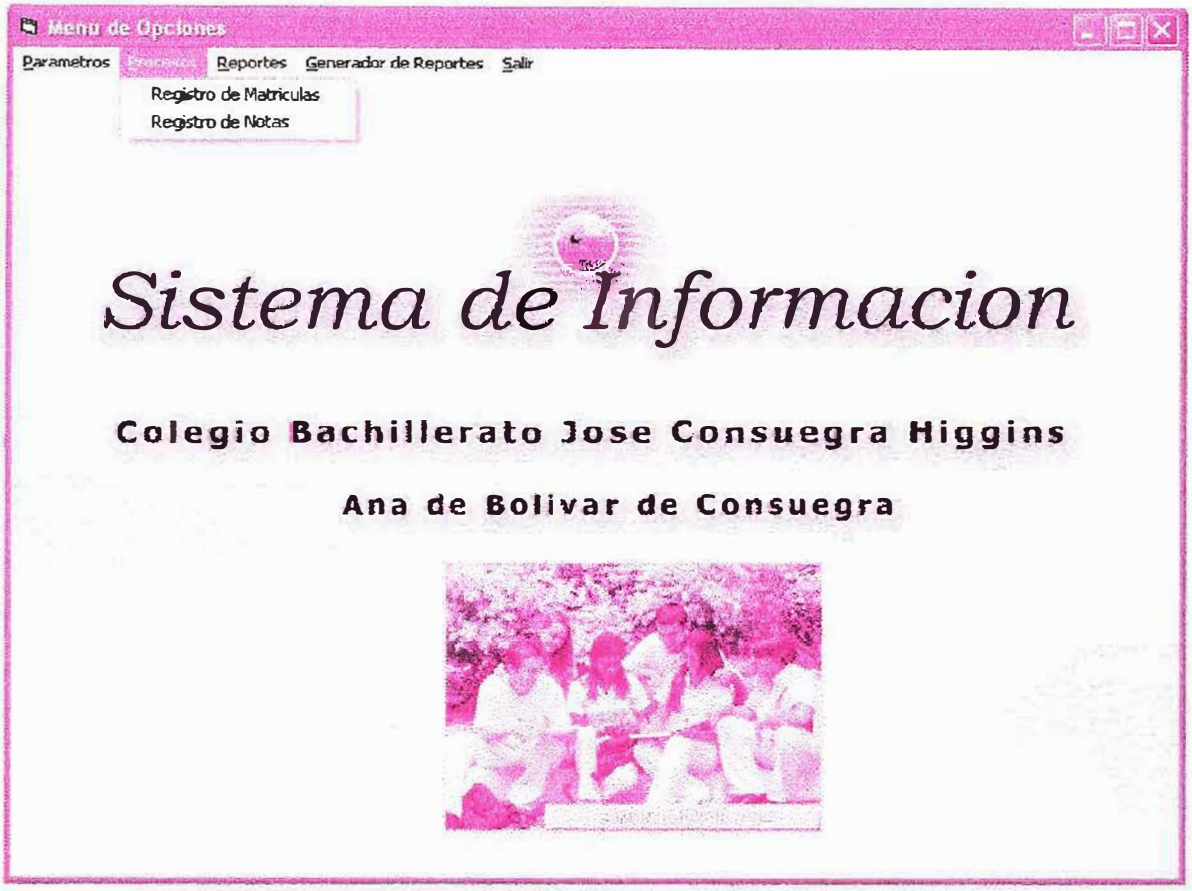
CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.



MENU DE OPCIONES "PROCESOS"

Este menú de opciones muestra al usuario los procesos que se realizan en el sistema. Estos procesos son: Registro de matriculas y registro de notas.

FORMULARIO DE REGISTRO DE MATRICULAS

Este formulario permite al usuario realizar la matricula correspondiente a los estudiantes que desean ingresar a la institución, pero antes de ser matriculados los estudiantes deben ser

ingresados a la base de datos por medio del formulario registro de estudiantes.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

