

**SISTEMA DE INFORMACION ACADEMICO-ADMINISTRATIVO
PARA LA GESTION DE DATOS EN LOS DEPARTAMENTOS DE
MATRICULAS Y PROCESOS ACADEMICOS EN LA
INSTITUCION EDUCATIVA DISTRITAL SAN SALVADOR DE
BARRANQUILLA**

**ELIAS ALBOR ARRIETA
LUIS BARROS ORTEGA
MARLON INSIGNARES RODRIGUEZ
JAIR PADILLA ROMERO**

**CORPORACION EDUCATIVA MAYOR DEL DESARROLLO
SIMON BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
AREA DE INVESTIGACION FORMTIVA IV
BARRANQUILLA**

2005

INDICE

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCION DEL PROBLEMA

1.2 FORMULACION DEL PROBLEMA

2. OBJETIVOS

2.1 OBJETIVO GENERAL

2.2 OBJETIVOS ESPECÍFICOS

3. JUSTIFICACIÓN DEL PROYECTO

4. MARCO DE REFERENCIA

4.1 MARCO TEÓRICO

4.2 MARCO CONCEPTUAL

4.3 MARCO ESPACIAL

4.4 MARCO LEGAL

5. METODOLOGÍA

6. RECURSOS

6.1 RECURSOS HUMANOS

6.2 RECURSOS INSTITUCIONALES



6.4 RECURSOS FINANCIEROS

7. CRONOGRAMAS

8. INGENIERÍA DE REQUISITOS

8.1 DESCRIPCIÓN DEL SISTEMA ACTUAL

8.2 DIAGRAMA DE FLUJO DEL SISTEMA ACTUAL

8.3 IDENTIFICACIÓN DE REQUISITOS

8.4 ANÁLISIS DE REQUISITOS

8.5 ESPECIFICACIÓN DE REQUISITOS

9. INGENIERÍA DE LA INFORMACIÓN

9.1 MISIÓN

9.2 VISIÓN

9.3 POLÍTICAS

10. ANÁLISIS DEL SISTEMA

10.1 ESPECIFICACIÓN DE ENTIDADES

10.2 DIAGRAMA DE FLUJO DE DATOS

10.2.1 DIAGRAMA DE CONTEXTO

10.2.2 DIAGRAMA DE NIVEL 1

10.2.2 DIAGRAMA DE NIVEL 2

10.2.3 DIAGRAMA DE NIVEL 3

10.3 MODELO ENTIDAD-RELACIÓN

10.4 MODELO RELACIONAL

10.5 DICCIONARIO DE DATOS

11. DISEÑO DEL SISTEMA

11.1 ESTRUCTURA FUNCIONAL

**11.2 DESCRIPCIÓN DE LOS COMPONENTES DE LA
ESTRUCTURA FUNCIONAL**

11.3 DISEÑO DE LAS INTERFACES

12. RECOMENDACIONES

CONCLUSIÓN

BIBLIOGRAFÍA

ANEXOS

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCION DEL PROBLEMA

En la actualidad, el manejo de la información es parte fundamental de cualquier empresa u organismo social sin importar tener un objetivo o no, para la realización de sus actividades en forma rápida y eficiente. Con los adelantos tecnológicos en el área computacional, área de comunicaciones y tecnologías de información las empresas, han dado suma importancia al uso de sistemas de información basados en computadora o sistemas de información computacionales, aprovechando los beneficios que estos les otorgan en el procesamiento de la información en forma rápida y confiable.

El colegio Distrital San Salvador, ubicado en la carrera 80 con calle 82 cuenta con los departamentos de administración y procesos académicos, los cuales se encargan de llevar la información del estudiante.

El departamento de administración se encarga de llevar el control de matriculas que se realiza cada año a través de un formato de datos establecido que contiene: datos personales del estudiante,

instituciones donde ha cursado estudios anteriores, cancelación o pago de ayuda institucional que se puede realizar por cuotas.

El departamento de procesos académicos lleva registros de notas de cada estudiante que son recopiladas por la evaluación de cada asignatura respecto a un banco de logros, además este departamento (procesos académicos) se encarga de la elaboración de certificados de estudios de estudiantes.

Este sistema tiene deficiencias en cuanto a la administración de la información, por que se maneja en forma manual. Los datos del estudiante son almacenados en fólder y fichas carpetas donde su conservación no es perdurable, y además permanecen en lockers, que hacen demasiado difícil y tardía la consulta, actualización y adición de información debido al material del cual están elaboradas las fichas carpetas y los fólder.

Los departamentos han sufrido pérdida de datos de gran importancia, obligando a la institución a invertir en nuevos materiales, y gastar tiempo en recopilar esa información para reinsertarla nuevamente.

Todas estas dificultades han generado una desorganización indeseable en el plantel, además se pierde espacio al utilizar contenedores (lockers) para el almacenamiento de ésta información. Esto ha llevado a la institución a perder información

de gran importancia, los resultados en el manejo de la información no son óptimos y se incrementara la perdida de datos.

1.2 FORMULACIÓN DEL PROBLEMA

¿Cómo se podría mejorar la gestión y organización de la información en los departamentos de Matriculas y Procesos Académicos en el Colegio Distrital San Salvador?



2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar y desarrollar un sistema de información que administre y gestione la información en los departamentos de Matriculas y Procesos Académicos en el Colegio Distrital San Salvador.

2.2 OBJETIVOS ESPECIFICOS

- Generar un modulo de captura de datos básicos, para poder llevar a cabo los procesos del área académica-administrativa.
 - Generar un modulo de captura de datos de los estudiantes, para llevar a cabo las matriculas correspondientes.
 - Generar un modulo de captura de notas de los estudiantes, para llevar a cabo los informes académicos.
 - Elaborar un modulo para la captura de información de los estudiantes que permita a la institución tener una hoja de vida de estos.
 - Diseñar un modulo de sistema para la importación y exportación de datos.
-

- Implementar un modulo generador de reportes para realizar consultas estáticas y dinámicas del sistema.
 - Diseñar un modulo de auditoria para saber quien tiene acceso al sistema.
 - Generar ayudas del sistema en línea para asesorar al usuario en el momento que se presente algún proceso que este desconozca.
-

3. JUSTIFICACION

El sistema de información a desarrollar permitirá una mejor administración y organización de la información, en los departamentos de matriculas y procesos académicos, debido a que este sistema agilizará las operaciones de adición, consulta y actualización de datos, con el fin de que se realicen en el menor tiempo posible.

De igual manera evitara que la institución genere gastos, en cuanto a materiales como fichas, fólderés e informes académicos.

Este trabajo será de gran utilidad a la institución y al personal encargado de llevar esta información, ya que les permitirá interactuar y conocer un conjunto de herramientas para las aplicaciones que se necesiten y al igual será un gran aporte a la institución en el campo de la tecnología de la información.

4. MARCO DE REFERENCIA

4.1 MARCO TEORICO

En la actualidad, en esta época en que estamos viviendo llegamos a asegurar que la información es igual de valiosa que los recursos financieros, recursos humanos, recursos materiales etc. con los que cuenta una empresa para existir en el mercado ya que la información es la base de las operaciones y actividades de la empresa.

La teoría de la organización y la práctica administrativa han experimentado cambios sustanciales en años recientes. La información proporcionada por las ciencias de la administración y la conducta ha enriquecido a la teoría tradicional. Estos esfuerzos de investigación y de conceptualización a veces han llevado a descubrimientos divergentes. Sin embargo, surgió un enfoque que puede servir como base para lograrla convergencia, el enfoque de sistemas, que facilita la unificación de muchos campos del conocimiento. Dicho enfoque ha sido usado por las ciencias

físicas, biológicas y sociales, como marco de referencia para la integración de la teoría organizacional moderna.

El primer expositor de la Teoría General de los Sistemas fue Ludwing von Bertalanffy, en el intento de lograr una metodología integradora para el tratamiento de problemas científicos.

La meta de la Teoría General de los Sistemas no es buscar analogías entre las ciencias, sino tratar de evitar la superficialidad científica que ha estancado a las ciencias. Para ello emplea como instrumento, modelos utilizables y transferibles entre varios continentes científicos, toda vez que dicha extrapolación sea posible e integrable a las respectivas disciplinas.

En la Corporación Educativa Mayor del Desarrollo Simón Bolívar se llevo a cabo un proyecto de investigación que trata sobre un “Sistema de Información Académico-Administrativo” que fue entregado a la “Institución de Educación Básica Primaria No 5” del municipio de Soledad, Atlántico.

Actualmente en la misma corporación se están llevando proyectos de investigación sobre sistemas de información académicos. Uno de estos proyectos es el de diseño e implementación de un “Sistema de Información Académico-Administrativo para el Colegio Ana Bolívar de Consuegra y José Consuegra Higgins en

el corregimiento de Isabel López, Atlántico. En este proyecto están trabajando los estudiantes Juan Carlos Casas, Ricardo Cortes y Osvaldo Villalba.

De igual manera se esta llevando a cabo un proyecto de investigación sobre un “Sistema de Información para el Centro Educativo Técnico Comercial Ajaib Singh”. Este software se esta desarrollando bajo la herramienta Visual Basic. El grupo de trabajo esta conformado por los estudiantes Saleh Acosta, Leonid Almanza, Ramses Escorcía y Deivis Ojeda.

4.2 MARCO CONCEPTUAL

En esta fase se establecerán conceptos claves que son de gran importancia para el desarrollo de las labores que ejecute el usuario con el sistema de información que se va a desarrollar. Tales conceptos se presentan a continuación:

Administración: Es el proceso de organizar, planear, dirigir y controlar actividades y recursos con el fin de lograr un objetivo común.

✓ **Aplicación:** Programa o conjuntos de programas diseñados para la realización de una tarea concreta, como puede ser una aplicación comercial, contable, etc.

Base de Datos: Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos. Dependiendo de cómo se vinculen dan lugar a B.D. jerárquicas, relacionales, etc. Un caso especial de éstas son las documentales, que, como su nombre indica, están diseñadas para almacenar volúmenes grandes de documentos, lo que genera una problemática distinta por los sistemas de búsqueda.

• **Información:** La información es todos aquellos datos transformados o modificados que tienen valor para aquellos usuarios que hacen uso de ellos.

"Los datos están constituidos por los registros de los hechos, acontecimientos, transacciones, etc. Por el contrario, la información implica que los datos estén procesados de tal manera que resulten útiles o significativos para el receptor de los mismos, por lo que en cierto modo, los datos se pueden considerar la materia prima para obtener información" llegamos a la conclusión que la información son datos procesados con un valor para aquel usuario que la necesita, aunque para otro solo sean datos que después de procesarlos obtenga otra información.



Logros: El MEN (Ministerio de Educación Nacional) los definió como los avances que se consideran deseables, necesarios y valiosos en los procesos de desarrollo del estudiante. Comprende los conocimientos, habilidades, los comportamientos, las actitudes y demás capacidades que deben alcanzar los alumnos de un nivel o grado en un área determinada dentro de su proceso de formación. (Generalmente se enuncian con un verbo conjugado en tercera persona del singular).

Modulo: Es una interfaz sobre la cual se van a realizar capturas de datos.

Procesos Académicos: Es el área encargada de llevar registros de notas y evaluaciones de estudiantes.

Reportes: Es un documento que muestra en forma detallada información en una fecha determinada

Sistema: Conjunto de componentes que interactúan entre si para lograr un objetivo.

Sistema de información: "Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) desempeñar su actividad de acuerdo a su estrategia de negocio".

Otra definición de sistemas de información es "Son procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas".

Por lo tanto podemos definir un sistema de información como un conjunto de subsistemas que incluyen hardware, software, medios de almacenamiento de datos ya sea primarios, secundarios y bases de datos relacionadas entre si con el fin de procesar entradas para realizar transformaciones a esas entradas y convertirlas en salidas de información importantes en la toma de decisiones.

El objetivo de un sistema de información es ayudar al desempeño de las actividades que desarrolla la empresa, suministrando la información adecuada, con la calidad requerida, a la persona o departamento que lo solicita, en el momento y lugar especificados con el formato más útil para el receptor.

4.3 MARCO ESPACIAL

El sistema de información a instalar en el Colegio Distrital San Salvador estará funcionando en las instalaciones de las oficinas principales, como lo son en este caso la oficina de Matriculas en donde la secretaria ejecuta sus labores administrativas, y en la oficina de Procesos Académicos, que en este caso se encuentran los coordinadores, licenciados y digitadotas de notas, que son las

encargadas de llevar el control de los procesos evaluativos y descriptivos que se realizan en el colegio.

4.4 MARCO LEGAL

MINISTERIO DE EDUCACIÓN NACIONAL
DECRETO NÚMERO 230 DE 2002
11 DE FEBRERO DE 2002

Por el cual se dictan normas en materia de currículo, evaluación y promoción de los educandos y evaluación institucional.

EL PRESIDENTE DE LA REPÚBLICA DE COLOMBIA

En ejercicio de sus atribuciones constitucionales y legales, especialmente de las que le confiere el numeral 11 del artículo 189 de la Constitución Política y en desarrollo de lo dispuesto en el artículo 79, en el literal d) del numeral 2 del artículo 148 de la **Ley** 115 de 1994 y en el numeral 5.5 del artículo 5° de la **Ley** 715 de 2001, y

)

CONSIDERANDO:

Que el artículo 79 de la **Ley** 115 de 1994 ordena que los establecimientos educativos al definir su plan de estudios, deben establecer entre otros aspectos los criterios de evaluación del educando; Que según lo establecido en el artículo 148 de la **Ley** 115 de 1994, el **Ministerio de Educación Nacional**, en cuanto al servicio público educativo, tiene entre las funciones de inspección y vigilancia la de fijar los criterios para evaluar el rendimiento escolar de los educandos y para su promoción a niveles superiores; Que de acuerdo con lo dispuesto por los numerales 5.5 y 5.6 del artículo 5° de la **Ley** 715 de 2001, le corresponde a la Nación establecer las normas técnicas curriculares y pedagógicas para los niveles de la **educación** preescolar, básica y media, sin perjuicio de la autonomía escolar que tienen los establecimientos educativos y de la especificidad de tipo regional y definir, diseñar y establecer instrumentos y mecanismos para el mejoramiento de la calidad de la **educación**,

DECRETA:

Artículo 1°. *Ámbito de aplicación.* Las normas reglamentarias contenidas en el presente decreto, se aplican al servicio público de

la **educación** formal que presten los establecimientos educativos del Estado, los privados, los de carácter comunitario, solidario, cooperativo o sin ánimo de lucro. Su interpretación debe favorecer la calidad, continuidad y universalidad del servicio público de la **educación**, así como el desarrollo del proceso de formación de los educandos. Se exceptúan para efectos de la promoción las modalidades de atención educativa a poblaciones, consagradas en el Título III de la **Ley 115** de 1994 y los educandos que cursen los programas de postprimaria rural, tele secundaria, escuela nueva, aceleración del aprendizaje y otras modalidades que determinará el Ministerio de **Educación** Nacional. La promoción de estas poblaciones será objeto de reglamentación especial por parte del Ministerio de **Educación** Nacional.

CAPITULO I

Normas técnicas curriculares

Artículo 2°. *Orientaciones para la elaboración del currículo.* El currículo es el conjunto de criterios, planes de estudio,

metodologías y procesos que contribuyen a la formación integral y a la construcción de la identidad cultural nacional, regional y local, incluyendo también los recursos humanos, académicos y físicos para poner en práctica las políticas y llevar a cabo el proyecto educativo institucional.

En virtud de la autonomía escolar ordenada por el artículo 77 de la **Ley** 115 de 1994, los establecimientos educativos que ofrezcan la **educación** formal gozan de autonomía para organizar las áreas obligatorias y fundamentales definidas para cada nivel, introducir asignaturas optativas dentro de las áreas establecidas en la **ley**, adaptar algunas áreas a las necesidades y características regionales, adoptar métodos de enseñanza y organizar actividades formativas, culturales y deportivas, dentro de los lineamientos que establezca el Ministerio de **Educación** Nacional. Por lo tanto, el currículo adoptado por cada establecimiento educativo debe tener en cuenta y ajustarse a los siguientes parámetros:

- a) Los fines de la **educación** y los objetivos de cada nivel y ciclo definidos por la **Ley** 115 de 1994;
 - b) Las normas técnicas, tales como estándares para el currículo en las áreas obligatorias y fundamentales del conocimiento, u otros instrumentos para la calidad, que defina y adopte el Ministerio de **Educación** Nacional;
 - c) Los lineamientos curriculares expedidos por el Ministerio de **Educación** Nacional.
-

Artículo 3°. *Plan de estudios*. El plan de estudios es el esquema estructurado de las áreas obligatorias y fundamentales y de áreas optativas con sus respectivas asignaturas que forman parte del currículo de los establecimientos educativos. El plan de estudios debe contener al menos los siguientes aspectos:

- a) La intención e identificación de los contenidos, temas y problemas de cada área, señalando las correspondientes actividades pedagógicas;
 - b) La distribución del tiempo y las secuencias del proceso educativo, señalando en qué grado y período lectivo se ejecutarán las diferentes actividades;
 - c) Los logros, competencias y conocimientos que los educandos deben alcanzar y adquirir al finalizar cada uno de los períodos del año escolar, en cada área y grado, según hayan sido definidos en el Proyecto Educativo Institucional, PEI, en el marco de las normas técnicas curriculares que expida el Ministerio de **Educación** Nacional. Igualmente incluirá los criterios y procedimientos para evaluar el aprendizaje, el rendimiento y el desarrollo de capacidades de los educandos;
 - d) El diseño general de planes especiales de apoyo para estudiantes con dificultades en su proceso de aprendizaje;
 - e) La metodología aplicable a cada una de las áreas, señalando el uso del material didáctico, textos escolares, laboratorios, ayudas audiovisuales, informática educativa o cualquier otro medio que oriente o soporte la acción pedagógica;
-

f) Indicadores de desempeño y metas de calidad que permitan llevar a cabo la auto-evaluación institucional.

CAPITULO II

Evaluación y promoción de los educandos

Artículo 4°. *Evaluación de los educandos.* La evaluación de los educandos será continua e integral, y se hará con referencia a cuatro períodos de igual duración en los que se dividirá el año escolar. Los principales objetivos de la evaluación son:

- a) Valorar el alcance y la obtención de logros, competencias y conocimientos por parte de los educandos;
 - b) Determinar la promoción o no de los educandos en cada grado de la **educación** básica y media;
 - c) Diseñar e implementar estrategias para apoyar a los educandos que tengan dificultades en sus estudios, y
 - d) Suministrar información que contribuya a la auto evaluación académica de la institución y a la actualización permanente de su plan de estudios.
-

Artículo 5°. *Informes de evaluación.* Al finalizar cada uno de los cuatro períodos del año escolar, los padres de familia o acudientes recibirán un informe escrito de evaluación en el que se dé cuenta de los avances de los educandos en el proceso formativo en cada una de las áreas. Este deberá incluir información detallada acerca de las fortalezas y dificultades que haya presentado el educando en cualquiera de las áreas, y establecerá recomendaciones y estrategias para mejorar. Además al finalizar el año escolar se les entregará a los padres de familia o acudientes un informe final, el cual incluirá una evaluación integral del rendimiento del educando para cada área durante todo el año. Esta evaluación tendrá que tener en cuenta el cumplimiento por parte del educando de los compromisos que haya adquirido para superar las dificultades detectadas en períodos anteriores. Los cuatro informes y el informe final de evaluación mostrarán para cada área el rendimiento de los educandos, mediante una escala dada en los siguientes términos:

Excelente

Sobresaliente

Aceptable

Insuficiente

Deficiente.

Parágrafo. Cada establecimiento educativo fijará y comunicará de antemano a los educandos, docentes y padres de familia o

acudientes la definición institucional de estos términos de acuerdo con las metas de calidad establecidas en su plan de estudios.

Artículo 6°. *Entrega de informes de evaluación.* Los informes de evaluación se entregarán a los padres de familia o acudientes en reuniones programadas preferencialmente en días y horas que no afecten su jornada laboral. La inasistencia de los padres de familia o acudientes a estas reuniones no puede acarrear perjuicios académicos a los educandos. El rector, director o coordinador, está en la obligación de programar y atender las citas que los padres de familia soliciten para tratar temas relacionados con la **educación** de sus hijos, en particular para aclaraciones sobre los informes de evaluación

Parágrafo. El establecimiento educativo no podrá retener los informes de evaluación de los educandos, salvo en los casos del no pago oportuno de los costos educativos siempre y cuando el padre de familia no demuestre el hecho sobreviniente que le impide el cumplimiento de las obligaciones contraídas con la institución en el momento de la matrícula.

Artículo 7°. *Registro escolar.* En todas las instituciones educativas se mantendrá actualizado un registro escolar que



contenga para cada alumno, además de los datos de identificación personal, el informe final de evaluación de cada grado que haya cursado en la institución.

Artículo 8°. *Comisiones de evaluación y promoción.* El Consejo Académico conformará, para cada grado, una Comisión de evaluación y promoción integrada por un número de hasta tres docentes, un representante de los padres de familia que no sea docente de la institución y el rector o su delegado, quien la convocará y la presidirá, con el fin de definir la promoción de los educandos y hacer recomendaciones de actividades de refuerzo y superación para estudiantes que presenten dificultades. En la reunión que tendrá la Comisión de evaluación y promoción al finalizar cada período escolar, se analizarán los casos de educandos con evaluación Insuficiente o Deficiente en cualquiera de las áreas y se harán recomendaciones generales o particulares a los profesores, o a otras instancias del establecimiento educativo, en términos de actividades de refuerzo y superación. Analizadas las condiciones de los educandos, se convocará a los padres de familia o acudientes, al educando y al educador respectivo con el fin de presentarles un informe junto con el plan de refuerzo, y acordar los compromisos por parte de los involucrados. Las Comisiones, además, analizarán los casos de los educandos con desempeños excepcionalmente altos con el fin de recomendar actividades especiales de motivación, o promoción anticipada.

Igualmente se establecerá si educadores y educandos siguieron las recomendaciones y cumplieron los compromisos del período anterior. Las decisiones, observaciones y recomendaciones de cada Comisión se consignarán en actas y éstas constituirán evidencia para posteriores decisiones acerca de la promoción de educandos.

Artículo 9°. *Promoción de los educandos.* Los establecimientos educativos tienen que garantizar un mínimo de promoción del 95% de los educandos que finalicen el año escolar en cada uno de sus grados. Al finalizar el año, la Comisión de evaluación y promoción de cada grado será la encargada de determinar cuáles educandos deberán repetir un grado determinado. Se considerarán para la repetición de un grado cualquiera de los siguientes educandos:

- a) Educandos con valoración final Insuficiente o Deficiente en tres o más áreas;
 - b) Educandos que hayan obtenido valoración final insuficiente o deficiente en matemáticas y lenguaje durante dos o más grados consecutivos de la **Educación** Básica;
 - c) Educandos que hayan dejado de asistir injustificadamente a más del 25% de las actividades académicas durante el año escolar.
- Es responsabilidad de la Comisión de evaluación y promoción estudiar el caso de cada uno de los educandos considerados para
-

la repetición de un grado y decidir acerca de evaluación y promoción de los educandos y evaluación institucional. Esta, pero en ningún caso excediendo el límite del 5% del número de educandos que finalicen el año escolar en cada grado. Los demás educandos serán promovidos al siguiente grado, pero sus evaluaciones finales no se podrán modificar.

Parágrafo. Si al aplicar el porcentaje mínimo de promoción, es decir el noventa y cinco por ciento, al número de alumnos de un grado y la operación da como resultado un número fraccionario, se tendrá como mínimo de promoción el número entero de educandos anterior a la fracción.

Artículo 10. *Recuperaciones*. Todo educando que haya obtenido insuficiente o deficiente en la evaluación final de una o más áreas presentará una nueva evaluación de esas áreas a más tardar la semana anterior al comienzo del siguiente año escolar. Esta evaluación se basará en un programa de refuerzo pertinente con las dificultades que presentó el educando y que el profesor del área le entregará al finalizar el año escolar. La evaluación se calificará de acuerdo con los términos del artículo 5° de este Decreto y su resultado, ya sea éste aprobatorio o no, deberá quedar consignado en el registro escolar del educando.

Artículo 11. *Educandos no promovidos.* El establecimiento educativo deberá diseñar programas específicos para educandos no promovidos al grado siguiente. En todos los casos, hará un seguimiento del educando y favorecerá su promoción durante el grado, en la medida en que éste demuestre la superación de las insuficiencias académicas que no aconsejaron su promoción.

CAPITULO III

Evaluación académica de las instituciones

Artículo 12. *Evaluación académica institucional.* La evaluación académica institucional, ya sea ésta auto evaluación o evaluación externa, es el proceso mediante el cual la institución educativa establece si ha alcanzado los objetivos y las metas de calidad académica propuestas en su Proyecto Educativo Institucional, PEI, y en su plan de estudios, y propone correctivos y planes de mejoramiento.

Artículo 13. *Auto evaluación académica institucional.* La evaluación institucional anual que debe llevarse a cabo en cada

una de las instituciones educativas según lo dispuesto en el artículo 84 de la **Ley** 115 de 1994 tiene por objeto mejorar la calidad de la **educación** que se imparte y por lo tanto, debe tomar en cuenta las metas de calidad académica propuestas cada año en el plan de estudios y formular recomendaciones precisas para alcanzar y superar dichas metas.

Artículo 14. *Evaluaciones académicas externas.* Las entidades territoriales periódicamente podrán contratar con entidades avaladas por el Ministerio de **Educación** Nacional evaluaciones académicas censales de los establecimientos educativos a su cargo. Dichas evaluaciones tendrán como referencia las normas técnicas que formule el Ministerio de **Educación** Nacional y se llevarán a cabo de acuerdo con la reglamentación que al respecto éste expida. Sus resultados deberán ser analizados tanto por las entidades territoriales, como por los establecimientos individuales con el propósito de tomar las medidas de mejoramiento necesarias.

DERECHOS DE AUTOR EN COLOMBIA

1. ¿PROPIEDAD INTELECTUAL?

La Propiedad Intelectual es una disciplina normativa que protege las creaciones intelectuales provenientes de un esfuerzo, trabajo o

destreza humanos, dignos de reconocimiento jurídico. La Propiedad Intelectual comprende:

- El derecho de autor y los derechos conexos;
- La propiedad industrial (que comprende la protección de los signos distintivos, de las nuevas creaciones, los circuitos integrados, los secretos industriales); y
- Las nuevas variedades vegetales.

2. ¿QUÉ ES EL DERECHO DE AUTOR?

Es la protección que le otorga el Estado al creador de las obras literarias o artísticas desde el momento de su creación y por un tiempo determinado.

3. ¿QUÉ SON LOS DERECHOS CONEXOS?