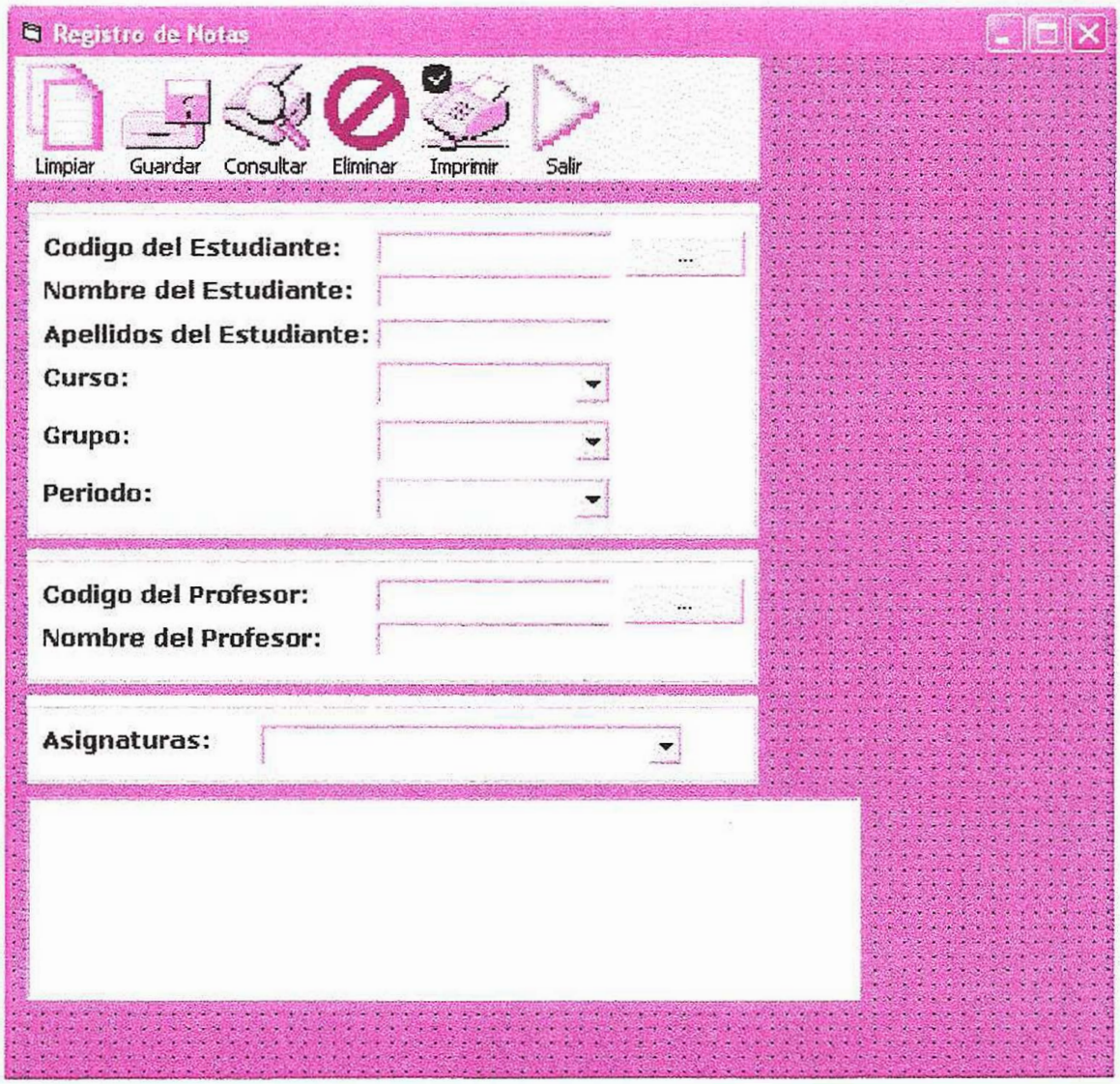
CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.



FORMULARIO REGISTRO DE NOTAS

Este formulario permite al usuario crear los boletines de todos los estudiantes.

El usuario presiona y automáticamente se muestran los códigos de los estudiantes, el usuario escoge un código y se cargan los datos de este mismo en las cajas de texto.

El usuario selecciona el curso, grado y periodo al que correspondan las notas de ese estudiante, además puede seleccionar el otro para cargar el código y nombre del docente. En la parte inferior selecciona el combo de las asignaturas y este carga todas las asignaturas que han sido ingresadas a ala base de datos y son cargados los logros en el espacio en blanco que esta en la parte inferior del formulario.

Luego se le presiona el botón imprimir y este nos imprime el reporte de los boletines.