Con esta expresión se conocen en su conjunto, los derechos de los artistas, intérpretes o ejecutantes, productores de fonogramas y organismos de radiodifusión, en relación con sus interpretaciones o ejecuciones, fonogramas y emisiones de radiodifusión, respectivamente.

4. ¿QUÉ NORMAS REGULAN EL DERECHO DE AUTOR EN COLOMBIA?

- * Artículo 61 de la Constitución Política de Colombia;
 - * Decisión Andina 351 de 1993;
 - * Código Civil, Artículo 671;
 - * Ley 23 de 1982;
 - * Ley 44 de 1993;
-

- * Ley 599 de 2000 (Código Penal Colombiano), Título VIII;
- * Ley 603 de 2000
- * Decreto 1360 de 1989;
- * Decreto 460 de 1995;
- * Decreto 162 de 1996.

5. ¿CUÁLES SON LOS CONVENIOS INTERNACIONALES SOBRE DERECHO DE AUTOR Y DERECHOS CONEXOS, RATIFICADOS POR COLOMBIA?

- * Acuerdo de Caracas de 1911 sobre Derechos de Autor entre Colombia, Bolivia, Ecuador, Perú y Venezuela, al cual adhirió Colombia mediante la Ley 65 de 1913.
 - * Convención sobre Propiedad Literaria y Artística (IV Conferencia Internacional Americana, Buenos Aires, 1910), a la cual adhirió Colombia mediante la Ley 7 de 1936.
 - * Convención Interamericana sobre Derechos de Autor en Obras Literarias, Científicas y Artísticas, firmado en Washington en 1946, al cual adhirió Colombia mediante la Ley 6 de 1970.
 - * Convención Universal sobre el Derecho de Autor, firmada en Ginebra en 1952 y revisada en París en 1971, al cual adhirió Colombia por medio de la Ley 48 de 1975.
 - * Convención Internacional para la protección de los Artistas Intérpretes o Ejecutantes, los Productores de Fonogramas y los Organismos de Radiodifusión, firmada en Roma en 1961, a la cual adhirió Colombia a través de la Ley 48 de 1975.
-

* Convenio que establece la Organización Mundial de Propiedad Intelectual (OMPI) suscrito en Estocolmo en 1967, al cual adhirió Colombia mediante la Ley 46 de 1979.

* Convenio de Berna para la Protección de las Obras Literarias y Artísticas de 1886, cuya última modificación se firmó en París en 1971, al cual adhirió Colombia a través de la Ley 33 de 1987.

* Convenio para la Protección de los Productores de Fonogramas contra la Reproducción no Autorizada de sus Fonogramas, suscrito en Ginebra en 1971, al cual adhirió Colombia mediante la Ley 23 de 1992.

* Tratado Internacional para el Registro de las Obras Audiovisuales, suscrito en Ginebra en 1989, al cual adhirió Colombia a través de la Ley 26 de 1992.

* Acuerdo por el que se establece la Organización Mundial de Comercio (OMC), que contiene el Acuerdo sobre los aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (ADPIC), al cual adhirió Colombia mediante la Ley 170 de 1994.

* Tratado de Libre Comercio, suscrito entre Colombia, México y Venezuela (G-3), al cual adhirió Colombia en virtud de la Ley 172 de 1994.

* Convenio de Integración Cinematográfica Iberoamericana, al cual adhirió Colombia mediante Decreto 1448 de 1995.



* Tratado OMPI sobre Interpretación o Ejecución y Fonogramas, suscrito en Ginebra en 1996, al cual adhirió Colombia mediante la Ley 545 de 1999.

o Tratado OMPI sobre Derecho de Autor, suscrito en Ginebra en 1996, al cual adhirió Colombia mediante la Ley 565 de 2000.

6. ¿CUÁLES SON LAS OBRAS PROTEGIDAS POR EL DERECHO DE AUTOR?

La protección del derecho de autor recae sobre todas las obras literarias y artísticas que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocer, y que incluye, entre otras, las siguientes:

1. Las obras expresadas por escrito, es decir, los libros, folletos y cualquier otro tipo de obra expresada mediante letras, signos o marcas convencionales;
 2. Las conferencias, alocuciones, sermones y otras obras de la misma naturaleza;
 3. Las composiciones musicales con letra o sin ella;
 4. Las obras dramáticas y dramático-musicales;
 5. las obras coreográficas y las pantomimas;
 6. Las obras cinematográficas y demás obras audiovisuales expresadas por cualquier procedimiento;
 7. Las obras de bellas artes, incluidos los dibujos, pinturas, esculturas, grabados y litografías;
 8. Las obras de arquitectura;
-

9. Las obras fotográficas y las expresadas por procedimiento análogo a la fotografía;
10. Las obras de arte aplicado;
11. Las ilustraciones, mapas, croquis, planos, bosquejos y las obras plásticas relativas a la geografía, la topografía, la arquitectura o las ciencias;
12. Los programas de ordenador;
13. Las antologías o compilaciones de obras diversas y las bases de datos, que por la selección o disposición de las materias constituyan creaciones personales.

La anterior enumeración es de carácter ejemplificativo, no taxativo. Esto significa que pueden existir otras obras protegidas, a pesar de no estar mencionadas en la ley, siempre y cuando cumplan los requisitos exigidos para ser obras, según su definición.

7. ¿QUÉ SE ENTIENDE POR AUTOR?

Debe entenderse por tal, la persona física (natural) que realiza la creación intelectual de carácter literario o artístico.

De esta definición debe colegirse, en primer término, que solo un ser humano puede tener la condición de autor.

En segundo lugar, la expresión "que realiza la creación intelectual", significa que para ser considerado autor o coautor de una obra, tal persona ha debido llevar a cabo, por si mismo, el

proceso mental que significa concebir y expresar una obra literaria o artística. Varias personas naturales tendrán la condición de coautores si respecto de cada una de ellas, se puede predicar el haber realizado la creación intelectual en los términos antes mencionados.

El mero aporte de ideas que sirven de antecedente para la creación de la obra, o la contribución puramente física o mecánica, no creativa, a la plasmación de la obra, no atribuyen la condición de autor a quien las realiza.

El autor es el titular originario de los derechos morales y patrimoniales reconocidos por la ley.

8. ¿QUÉ SON LOS DERECHOS MORALES?

Desde el momento mismo de la creación de la obra, se le reconocen a los autores dos clases de prerrogativas: los derechos morales y los derechos patrimoniales.

Los derechos morales son derechos personalísimos, a través de los cuales se busca salvaguardar el vínculo que se genera entre el autor y su obra, en tanto ésta constituye la expresión de su personalidad. En tal carácter, los derechos morales son inalienables, inembargables, intransferibles e irrenunciables.

En virtud de los derechos morales, el autor puede:

- * Conservar la obra inédita o divulgarla;
 - * Reivindicar la paternidad de la obra en cualquier momento;
 - * Oponerse a toda deformación, mutilación o modificación que atente contra el mérito de la obra o la reputación del autor;
-

- * Modificar la obra, antes o después de su publicación;
- * Retirar la obra del mercado, o suspender cualquier forma de utilización aunque ella hubiese sido previamente autorizada.

9. ¿QUÉ SON LOS DERECHOS PATRIMONIALES?

Son prerrogativas de naturaleza económico - patrimonial, con carácter exclusivo, que permiten a su titular controlar los distintos actos de explotación de que la obra puede ser objeto. Lo anterior implica que todo acto de explotación de la obra, amparado por un derecho patrimonial, deberá contar con la previa y expresa autorización del titular del derecho correspondiente, quien podrá señalar para tal efecto las condiciones onerosas o gratuitas que tenga a bien definir, en ejercicio de su autonomía privada.

En virtud de los derechos patrimoniales, el autor o la persona natural o jurídica a quien se le transfieran estos derechos, puede realizar, autorizar o prohibir:

- * La reproducción,
- * La comunicación pública,
- * La distribución pública de ejemplares;
- * La traducción, adaptación, arreglo u otra transformación de la obra;
- * La importación de ejemplares de su obra reproducidos sin su autorización.

A diferencia de los derechos morales, los derechos patrimoniales son en esencia transferibles y sometidos a un término de duración

de la protección que en Colombia, por regla general, es el de la vida del autor más ochenta años después de su muerte. Así mismo, los derechos patrimoniales pueden ser expropiados y están sujetos a licencias obligatorias y al régimen de las limitaciones o excepciones al derecho de autor consagradas por la Ley.

10. ¿COMO SE TRANSFIEREN LOS DERECHOS PATRIMONIALES DEL AUTOR?

Si bien los derechos morales son intransferibles, una persona natural o jurídica diferente al autor (titular originario) puede detentar la titularidad derivada de los derechos patrimoniales, a través de cualquiera de las siguientes modalidades de transmisión:

1. Transmisión por un acto entre vivos

Entre las modalidades de transferencia de los derechos patrimoniales, el acuerdo entre el autor y un tercero, resultante de la manifestación de la autonomía de la voluntad de aquellos, es una de las más importantes.

Dentro de las diferentes modalidades de acuerdos, es preciso resaltar dos de ellos: el contrato de cesión de derechos y el contrato de obra por encargo.

1.1. Contrato de cesión de derechos

La cesión es un contrato por medio del cual, el autor o titular de una obra, denominado cedente, transmite total o parcialmente sus derechos a otra persona, denominada cesionario, a cambio de una remuneración, o sin ella. Este contrato, regulado por el artículo

182 y siguientes de la Ley 23 de 1982, tiene como característica principal que el cedente se desprende de los derechos, convirtiendo al cesionario, por virtud de la transferencia, en el nuevo titular o titular derivado.

De acuerdo con el artículo 183 de la Ley 23 de 1982, toda enajenación y transferencia del derecho de autor, sea ésta total o parcial, deberá constar en escritura pública o en instrumento privado reconocido ante notario, de lo anterior se desprende que la cesión es cualificada y sólo se perfecciona con el cumplimiento de este requisito.

Ahora bien, si los contratantes desean hacer oponible a terceros estos contratos, deberán inscribirlos en el Registro Nacional de Derecho de Autor.

Al transferirse el derecho de autor mediante la cesión, el cesionario se transforma en titular del derecho, permitiéndole actuar en nombre propio, incluso en lo que respecta a entablar acciones judiciales contra los infractores. En el caso de que la cesión sea parcial, los autores conservarán las prerrogativas que no han transferido expresamente.

1.2 Contrato de obra por encargo.

En virtud de este contrato, regulado por el artículo 20 de la Ley 23 de 1982, se realiza un encargo a uno o varios autores para la elaboración de una obra, según plan señalado por persona natural o jurídica y por cuenta y riesgo de ésta, pactándose una determinada remuneración. En este tipo de contratos de servicios,

para que la titularidad de los derechos patrimoniales se radique en cabeza del contratante, es preciso que además de la responsabilidad del encargado exista un plan de éste al autor o autores en el que se describa sumariamente la estructura de la obra que se encarga. Por tratarse de un contrato de transferencia de derechos no escapa a los actos sometidos al cumplimiento de los requisitos establecidos en el artículo 183 de la Ley 23 de 1982.

2. Transmisión por ministerio de la ley

2. 1 Obras creadas por servidores públicos.

Por disposición legal (Art 91, Ley 23 de 1982) la titularidad de derechos patrimoniales de las obras creadas por servidores públicos en ejercicio de sus funciones legales y constitucionales se radica en cabeza de la entidad pública correspondiente, y el servidor público conservará los derechos morales, con el compromiso de no ejercerlos de una manera incompatible con los derechos y obligaciones de la entidad pública.

2. 2 Obras colectivas

Las obras colectivas son aquellas producidas por un grupo de autores, por iniciativa y bajo la orientación de una persona natural o jurídica que la coordina, divulga y publica bajo su nombre.

El artículo 92 de la Ley 23 de 1982, establece que el titular de los derechos de autor será el editor o persona jurídica o natural por cuya cuenta y riesgo se realizan los aportes de las personas naturales que contribuyen en las obras colectivas creadas dentro

de un contrato laboral o de arrendamiento de servicios, en las que sea imposible identificar el aporte individual de cada participante. Una mención especial requiere la obra audiovisual, pues es un ejemplo de obra colectiva, en la cual intervienen varias personas físicas, con aportes destinados a un fin común, los que son identificables, así como el autor de cada contribución.

El artículo 98 de la Ley 23 de 1982, regula específicamente el tema de la titularidad de los derechos respecto a este tipo de obras, precisando que las prerrogativas patrimoniales sobre la obra cinematográfica, salvo estipulación en contrario, se reconocerán a favor del productor cinematográfico.

3 Transmisión por causa de muerte

La transmisión de los derechos patrimoniales por causa de muerte del autor, ocurrirá por vía testamentaria o a título de sucesión intestada, de conformidad con las disposiciones legales sobre la materia, y no tendrá limitaciones, salvo que el autor en vida haya cedido a un tercero, total o parcialmente y por un lapso determinado o por toda la duración, el derecho patrimonial, o que la titularidad en cabeza del tercero surja por alguna otra causa legal.

11. ¿CÓMO SE APLICA LA PROTECCIÓN DEL DERECHO DE AUTOR Y LOS DERECHOS CONEXOS EN EL ENTORNO DIGITAL?

En primer término es necesario considerar que una obra literaria o artística se entiende protegida por el derecho de autor,

independientemente del soporte en que ella esté plasmada, o la forma en que ésta haya sido expresada. En otras palabras, las obras serán protegidas independientemente de si tienen un soporte material o no.

A partir de la declaración concertada respecto del artículo 1.4): del Tratado de la OMPI sobre Derecho de Autor (WCT) (1996), el derecho de reproducción tal como se establece en el artículo 9 del Convenio de Berna, y las excepciones permitidas en virtud del mismo, son totalmente aplicables en el entorno digital, en particular a la utilización de obras en forma digital.

Queda entendido que el almacenamiento en forma digital en un soporte electrónico de una obra protegida, constituye una reproducción en el sentido del artículo 9 del Convenio de Berna.

Por otra parte, en virtud de los tratados OMPI sobre Derecho de Autor y sobre Interpretación o Ejecución y Fonogramas (WPPT), se ha reconocido a los artistas interpretes o ejecutantes y productores de fonogramas, el derecho exclusivo de autorizar la puesta a disposición del público de sus obras, interpretaciones o ejecuciones o fonogramas, ya sea por hilo o por medios inalámbricos, de tal manera que los miembros del público puedan tener acceso a ellos desde el lugar y en el momento que cada uno de ellos elija, haciéndose con esto clara alusión a la comunicación de tales contenidos a través de las redes digitales interactivas, como el Internet.

El Tratado OMPI sobre Derecho de Autor ha sido aprobado en Colombia mediante la Ley 565 de 2000, y en el ámbito internacional ha entrado en vigencia el 6 de marzo de 2002.

El Tratado OMPI sobre Interpretación o Ejecución y Fonogramas ha sido aprobado en Colombia mediante la Ley 545 de 1999, y a escala internacional ha entrado en vigencia el 20 de mayo de 2002.

12. ¿EN QUE CONSISTE LA AUSENCIA DE FORMALIDADES?

La protección que brinda el derecho de autor a las obras literarias y artísticas, así como los derechos que se reconocen sobre las interpretaciones o ejecuciones artísticas, los fonogramas y las emisiones de radiodifusión, no están subordinados al cumplimiento de ninguna formalidad (v.gr. registro, depósito, mención de reserva del derecho, certificado notarial, pago de tasas, etc.). En consecuencia, la omisión del registro o de cualquiera de tales formalidades no impide el goce o el ejercicio del derecho de autor.

Este principio está desarrollado por los artículos 52 y 60 de la Decisión 351 de 1993, 9 de la Ley 23 de 1982, 5.2 del Convenio de Berna, 18-02 del Tratado de Libre Comercio entre Colombia, Méjico y Venezuela (Acuerdo G-3), III de la Convención Universal y 20 del Tratado OMPI sobre Interpretación o Ejecución y Fonogramas.

13. ¿QUÉ ES LA INSPECCION Y VIGILANCIA A LAS SOCIEDADES DE GESTION COLECTIVA DE DERECHO DE AUTOR Y DE DERECHOS CONEXOS?

Las Sociedades de Gestión Colectiva de derecho de autor o de derechos conexos, están sometidas a la inspección y vigilancia del Estado, por intermedio de la Dirección Nacional de Derecho de Autor. En cumplimiento de lo anterior esta entidad realiza las siguientes funciones:

- * Reconocimiento de personería jurídica y autorización de funcionamiento de las sociedades;
- * Control de legalidad de sus reformas estatutarias;
- * Registro de los convenios de representación recíproca con otras sociedades;
- * Registro de contratos generales celebrados con usuarios;
- * Inscripción de los miembros del Consejo Directivo, Comité de Vigilancia, Gerente, Secretario, Tesorero y del Fiscal;
- * Trámite y decisión de las impugnaciones contra los actos de elección realizados por las Asambleas Seccionales y los actos de administración del Consejo Directivo;
- * Realización de investigaciones con el fin de verificar el cumplimiento de las normas legales y estatutarias por parte de la sociedad.

14. ¿QUÉ SON LAS LIMITACIONES O EXCEPCIONES AL DERECHO DE AUTOR?

Son aquellos casos en que las obras protegidas por el derecho de autor, pueden ser objeto de determinados usos que no requieren la previa expresa autorización del autor o su derechohabiente, ni el pago de una remuneración.

De la misma forma en que la propiedad común se restringe para dar prevalencia al interés general sobre el interés particular (función social de la propiedad), el derecho de autor, como derecho individual, se ve limitado en aquellos casos descritos taxativamente por la ley.

Las limitaciones o excepciones al derecho de autor se encuentran establecidas de manera taxativa, circunscribiéndose a casos especiales que no atenten contra la normal explotación de las obras y no causen un perjuicio injustificado a los legítimos intereses del autor o titular de derechos.

15. ¿QUÉ ES LA DIRECCION NACIONAL DE DERECHO DE AUTOR?

Es una Unidad Administrativa Especial, con personería jurídica, autonomía administrativa y patrimonio independiente, adscrita al Ministerio del Interior y De Justicia.

A la Dirección Nacional de Derecho de Autor le compete el diseño, dirección, administración y ejecución de las políticas gubernamentales en materia de derechos de autor; llevar el registro nacional de las obras literarias y artísticas y ejercer la inspección y vigilancia sobre las sociedades de gestión colectiva de derecho de autor o derechos conexos.



En desarrollo de su misión, cuenta con un programa de capacitación al público interesado en la materia, así como con el servicio de un Centro de Documentación especializado, debidamente sistematizado, ubicado en sus instalaciones.

El ámbito de las funciones de la Dirección Nacional de Derecho de Autor comprende todo el territorio nacional, teniendo su domicilio principal en la ciudad de Bogotá.

Sus datos de contacto son:

Dirección Nacional de Derecho de Autor

Carrera 13 No. 27-00 piso 6

Teléfono: 3418177

Fax: 2860816

Sitio Web: www.derautor.gov.co

Correo electrónico: derautor@derautor.gov.co

16. ¿CUÁL ES LA MISIÓN INSTITUCIONAL DE LA DIRECCIÓN NACIONAL DE DERECHO DE AUTOR?

Fortalecer la debida y adecuada protección de los titulares de Derecho de Autor y los Derechos Conexos, contribuyendo al desarrollo de una cultura de respeto a estos derechos.

Inculcamos la noción del Derecho de Autor como un nuevo concepto de riqueza que genera empleo, impulsa el desarrollo de la Nación, protege la creación de las obras literarias y artísticas que contribuyen a engrandecer la cultura, el conocimiento, el arte, el entretenimiento y la calidad de vida, y que responde a los retos impuestos por los avances tecnológicos.

17. ¿QUÉ ACTIVIDADES DE CAPACITACIÓN DESARROLLA LA DIRECCION NACIONAL DE DERECHO DE AUTOR?

El programa de capacitación en derecho de autor a la población adulta, comprende las siguientes actividades:

- * Realización de conferencias en diversas entidades del sector público y del sector privado, comprendiendo industrias culturales, instituciones educativas, asociaciones de autores, artistas, profesionales y público en general.

- * Realización de seminarios especializados en donde se abordan diversos aspectos de la temática autoral, en coordinación con otras entidades organizadoras nacionales e internacionales.

- * Participación en ferias y exposiciones cuya temática tenga relación con el derecho de autor y los derechos conexos. Así mismo, un stand itinerante de la Dirección Nacional de Derecho de Autor visita periódicamente universidades e instituciones, con el fin de atender consultas y distribuir material de divulgación

- * Las personas interesadas en la realización de alguna de estas actividades pueden contactar a la Dirección Nacional de Derecho de Autor.

- * Para el desarrollo de la actividad en comento, contamos con un portafolio de conferencias de la Dirección Nacional de Derecho de Autor.

18. ¿QUÉ ES EL REGISTRO NACIONAL DE DERECHO DE AUTOR?

Es un servicio que presta el Estado a través de la Unidad Administrativa Especial Dirección Nacional de Derecho de Autor, para todo el territorio nacional. Su finalidad es brindarle a los titulares de derecho de autor y derechos conexos un medio de prueba y de publicidad a sus derechos, así como a los actos y contratos que transfieran o cambien ese dominio amparado por la ley. Igualmente, ofrece garantía de autenticidad y seguridad a los títulos de derecho de autor y de derechos conexos y a los actos y documentos que a ellos se refiere.

19. ¿CÓMO SE OBTIENE EL REGISTRO ANTE LA DIRECCION NACIONAL DE DERECHO DE AUTOR?

La inscripción en el Registro Nacional de Derecho de Autor se lleva a cabo en la Oficina de Registro de la Unidad Administrativa Especial Dirección Nacional de Derecho de Autor, ubicada en la Carrera 13 No. 27-00 Piso 6, en la ciudad de Bogotá.

Los pasos a seguir son los siguientes:

Obtenga el formulario de registro en las oficinas de la Unidad Administrativa Especial Dirección Nacional de Derecho de Autor o en nuestro sitio Web (www.derautor.gov.co). Este formulario no tiene ningún costo.

Diligencie el formulario que consta de dos hojas "Registro" y "Certificado".

Los datos allí requeridos deberán consignarse de idéntica manera con letra clara y legible, preferiblemente a máquina, sin

enmiendas ni correcciones, firmar ambos folios en original, adjunto al formulario debe remitirse un ejemplar de la obra a la Unidad Administrativa Especial Dirección Nacional de Derecho de Autor.

Si su obra ha sido editada (es decir puesta al alcance del público), deberá aportar un ejemplar, donde consigne en lugar visible el título de la obra, el nombre o seudónimo del autor o los autores, y del traductor (si fuere el caso), la mención de reserva de derecho de autor y del año de la primera publicación, precedida del símbolo (c), el nombre y la dirección del editor, así como del impresor.

Si su obra es inédita debe indicar en el ejemplar que aporta a la solicitud el título de la obra, el nombre del autor y año de creación.

20. ¿QUÉ VALOR PROBATORIO TIENE EL CERTIFICADO DE REGISTRO ANTE LA DIRECCION NACIONAL DE DERECHO DE AUTOR?

El registro del derecho de autor es declarativo y no constitutivo de derechos. Sin perjuicio de ellos, la inscripción en el registro presume ciertos los hechos y actos que en ella consten, salvo prueba en contrario.

De esta forma, sin descartar la validez de otros medios probatorios que se puedan presentar en el proceso para acreditar la autoría y titularidad sobre una obra, la parte que presente el

certificado del Registro Nacional de Derecho de Autor se beneficiará de una presunción que invierte en su favor la carga de la prueba, quedando en su contraparte la carga procesal de desvirtuar probatoriamente la información consignada en el registro.

21. ¿QUÉ ES EL TRATADO DE CARTAGENA Y QUÉ RELACIÓN TIENE CON LOS DERECHOS DE AUTOR?

Aun cuando muchos países protegen la propiedad intelectual, son pocas las leyes entre países que se han promulgado en esta área. Se han negociado varios tratados y convenciones multinacionales que obligan a los países firmantes a proporcionar una norma mínima de protección de la propiedad intelectual a los extranjeros.

Colombia suscribió el Acuerdo de Cartagena, el cual cobija a los países del Pacto Andino. Este acuerdo realiza un manejo de la prioridad similar al Acuerdo de Paris, pero sólo para los países del pacto Andino que lo han suscrito.

Este convenio confiere a los ciudadanos de cualquier país firmante la misma protección y los mismos beneficios sin discriminación.

Una de las estipulaciones más importantes tiene que ver con la "Prioridad de la Convención". La fecha de prioridad del solicitante es la fecha en la que inicialmente se presenta la solicitud en el país de origen del solicitante.

El convenio hace posible que el solicitante establezca esta fecha como la fecha efectiva para la aplicación de la patente, presentada en cualquier otro país firmante del convenio, siempre y cuando la solicitud se presente dentro de los 12 meses (en el caso de patentes o marcas de utilidad) o 6 meses (en el caso de diseño industrial o marcas registradas) siguientes a la fecha de prioridad. Esto elimina al propietario la necesidad y el costo de presentar su solicitud por separado en cada país, a la vez que se obtiene una protección eficaz. De esta manera, las solicitudes de terceros no pueden ser barreras a la solicitud del propietario, si el propietario presenta su solicitud dentro del período de tiempo que se prescribe en el convenio.

5. METODOLOGIA

El método de investigación utilizado para el análisis de la situación presente en la institución fue el método **inductivo**, porque se realizó un estudio de procesos llevado a cabo en cada departamento para establecer en conjunto cuáles son los problemas, las causas y las consecuencias que conducen al sistema actual a la ineficiencia.

El tipo de investigación bajo el cual se desarrolla este método con respecto a la observación y el análisis de la información suministrada por los departamentos y directivos del colegio es **descriptiva**, debido a la manera de cómo se interpretan y se describen los fenómenos. Los datos con los cuales se trabaja son difíciles de manejar, ya que no pueden brindar una respuesta cuantificable, sino cualitativa, que debe ser muy bien comprendida para buscarle una explicación y desarrollar de forma sencilla el problema.

Durante las visitas que se realizaron a la institución, se recibieron formatos y documentos por parte de cada uno de los departamentos. En estos formatos y documentos se encuentra toda la información referente al estudiante para llevar a cabo las labores de la institución.

La línea de investigación a la cual esta encaminado nuestro proyecto es “INGENIERIA DEL SOFTWARE”.

A la vez se consulto sobre las áreas de trabajo del colegio y otros aspectos relacionados a través de herramientas como el Internet, el PEI (Proyecto Educativo Institucional), libros, manuales, entre otros.

Además, se llevo a cabo una entrevista que tuvimos con la rectora y la coordinadora académica del colegio acerca de cómo trabaja el sistema actualmente y sobre el presente proyecto que estamos trabajando en la institución.