Este formulario contiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

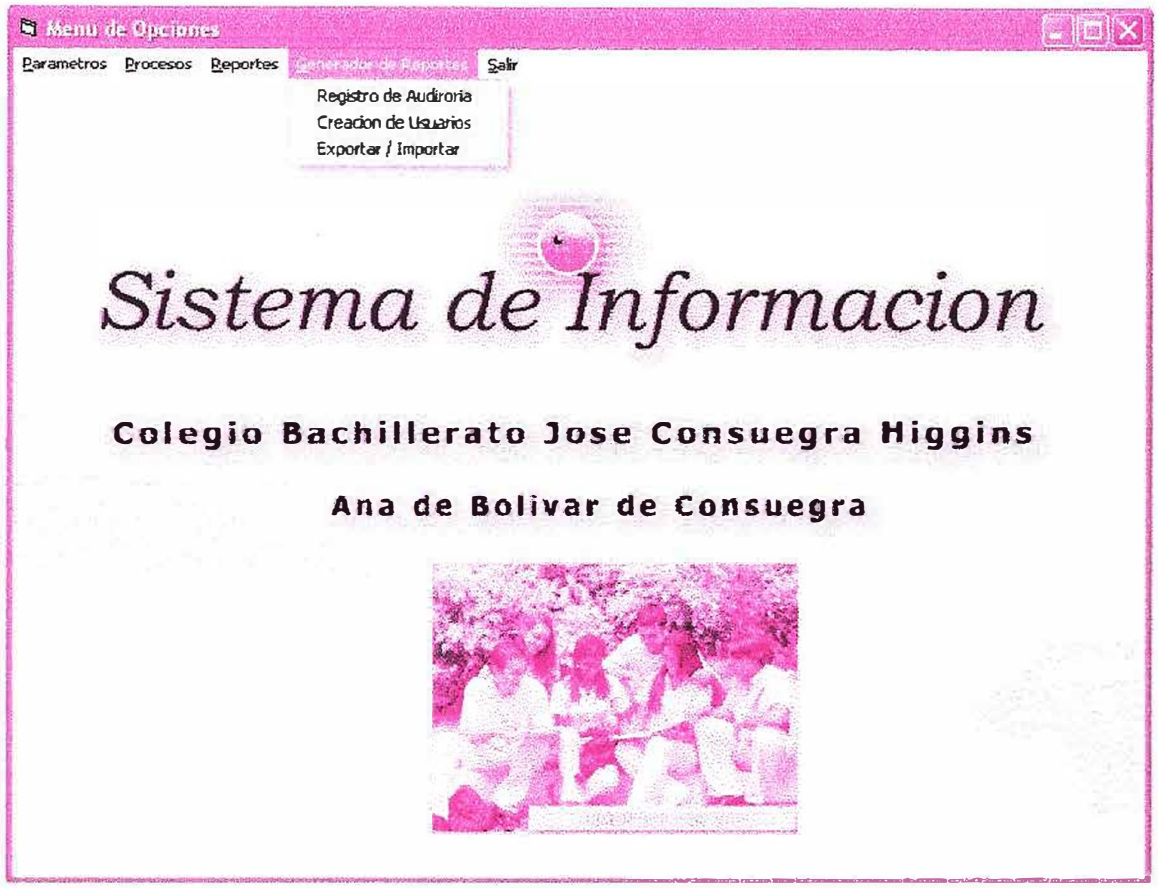
CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

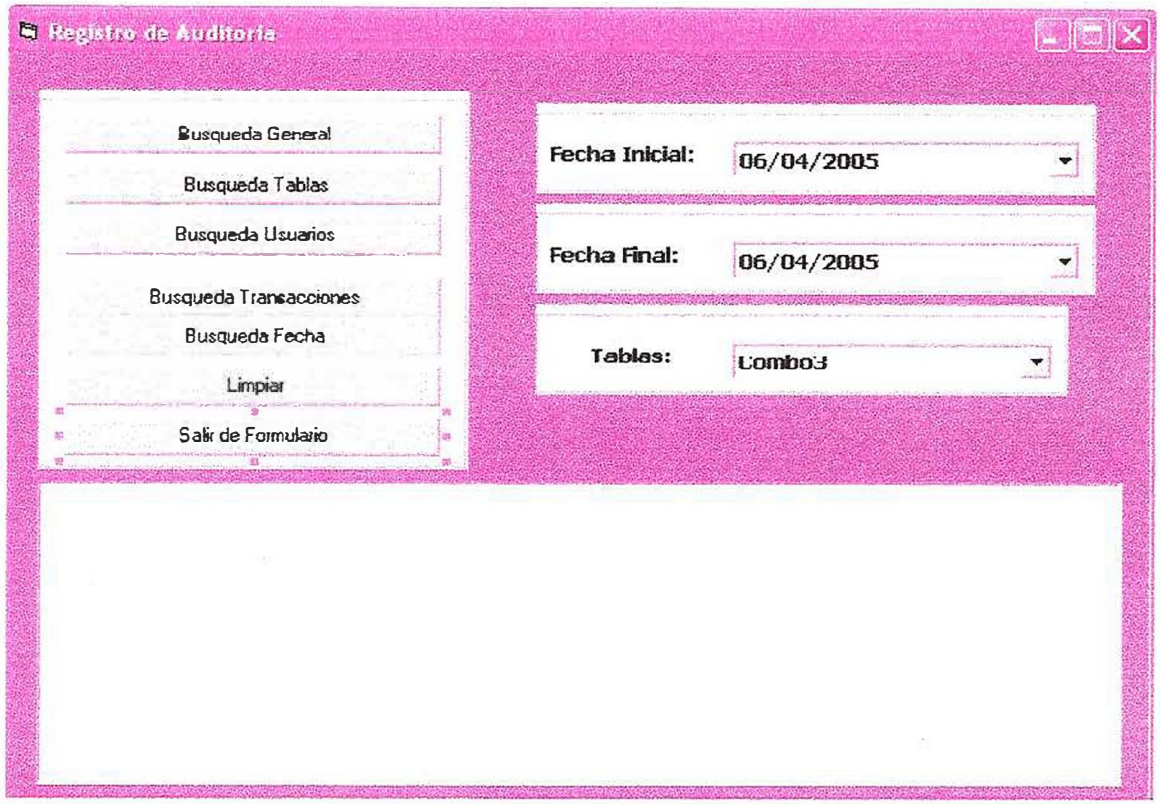
IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.



MENU DE OPCIONES GENERADOR DE REPORTES

Esta opción permite al usuario realizar los diferentes reportes que maneja el sistema. En esta opción encontramos el registro de auditoria, la creación de usuarios y exportar e importar datos del sistema.



FORMULARIO REGISTRO DE AUDITORIA

Este formulario permite al usuario identificar las diferentes modificaciones que se realizaron en el sistema y quien fue el que las realizó.

Búsqueda general: Muestra todos los registros de la tabla Auditoria.

Búsqueda tablas: Muestra los registros por tablas.

Búsqueda Usuarios: Muestra todos los registros por usuario.

Búsqueda por Transacciones: Muestra todos los registros hechos por el usuario según la transacción que haya hecho.

Búsqueda por fecha: Muestra los registros según la fecha que se Estipule.

Limpiar: Limpia todos los campos del formulario.

Salir del sistema: Sirve para salir del formulario.

FORMULARIO CREACION DE USUARIO

Este formulario permite al usuario crear los diferentes usuarios que van a manejar el sistema, el usuario toma los datos a la persona que quiera registrase, además le crea un login y un password, también le agrega un tipo de usuario a esta persona como puede ser el de administrador.

Este formulario tiene los siguientes botones que son:

LIMPIAR: Este botón sirve para limpiar los campos de la pantalla.

GUARDAR: Este botón sirve para almacenar todos los registros en la base de datos.