El Colegio Distrital San Salvador cuenta con una cobertura de 600 estudiantes en la ciudad de Barranquilla. De esta población general tomamos un grupo de 50 alumnos entre los cursos mayores (9°,10°,11°), y se les formulo el siguiente interrogante: ¿Esta de acuerdo que en la institución se implemente un sistema de información asistido por computador para la mejora de los procesos académicos y de matriculas? Los resultados obtenidos se reflejan en la siguiente tabla:

SI	NO	NO SABE
87%	5%	8%

Los resultados obtenidos fueron óptimos, debido a que el 87% estuvo de acuerdo con la implementación de este nuevo sistema.

6. RECURSOS

6.1. HUMANOS

Para desarrollar este proyecto, necesitamos de la colaboración de los ingenieros asesores de nuestro proyecto como lo son el Ing. Sergio Jiménez y el Ing. Juan Carlos Calabria. También contamos con el personal laboral de la institución, como lo son la coordinadora académica, la Dra., Ledys Hincapié y el Lic. Armando Ortega.

ANALISTAS DEL SISTEMA

Elías José Albor Arrieta

Luis Leonardo Barros Ortega

Marlon José Insignares Rodríguez

Jair Eduard padilla Romero

6.2. INSTITUCIONALES

Para el desarrollo de este propósito se utilizaron las salas de Informática de la Corporación Educativa Mayor Del Desarrollo Simón Bolívar.

6.3. MATERIALES

En los recursos materiales se utilizaron los siguientes implementos: los PC de las salas de informática de la universidad Simón Bolívar, implementos de papelería , medios de información magnéticos (CDs, Diskettes,)

6.4 FINANCIEROS

En los recursos financieros fueron los gastos de transporte hasta el Colegio Distrital San Salvador, gastos de investigación en Internet.

8. INGENIERIA DE REQUISITOS

8.1 Descripción del sistema actual

En el Colegio Distrital San Salvador se realizan matriculas anualmente, que se llevan acabo con la participación del estudiante y un acudiente que este debe presentar y la respectiva secretaria del departamento de matriculas. El acudiente puede ser el padre de familia o un particular autorizado por los padres, el cual debe llevar consigo los documentos requeridos para la matricula, si el alumno va a ingresar por primera vez a la institución. De otra manera el estudiante solo da su identificación y la secretaria acude a un archivador o locker, que almacena información de los estudiantes matriculados para extraer los datos necesarios para continuar con el proceso.

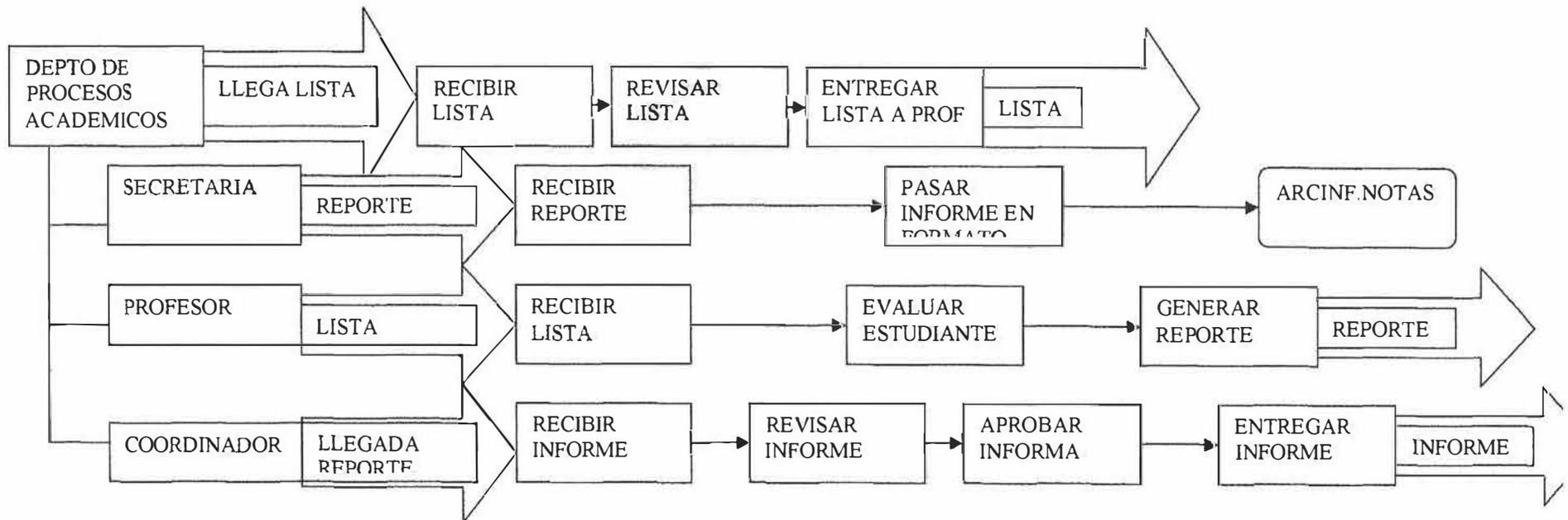
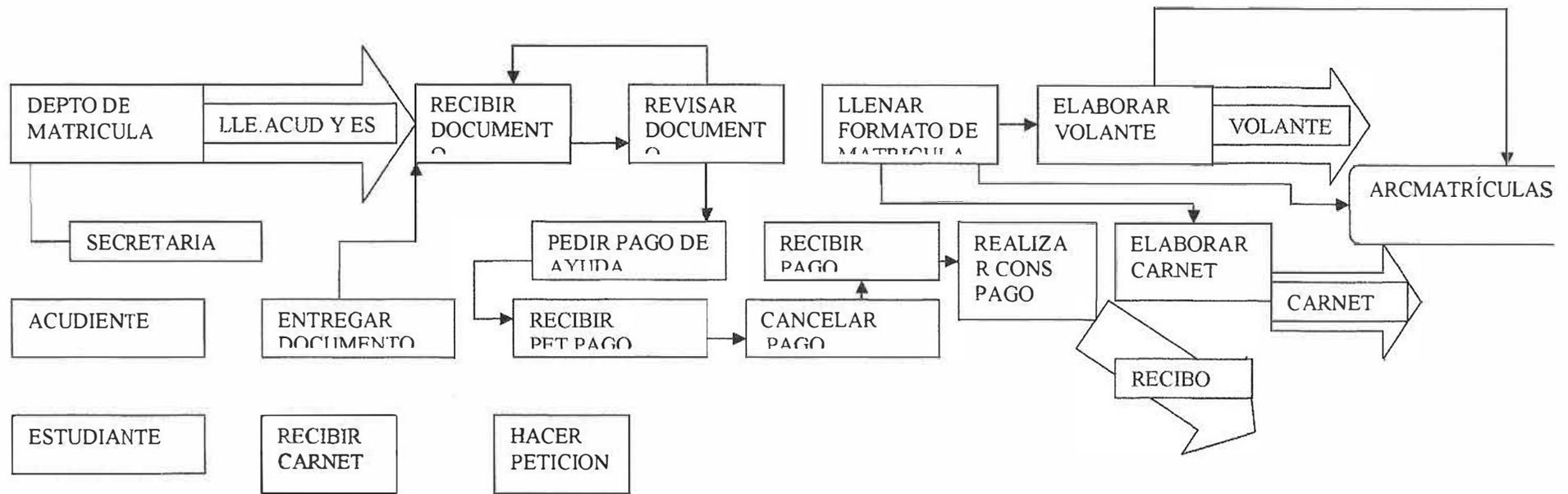
La secretaria llena el formato de matriculas con los requisitos y con información adicional del acudiente o padre y del estudiante. Una vez que el formato este listo, el acudiente debe cancelar un pago de ayuda institucional que depende del curso el cual el estudiante va hacer parte, y este pago se debe efectuar automáticamente. Una vez que el estudiante haya cancelado la

secretaria elabora un recibo de pago y se lo entrega a este. Posteriormente en el departamento se elabora el volante de matricula y el carnet estudiantil que se entrega al acudiente del estudiante respectivamente. El formato, el recibo y demás documento son guardados en una carpeta con el nombre de estudiante, y a su vez esta carpeta es almacenada en un locker que contiene todas las matriculas.

Una vez que el departamento haga el cierre de matriculas, elabora listas por curso de los estudiantes matriculados y las envía al departamento de procesos académicos.

El departamento de procesos académicos recibe la lista de los alumnos por curso enviados por el departamento de matriculas, son revisadas y hace un archivo por estudiante que se almacena en un locker que guardara información de las notas; y luego las listas son entregadas a los profesores. Estos reciben la lista, proceden a evaluar al estudiante con respecto a la asignatura que se esta desarrollando y genera un reporte de notas de cada estudiante que es entregado al coordinador, el cual es el encargado de revisarlo y realizar las operaciones correspondientes para luego enviarlo a la secretaria del departamento de procesos académicos, la cual pasa el informe en un formato elaborado a través de un procesador de texto en computador; este informe es almacenado en el locker de notas y también es enviado al coordinador para que lo revise, le

8.2 Diagrama de flujos del sistema actual (Modelador de Procesos)



de el visto bueno, firmarlo y finalmente entregarlo a los padres de familia.

Cuando el estudiante o el acudiente requieran una constancia de estudio o certificados de notas deben dirigirse al departamento de procesos académicos y llevar consigo el carnet estudiantil , hacer la petición a la secretaria, la cual se encarga de extraer la información necesaria para elaborar el documento. Ya elaborado el documento es llevado al coordinador para que este lo revise, lo apruebe, lo firme y por ultimo se entrega al solicitante.

8.3 Identificación de requisitos

Con base en la entrevista celebrada entre los directivos, coordinadores y desarrolladores, los formatos y documentos que fueron entregados, se acordó una lista de requisitos para el desarrollo del sistema de información. Este conjunto de requisitos se expresan en los siguientes ítems:

- Generar lista de estudiantes que se encuentren matriculados en un curso específico
 - Generar informes descriptivos-evaluativos por alumno que muestren el comportamiento académico en un periodo.
 - Generar registros de matriculas.
-

8.4 Análisis de requisitos

Los requisitos que se listaron anteriormente dan como resultado un software sobre el cual se maneje en forma integrada las áreas de matriculas y procesos académicos. El sistema debe contener las especificaciones necesarias para elaboración de informes y reportes sobre los cuales se resalten los aspectos descriptivos, valorativos y evaluativos del estudiante. El sistema a desarrollar tiene una viabilidad considerable, debido a que se pueden desarrollar todos sus requerimientos para una mejora en los procesos que se manejan en la institución.

8.5 Especificación de requisitos

El proyecto que se va a desarrollar para el Colegio Distrital San Salvador se basa en los siguientes requisitos:

- Modelar todos los aspectos evaluativos e intelectuales del estudiante e implementar una base de conocimientos con ellos para poder trabajar en las evaluaciones de los estudiantes.
 - Diseñar una interfaz comprensible para el usuario y dar una mejor comodidad y facilidad en el manejo de integración de las áreas.
-

- Generar informes y reportes dinámicos de estudiantes por curso que muestren el proceso descriptivo-evaluativo por periodos y año lectivo.
- Diseñar un modulo de importación y exportación de datos.
- Diseñar un modulo de auditoria para la seguridad de las transacciones que se realizan sobre el sistema.

Las anteriores son las características de sistemas de información a realizar en conformidad con las personas involucradas en el desarrollo del proyecto

9.INGENIERIA DE LA INFORMACION

9.1 FILOSOFÍA DE LA INSTITUCIÓN.

El COLEGIO DISTRITAL SAN SALVADOR, es una institución comprometida con los cambios sociales y los avances de la ciencia y la tecnología. Por esta razón y apoyados en una concepción humana, buscamos coadyuvar en la consecución de una sociedad justa, progresiva y equitativa que permita la formación integral del hombre, para ello contamos con un plan de estudio muy bien estructurado, en el cual priman la ciencia, la tecnología y los valores humanos.

Por todo lo anterior, nuestra labor pedagógica tiene como propósito central orientar a los jóvenes para que sean personas participativas, críticas, responsables, cuestionadores de la realidad que los circunda e investigadores del saber interdisciplinario que le ofrece el plan de estudios.

CONCEPTO ANTROPOLÓGICO.

El COLEGIO DISTRITAL SAN SALVADOR, concibe al hombre como un ser social, cultural, político, fundamenta su quehacer respetando la singularidad, su naturaleza espiritual, establece unas relaciones de carácter dialogal y participativo respetando la autonomía, derechos y deberes de los alumnos, propiciando una convivencia pacífica y el desarrollo de la persona para sea capaz de desarrollarse como persona que laborará en mejores condiciones que sus padres, esto, mostrando la realidad concreta despertando conciencia para el crecimiento y formación.

Acorde con ésta concepción, tendrá como finalidad la reconstrucción de un hombre y de una mujer nuevos capaces de autocrecer, auto desarrollarse, en procura de una mejor calidad de vida, por la paz y el desarrollo del país.

9.1 MISIÓN.

El COLEGIO DISTRITAL SAN SALVADOR, teniendo en cuenta la Constitución Nacional, la Ley General de Educación, Los Derechos del Niño, y los principios rectores del Plantel, se plantea la necesidad de reorientar su misión en el sentido de

formar hombres nuevos, participantes, autónomos, críticos y activos, capaces de desarrollar formas de convivencia social y de producir conocimientos que garanticen el respeto a la vida y a los deberes fundamentales, también la de ser un bachiller académico o en cualquier opción que la institución pueda ofrecer, con una formación integral capaz de transformarse y transformar la sociedad.