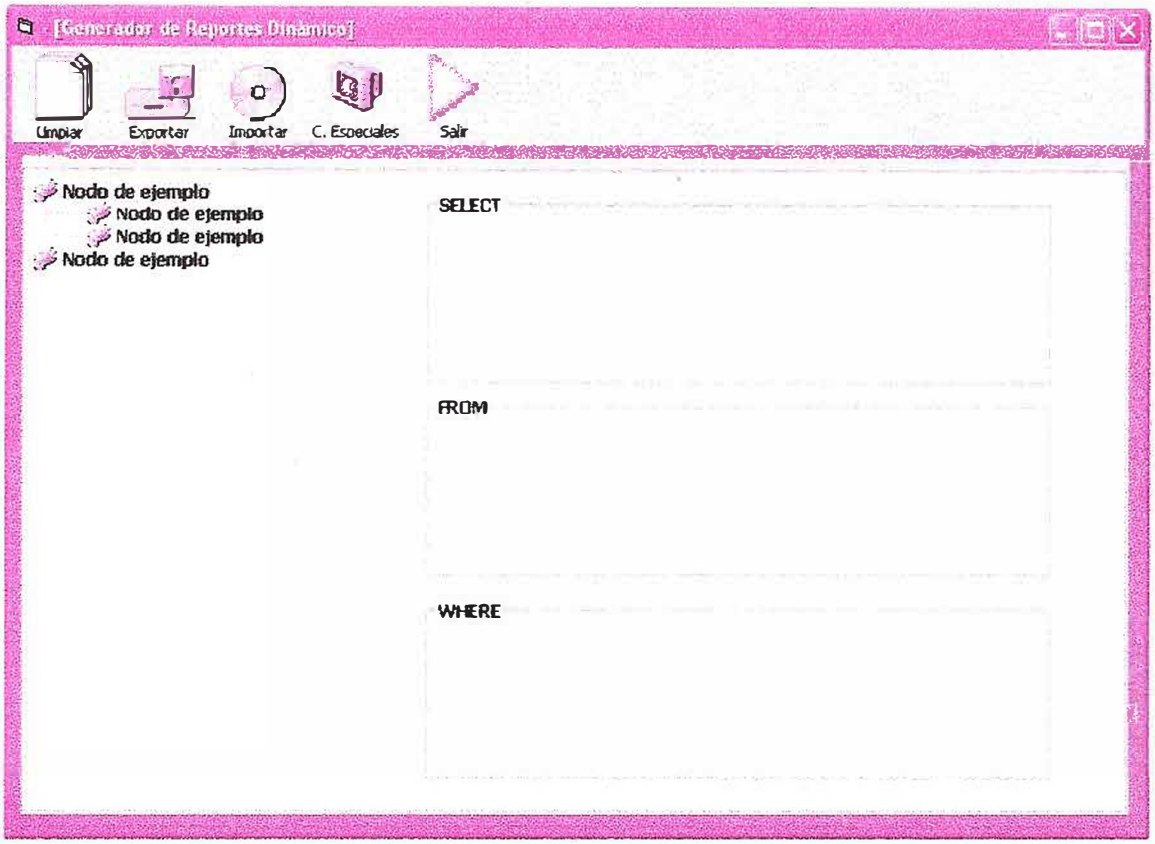
CONSULTAR: Este botón permite consultar los registros ya Insertados en la base de datos del sistema.

MODIFICAR: Este botón permite modificar un registro existente en la base de datos.

ELIMINAR: Este botón permite eliminar un registro de la base de datos.

IMPRIMIR: Este botón permite imprimir los registros que existen en la base de datos.

SALIR: Este botón permite cerrar la ventana y regresar al menú principal.



FORMULARIO GENERADOR DE REPORTES DINAMICOS

Este formulario permite al usuario realizar las consultas que el desee.