El COLEGIO DISTRITAL SAN SALVADOR, tiene su misión centrada en la búsqueda de la excelencia educativa a través de la utilización de un nuevo modelo pedagógico, del fomento de los valores humanos, sociales, culturales y cívicos y de la gestión de recursos financieros, materiales y humanos que satisfagan las necesidades detectadas.

9.2 VISIÓN.

Buscamos formar integralmente al individuo y capacitarlo para que se adapte a los nuevos avances de la ciencia y la tecnología y pueda desempeñarse en el campo laboral, especialmente en el industrial que es el que mayor demanda tiene en la zona de influencia en la cual se encuentra ubicada nuestra institución. Para ello diseñaremos y presentaremos convencidos y proyectos a las empresas públicas y privadas que se encuentran a nuestro

alrededor, esperando que ellas nos brinden el apoyo necesario para poder cristalizar nuestra visión.

FINES DE LA EDUCACIÓN.

Los fines educativos del COLEGIO DISTRITAL SAN SALVADOR, son de conformidad a lo establecido en el Artículo N° 67 de la Constitución Política de Colombia y señalados en el Título I y artículo 5° de la Ley general de Educación, y a continuación se detallan:

- 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 formación integral, física, psíquica, intelectual, moral, espiritual, social, afectiva, ética y demás valores humanos.

- La formación en el respeto a la vida y a los demás derechos humanos, a la paz, a los principios democráticos, de convivencia, pluralismo, justicia, solidaridad y equidad, a sí como el ejercicio de la tolerancia y de la libertad.



- La formación para facilitar la participación de todos en las decisiones que los afectan en la vida económica, política, administrativa y cultural de la nación.
 - La formación en el respeto a la comunidad legítima y a la Ley a la cultura nacional, a la historia colombiana y a los símbolos patrios.
 - La adquisición y generación de los conocimientos científicos y técnicos más avanzados, humanos, históricos, sociales, geográficos y estéticos, mediante la apropiación de hábitos intelectuales adecuados para el desarrollo del saber, el fomento de la investigación y el estímulo a la creación artística en sus diferentes manifestaciones.
 - La creación y fomento de una conciencia de la soberanía nacional y para la práctica de la solidaridad y la integración con el mundo, en especial con Latinoamérica y el Caribe.
 - El desarrollo de la capacidad crítica, reflexiva y analítica que fortalezca el avance científico y tecnológico nacional orientado con prioridad al mejoramiento cultural y de la calidad de la vida de la población, a la participación en la búsqueda de alternativas de solución a los problemas y al proceso social y económico del país.
-

- La adquisición de una conciencia para la conservación, protección y mejoramiento del medio ambiente, de la calidad de vida, del uso racional de los recursos naturales, de la prevención de desastres, dentro de una cultura ecológica y del riesgo y la defensa del patrimonio cultural de la nación.

 - La formación para la promoción y prevención integral de problemas socialmente relevantes, la educación física, la recreación, el deporte y la utilización adecuada del tiempo libre.

 - La promoción en la persona y en la sociedad de la capacidad para crear, investigar adoptar la tecnología que se requiere en los procesos de desarrollo del país y le permita al educando ingresar al sector productivo.

 - El estudio y la comprensión crítica de la cultura nacional y de la diversidad étnica y cultural del país como fundamento de la unidad y de su identidad;

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

- 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;

PRINCIPIOS Y FUNDAMENTOS QUE ORIENTAN LA COMUNIDAD EDUCATIVA.

La Educación es un proceso de formación que propende por la construcción de un hombre que es proyecto de vida, y que está en constante crecimiento, a través del conocimiento, valores y acciones, en búsqueda de una estructura moral y de un mundo con justicia, solidaridad, participación democrática y humana.

Principios éticos. El respeto a la persona, debe ser fundamento que marque las relaciones entre los integrantes de la comunidad educativa.

Principios Axiológicos. La responsabilidad en el pensar y sentir, orientarán las acciones de los integrantes de la comunidad educativa, reflejando valores morales y humanos.

Principios semiológicos. La verdad, es el componente esencial en los procesos de comunicación oral y escrita.

Principio Filosófico. El Centro promoverá permanentemente la formación integral como elemento constante de su quehacer.

Principio Pedagógico. Las prácticas pedagógicas de los docentes deben girar entorno a la formación del individuo en forma integral, de tal manera que emplee estrategias innovadoras que les permitan fundamentar la comunicación y el diálogo abierto y propender por una mejor calidad de los procesos.

Principio Epistemológico. Las teorías que fundamentan la Ley General de Educación, guiarán el proceso de construcción de conocimiento y del currículo de acuerdo con las necesidades del contexto, estas deben ser objeto de análisis, revisión, renovación, innovación permanente, coherente con la idiosincrasia y características propias del contexto institucional.

PRINCIPIOS INSTITUCIONALES.

- La educación tiene carácter autónomo, democrático, participativo y pluralista, fundado en el respeto de la dignidad humana, en el trabajo y en la solidaridad de la comunidad educativa.

 - Servir a la comunidad y garantizar los derechos fundamentales.
-

- La constitución es norma de normas. En todo caso de incompatibilidad entre Constitución y el Manual de Convivencia, se aplicarán los principios constitucionales.
- Se reconoce, sin discriminación alguna, la primacía de los derechos inalienables de la persona.
- Se reconoce y protege la diversidad étnica y cultural de la nación colombiana.
- Es obligación del estado y de las personas, proteger las riquezas culturales y naturales de la nación.
- El castellano es la lengua oficial de la institución.

PROPÓSITOS INSTITUCIONALES.

- Alcanzar la excelencia académica desarrollando actividades pedagógicas que permitan el cumplimiento de los planes y programas curriculares adoptados en nuestra institución.
 - Fomentar la adquisición y cultivo de valores mediante el desarrollo del proyecto de ética y valores.
-

- Promover la defensa y cuidado del medio ambiente a través del proyecto de educación ambiental.
- Convertir el COLEGIO DISTRITAL SAN SALVADOR, en un espacio de formación democrática a través del Gobierno Escolar.
- Promover la actividad investigativa de los profesores y alumnos a partir del quehacer pedagógico.

9.3 POLÍTICAS INSTITUCIONALES.

La institución propende por actividades que desarrollen y hagan crecer al individuo en todas sus dimensiones tal cual esta señalado en la misión y principios institucionales:

- Consideramos al estudiante como el eje del proceso educativo y a los docentes como sus directos orientadores.
 - Se respetan, sin discriminación alguna, los derechos individuales de las personas, buscando propiciar un ambiente democrático en el cual se formen ciudadanos capaces de convivir pacíficamente.
-

- Consideramos que el conocimiento es el resultado de una construcción que puede hacer el hombre individual o colectivamente.
- Evaluamos la capacidad intelectual mediante el desarrollo de habilidades y destrezas.

OBJETIVOS DE LA INSTITUCIÓN.

Objetivo General.

Desarrollar procesos de formación permanente en lo cultural, lo social, lo personal, lo intelectual, lo político, fundamentados en una concepción integral de la persona humana, de su dignidad, de sus derechos y deberes, conforme al objeto de la Ley 115 del 8 de Febrero de 1.994, y de la necesidad propia de la comunidad en la cual está inmerso el Centro de Educación Básica N° 70 de Barranquilla.

Objetivos Específicos. De la Educación Básica Primaria.

- Desarrollar habilidades comunicativas básicas para la lectura oral y comprensiva de los códigos culturales, sociales, intelectuales y científicos de acuerdo a la lengua castellana.
-

- Desarrollar conocimientos matemáticos necesarios para mejorar y utilizar operaciones simples de cálculo y procedimientos lógicos elementales en diferentes situaciones, así como la capacidad de solucionar problemas cotidianos que impliquen estos conocimientos.

 - Ejecutar procesos que estimulen la asimilación de conceptos científicos, en las áreas del conocimiento que sean objeto de estudio, de acuerdo con el desarrollo intelectual, la edad y el medio.

 - Educar para la higiene y la salud de su propio cuerpo y la formación protección del medio ambiente.

 - Desarrollar el conocimiento y ejercitación del propio cuerpo mediante la práctica de la educación física, la recreación y los deportes, adecuados a su edad y conducentes a un desarrollo físico armónico.

 - Desarrollar habilidades artísticas mediante la expresión corporal, la representación, la música, la plástica y la literatura.

 - Desarrollar un pensamiento crítico y analítico a través de la participación y el cultivo de valores de organización social y convivencia humana.
-

- Conocer la Constitución política, la organización escolar, el manual de convivencia a través de la experiencia cotidiana para desempeñarse con autonomía en la sociedad.

Objetivos Específicos. De la Educación Básica Secundaria.

- Desarrollar la capacidad de comprender textos y expresar correctamente mensajes complejos orales y escritos en lengua castellana, así como para entender mediante un estudio sistemático, los diferentes elementos constitutivos de la lengua.
 - Desarrollar las capacidades para el razonamiento lógico mediante el dominio de sistemas numéricos, geométricos, lógicos, analíticos, de conjuntos de operaciones y relaciones así como de la ciencia, de la tecnología y de la vida cotidiana.
 - Desarrollar habilidades para la apropiación del conocimiento científico de los fenómenos químicos y biológicos, mediante la comprensión de leyes, el planteamiento de problemas y la observación experimental.
 - Desarrollar la comprensión de la dimensión práctica de los conocimientos teóricos de los conocimientos prácticos y la
-

capacidad para utilizarla en la solución de problemas cotidianos.

- Desarrollar actitudes favorables al conocimiento valoración y conservación de la naturaleza y el ambiente.

Objetivos Institucionales.

En la formación académica e investigativa:

- Orientar al joven en el aprendizaje de los conocimientos de manera crítica, problematizadora y participativa.
- Desarrollar habilidades y destrezas que le permitan al estudiante ser una persona creativa, observadora y transformadora de la sociedad.
- Estimular la autoestima de los jóvenes y la confianza en sus propias capacidades.

En la formación humana.

- Considerar al estudiante como centro del proceso educativo.
-

- Fomentar la adquisición y cultivo de valores tales como. Autonomía, solidaridad, responsabilidad, honestidad, justicia, etc.
 - El COLEGIO DISTRITAL SAN SALVADOR, pretende formar un individuo capaz de proyectarse con valores humanos propios de su formación ética.
 - Concienciar a los alumnos de las funciones que debe cumplir en la comunidad a la cual pertenece y debe hacer crecer y desarrollar.
 - El COLEGIO DISTRITAL SAN SALVADOR promoverá el desarrollo integral alumno en todas sus dimensiones, social, humano, cultural, política e intelectual.
-

10. ANALISIS DEL SISTEMA

10.1 ESPECIFICACION DE LAS ENTIDADES

- **ESTUDIANTE**
 - ID (Llave Primaria)
 - Nombres
 - Apellidos
 - Sexo
 - Estado
 - Fecha de Nacimiento
 - Dirección
 - Teléfono
 - Tipo de Cliente

 - **ACUDIENTE**
 - ID
 - Nombres
 - Apellidos
 - Dirección
 - Teléfono
 - Estado
 - Sexo
 - Parentesco
 - Fecha de Nacimiento
 - Tipo de Documento
-

- **DOCENTE**

- ID
- Nombres
- Apellidos
- Dirección
- Teléfono
- Num. De Escalafón
- Sexo
- Estado
- Fecha de Nacimiento
- Tipo de Documento

- **MATRICULA**

- NumMatricula
- Fecha
- Estado
- Observaciones
- Año

- **ASIGNATURA**

- Código
- Nombre
- Estado

- **CURSO**

- Código
 - Nombre
 - Estado
-

- **GRUPO**
 - Código
 - Nombre
 - Estado

 - **PENSUM**
 - Código
 - Nombre
 - Estado

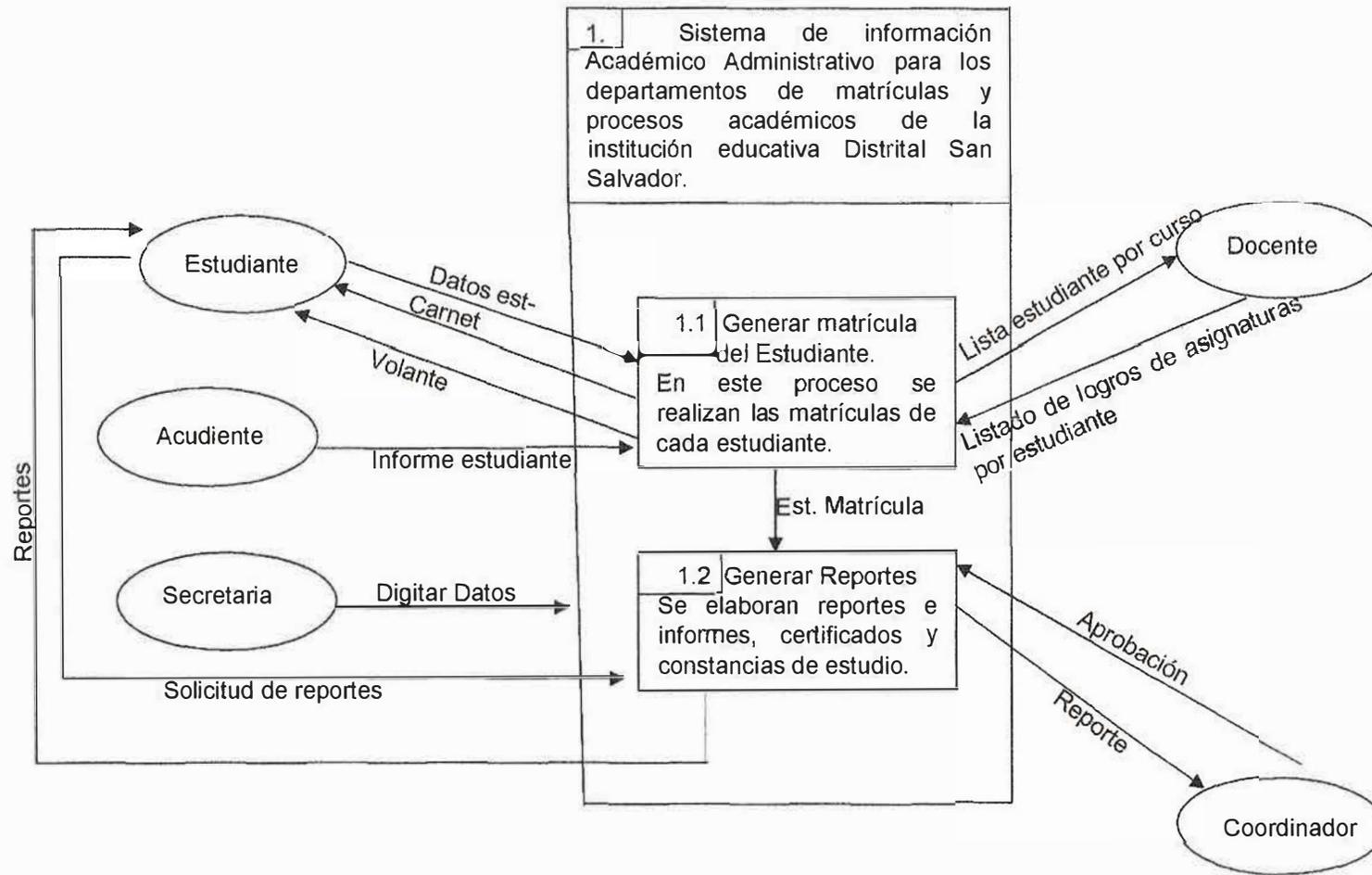
 - **LOGRO**
 - Código
 - Descripción
 - Estado