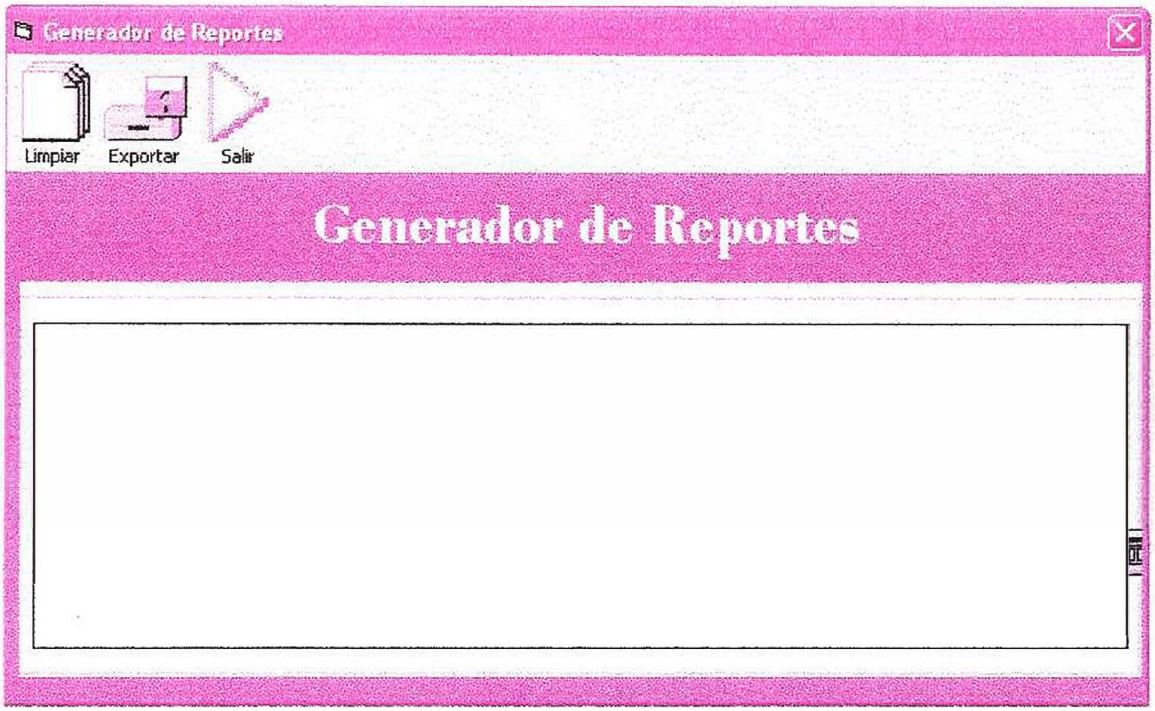
LIMPIAR: Permite al usuario limpiar los campos del formulario.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

EXPORTAR: Este botón sirve para realizar la carga de los registros obtenidos de la base de datos a un archivo de textos.

IMPORTAR: Este botón permite al usuario realizar la carga de los registros obtenidos del a archivo te texto a la base de de datos.

CONSULTAS ESPECIALES: Este botón permita a usuario restringir las consultas.



FORMULARIO GENERADOR DE REPORTES

"IMPORTAR"

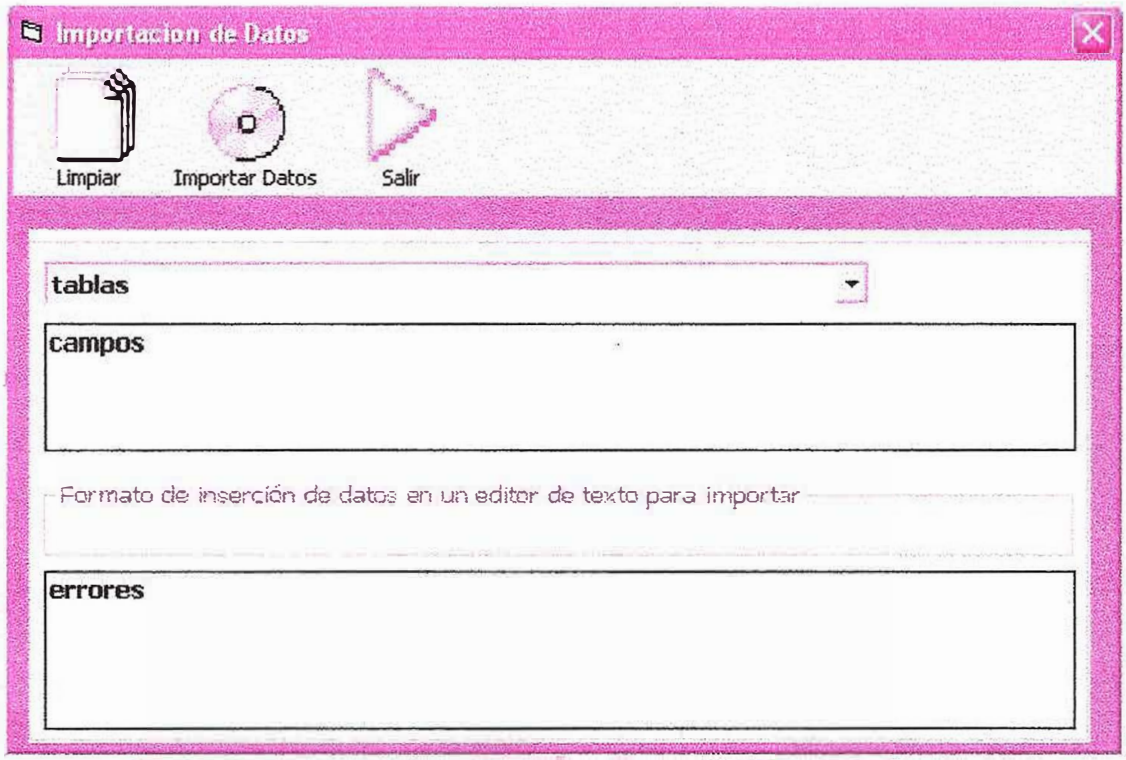
Este formulario permita al usuario crear las exportaciones de la base de datos a un archivo de texto.

LIMPIAR: Sirve para limpiar los campos del formulario.

EXPORTAR: Exporta los datos de la base de datos hacia un archivo de textos.

UNIVERSIDAD SIMON BOLIVAR
HEMEROTECA
BARRANQUILLA

SALIR : Este botón permite al usuario salir del formulario.



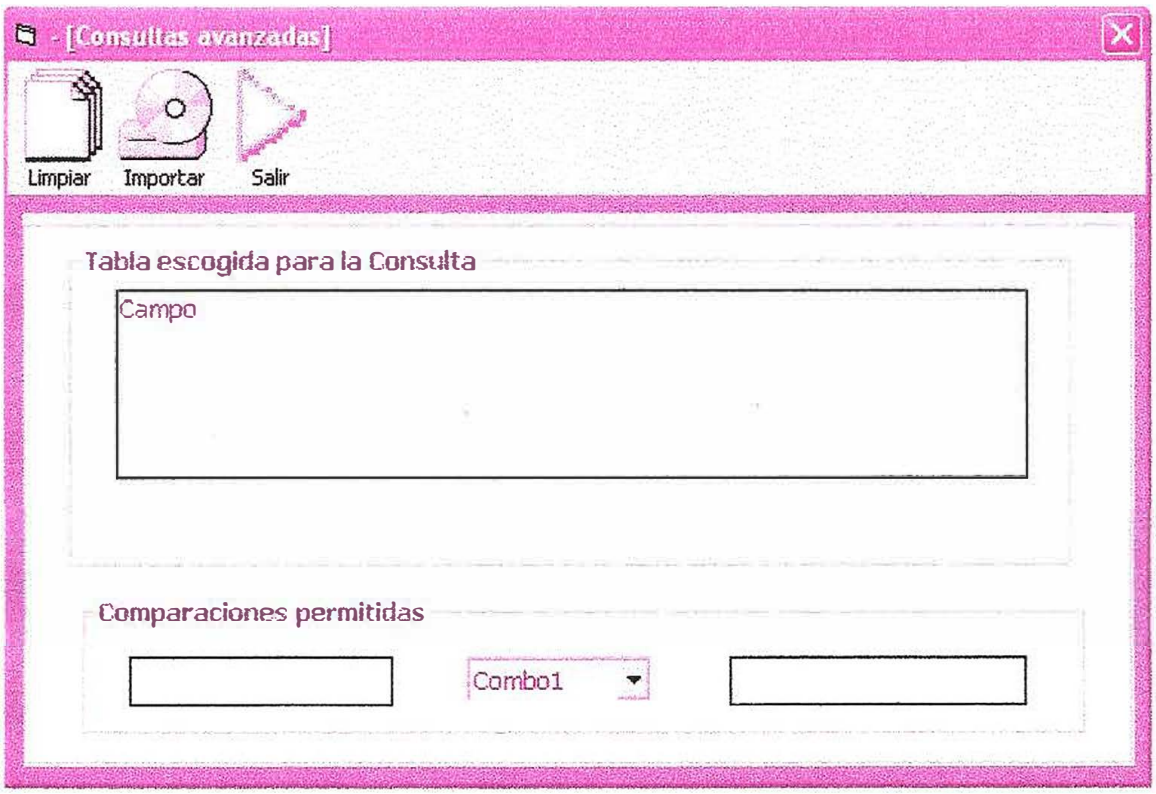
FORMULARIO DE IMPORTACION DE DATOS

Este formulario permite al usuario cargar un registro desde un archivo de texto a la base de datos.

LIMPIAR: Sirve para limpiar los campos del formulario.

IMPORTAR: Sirve para cargar el archivo de texto.

SALIR: Este formulario permite al usuario salir del formulario.



FORMULARIO DE CONSULTAS AVANZADAS

Este formulario permite al usuario restringir todas las consultas que se realicen en el sistema.