 - **PROFESION**
 - Código
 - Nombre
 - Estado

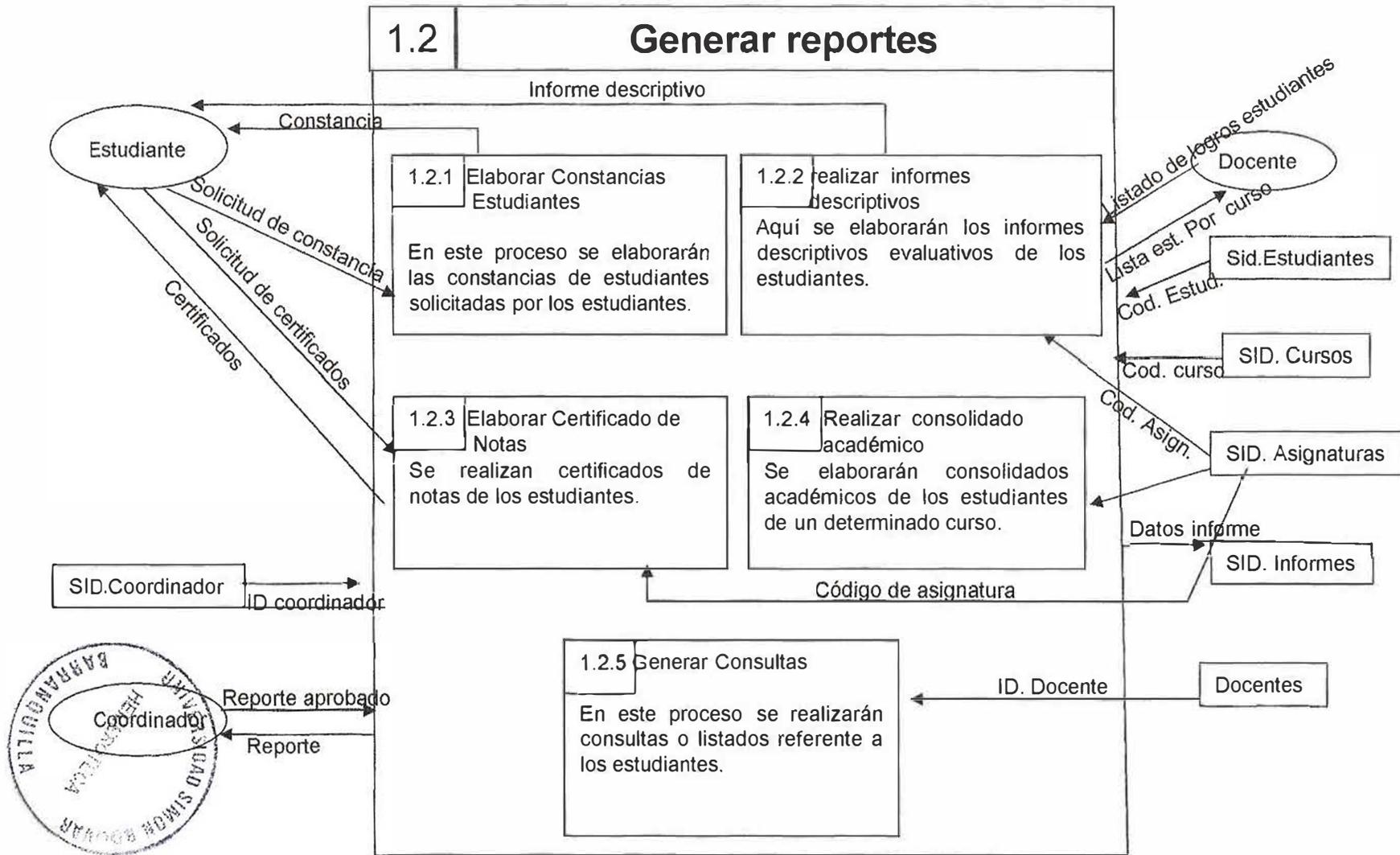
 - **JORNADA**
 - Código
 - Nombre
-

10.2 DIAGRAMA DE FLUJO DE DATOS DEL PROYECTO

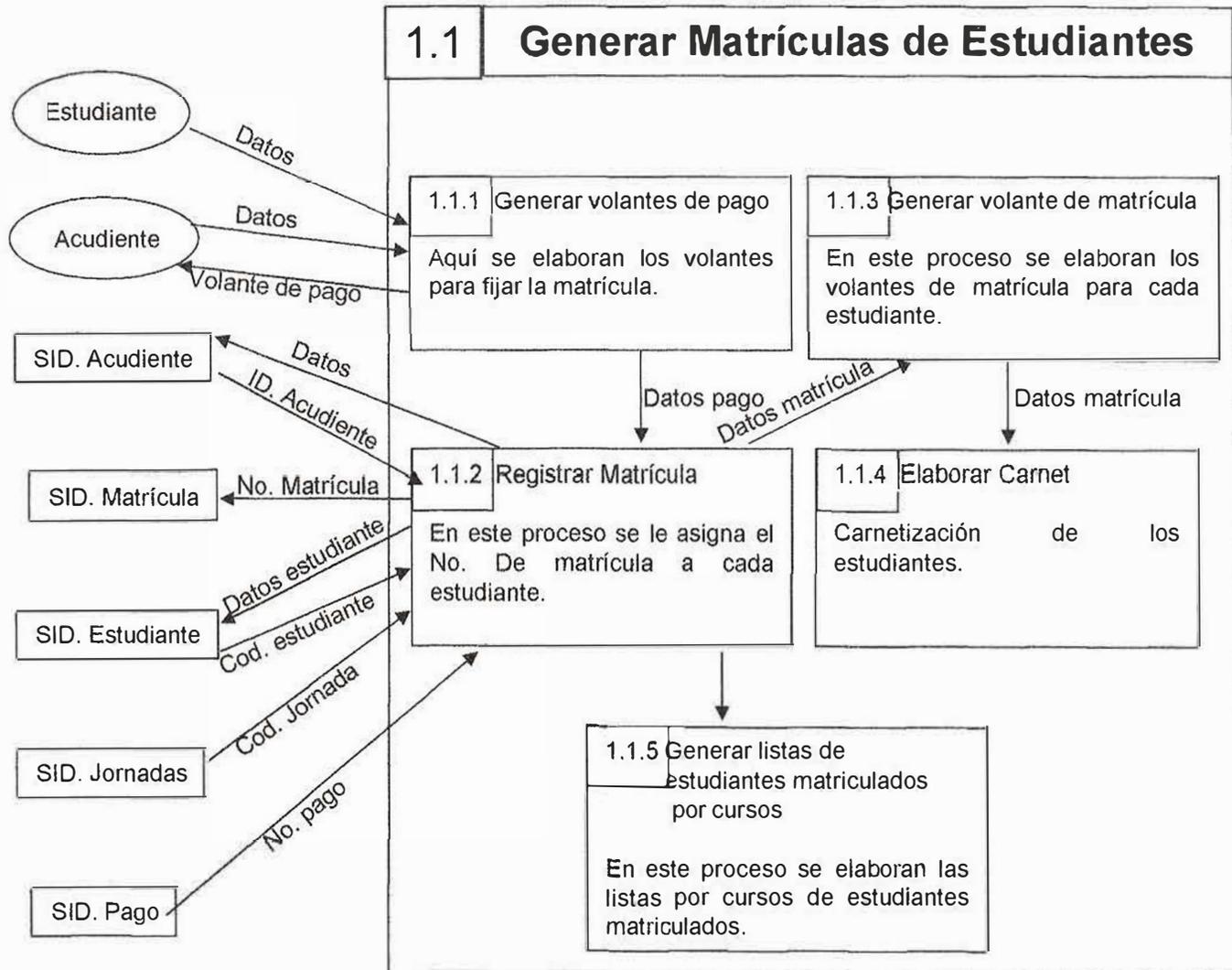
10.2.1 Diagrama de Contexto



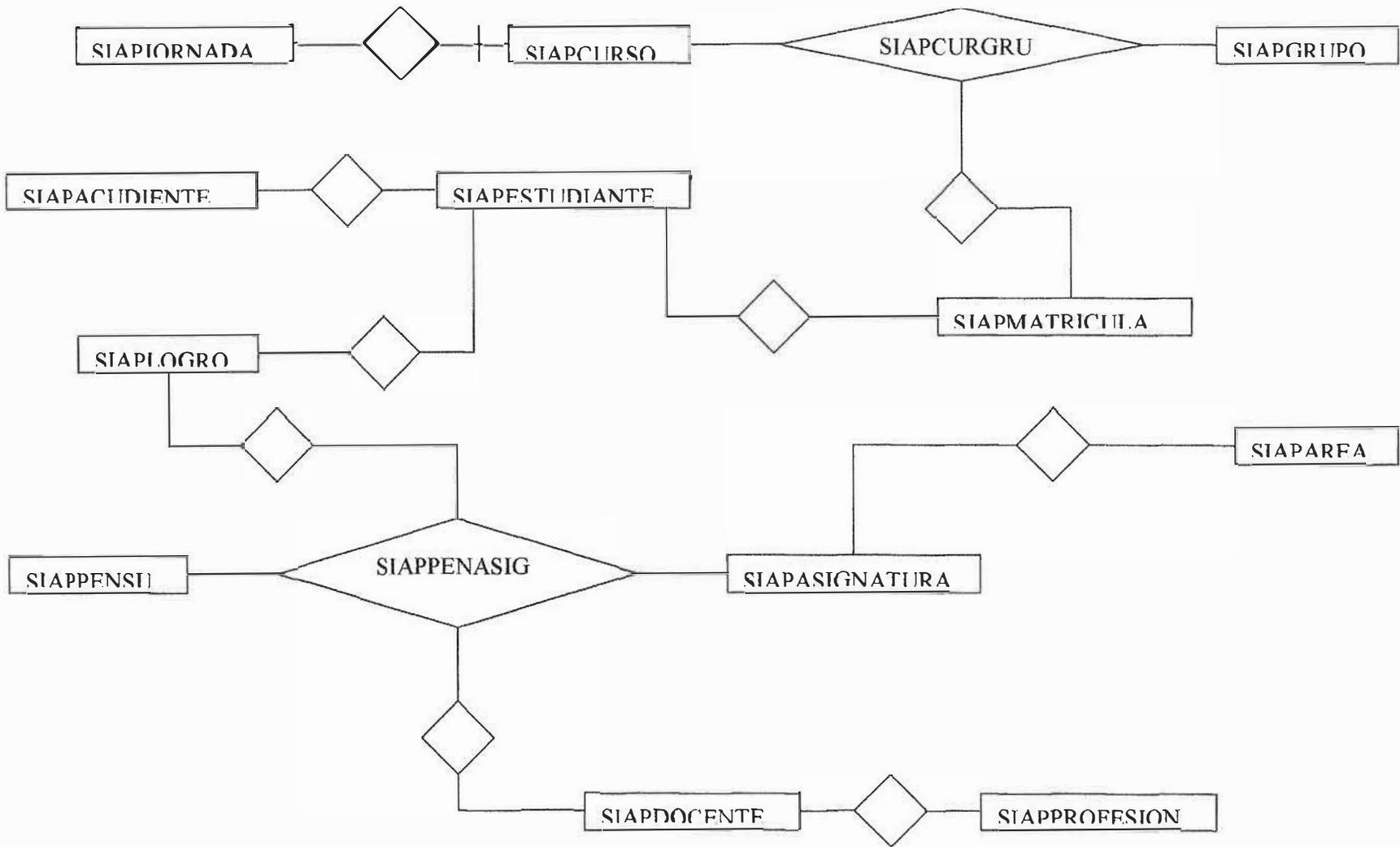
10.2.2 Diagrama de Nivel 2



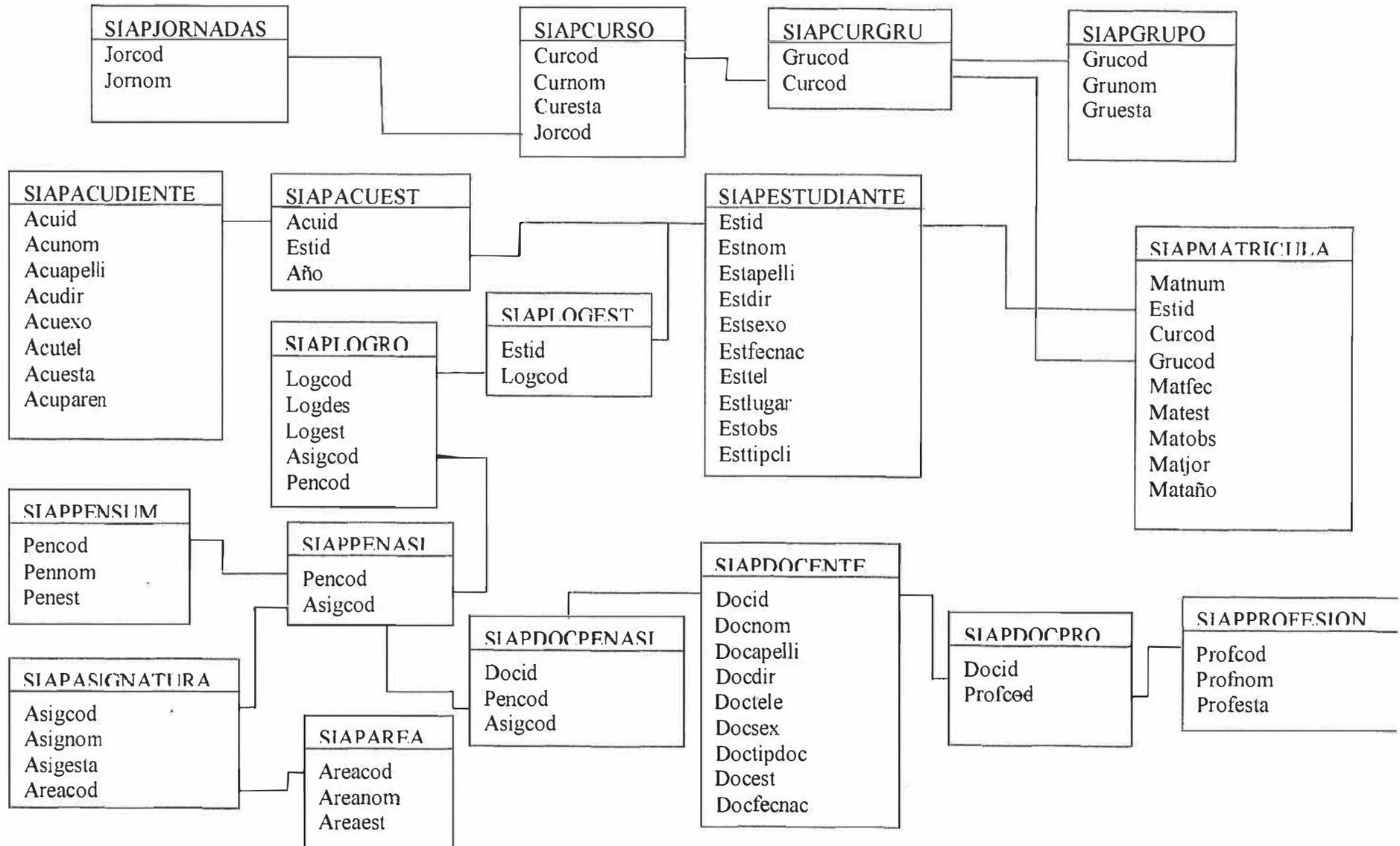
10.2.3 Diagrama de Nivel 1



10.3 MODELO ENTIDAD RELACION



10.4 MODELO RELACIONAL



10.5 DICCIONARIO DE DATOS

10.5.1 Tabla Estudiantes

Nombre tabla: SIAPESTUDIANTES	Descripción de la tabla:						
Nombre Largo: SIAPESTUDIANTES	Almacena los datos personales de los estudiantes o aspirantes que pertenezcan o que se encuentren interesados en formar parte de esta institución.						
Aplicación: ACADÉMICO _ ADMINISTRATIVO							
N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ESTID	Primaria	Varchar	10	0	No	Identificación del estudiante
2.	ESTNOM		Varchar	30	0	No	Nombres del estudiante
3.	ESTAPELLI		Varchar	30	0	No	Apellidos del estudiante
4.	ESTDIR		Varchar	20	0	No	Dirección del estudiante
5.	ESTTEL		Varchar	10	0	No	Teléfono del estudiante
6.	ESTSEX		Varchar	10	0	No	Sexo del estudiante
7.	ESTOBS		Varchar	80	0	No	Observaciones del estudiante
8.	ESTTIPDOC		Varchar	20	0	No	Tipo de documento del estudiante
9.	ESTFECNAC		Fecha	10	0	No	Fecha de nacimiento del estudiante

10.5.2 Tabla Cursos

Nombre tabla: SIAPCURSOS

Nombre Largo: SIAPCURSOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla cursos es el archivo en el cual se van a mantener guardados los datos correspondientes a los cursos con que cuenta la institución por año

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	CURCOD	Primaria	Varchar	2	0	No	Codigo del curso
2.	CURNOM		Varchar	15	0	No	Nombre del curso
3.	CURESTA		Varchar	10	0	No	Estado del curso
4.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la jornada

10.5.3 Tabla Grupos

Nombre tabla:SIAPGRUPOS

Nombre Largo: SIAPGRUPOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla grupos es la encargada de guardar los datos de cada uno de los grupos que existen respectivamente por cada curso

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	GRUCOD	Primaria	Varchar	2	0	No	Codigo del grupo
2.	GRUNOM		Varchar	15	0	No	Nombre del grupo
3.	GRUESTA		Varchar	10	0	No	Estado del grupo

10.5.4 Tabla Pensum

Nombre tabla: SIAPPENSUM

Nombre Largo: SIAPPENSUM

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla siappensum posee los diferentes pensum o asignación de asignaturas a los diferentes cursos existentes en la institucion

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PENCOD	Primaria	Varchar	2	0	No	Codigo del pensum
2.	PENNOM		Varchar	15	0	No	Nombre del pensum
3.	PENESTA		Varchar	10	0	No	Estado del pensum

10.5.5 Tabla Asignaturas

Nombre tabla:SIAPASIGNATURAS

Nombre Largo: SIAPASIGNATURAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla cursos es el archivo en el cual se van a mantener guardados los datos correspondientes a los cursos con que cuenta la institución por año

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ASIGCOD	Primaria	Varchar	2	0	No	Codigo de la asignatura
2.	ASIGNOM		Varchar	20	0	No	Nombre de la asignatura
3.	ASIGESTA		Varchar	10	0	No	Estado de la asignatura

10.5.6 Tabla Areas

Nombre tabla: SIAPAREAS

Descripción: La tabla areas es la encargada de llevar las areas de trabajo academico en la institución divididas en ciencias

Nombre Largo: SIAPAREAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	AREACOD	Primaria	Varchar	2	0	No	Codigo del area
2.	AREANOM		Varchar	20	0	No	Nombre del area
3.	AREAESTA		Varchar	10	0	No	Estado del area

10.5.7 Tabla Logros

Nombre tabla: SIAPLOGROS

Nombre Largo: SIAPLOGROS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: En esta tabla almacenamos los diferentes logros estipulados por los docentes de la institución como medio de evaluación a los estudiantes, cada uno de ellos identificados por un unico codigo

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	LOGCOD	Primaria	Varchar	2	0	No	Codigo del logro
2.	DESCRIPCION		Varchar	200	0	No	Descripcion
3.	LOGESTA		Varchar	10	0	No	Estado del area
4.	ASIGCOD	Foranea	Varchar	2	0	No	Codigo de la asignatura
5.	PENCOD	Foranea	Varchar	2	0	No	Codigo del pensum

10.5.8 Tabla Acudientes

Nombre tabla: SIAPACUDIENES

Descripción de la tabla:

Nombre Largo: SIAPACUDIENES

La tabla acudiente es la encargada de guardar los datos personales del acudiente y su relacion con el estudiante.

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ACUID	Primaria	Varchar	10	0	No	Identificación del acudiente
2.	ACUNOM		Varchar	30	0	No	Nombres del acudiente
3.	ACUAPELLI		Varchar	30	0	No	Apellidos del acudiente
4.	ACUDIR		Varchar	20	0	No	Direccion del acudiente
5.	ACUTEL		Varchar	10	0	No	Telefono del acudiente
6.	ACUSEX		Varchar	10	0	No	Sexo del acudiente
7.	ACUPAREN		Varchar	20	0	No	Parentesco del acudiente
8.	ACUFECNAC		Fecha	20	0	No	Fecha de nacimiento del acudiente



10.5.9 Tabla Docentes

Nombre tabla: SIAPDOCENTES

Descripción de la tabla:

Nombre Largo: SIAPDOCENTES

En esta tabla se almacenaran los datos personales de los docentes que laboran en la institución.

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	DOCNOM		Varchar	30	0	No	Nombres del docente
3.	DOCAPELLI		Varchar	30	0	No	Apellidos del docente
4.	DOCDIR		Varchar	20	0	No	Direccion del docente
5.	DOCTEL		Varchar	10	0	No	Telefono del docente
6.	DOCSEX		Varchar	10	0	No	Sexo del docente
7.	DOCTIPDOC		Varchar	20	0	No	Tipo de documento del estudiante
8.	DOCFECNAC		Fecha	10	0	No	Fecha de nacimiento del docente
9.	DOCESTA		Varchar	10	0	No	Estado del docente
10.	DOCNUMESC		Varchar	2	0	No	Numero de escalafon del docente

10.5.10 Tabla Cursos_Grupos

Nombre tabla: SIAPCURGRU

Descripción: En esta tabla almacenamos la relacion entre los cursos y los diferentes grupos que hay en la institución y el año de creación del curso.

Nombre Largo: SIAPCURSOSGRUPOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	CURCOD	Foranea	Varchar	2	0	No	Codigo del curso
2.	GRUCOD	Foranea	Varchar	2	0	No	Codigo del grupo
3.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la Jornada

10.5.11 Tabla Pensum_Asignatura

Nombre tabla: SIAPPENASIG

Descripción: En esta tabla se manejan directamente la relacion de las asignaturas con los diferentes pensum existentes por cursos

Nombre Largo: SIAPPENSUMASIGNATURA

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PENCOD	Foranea	Varchar	2	0	No	Codigo del pensum
2.	ASIGCOD	Foranea	Varchar	2	0	No	Codigo de la asignatura

10.5.12 Tabla Matriculas

Nombre tabla: SIAPMATRICULAS

Descripción de la tabla:

Nombre Largo: SIAPMATRICULAS

En esta tabla se guardaran los datos correspondientes a las matriculas realizadas anualmente por los estudiantes por cada curso

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	MATNUM	Primaria	Varchar	6	0	No	Numero de matricula
2.	ESTID	Foranea	Varchar	10	0	No	Identficação del estudiante
3.	CURCOD	Foranea	Varchar	2	0	No	Codigo del curso
4.	GRUCOD	Foranea	Varchar	10	0	No	Codigo del grupo
5.	MATFEC		Fecha		0	No	Fecha de matricula
6.	MATEST		Varchar	10	0	No	Estado de matricula
7.	MATOBS		Varchar	200	0	No	Observaciones de la matricula
8.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la jornada
9.	MATAÑO		Varchar	4	0	No	Año de matricula

10.5.13 Tabla Profesiones

Nombre tabla: SIAPPROFESIONES

Descripción: La tabla profesiones es la encargada de llevar las diferentes profesiones de trabajo que posee el personal que labora en la institución.

Nombre Largo: SIAPPROFESIONES

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Nº	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PROFCOD	Primaria	Varchar	2	0	No	Codigo de la profesion
2.	PROFNOM		Varchar	20	0	No	Nombre de la profesion
3.	PROFESTA		Varchar	10	0	No	Estado de la profesion

10.5.14 Tabla Jornadas

Nombre tabla: SIAPJORNADAS

Nombre Largo: SIAPJORNADAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla jornadas se encarga de llevar las jornadas de estudio en la institución.

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	JORCOD	Primaria	Varchar	10	0	No	Codigo de la jornada
2.	JORNOM		Varchar	20	0	No	Nombre de la jornada
3.	JORFESTA		Varchar	10	0	No	Estado de la jornada

10.5.15 Tabla Logro _ estudiante

Nombre tabla: SIAPLOGEST

Descripción: La tabla logro _ estudiante se encarga de asignarle logros a cada estudiante.

Nombre Largo: SIAPLOGEST

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	LOGCOD	Primaria	Varchar	2	0	No	Codigo de la profesion
2.	ESTID		Varchar	20	0	No	Nombre de la profesion
3.	AÑO		Varchar	4	0	No	Año lectivo
4.	PERIODO		Varchar	10	0	No	Periodo de trabajo

10.5.16 Tabla Docente_Pensum_Asignatura

Nombre tabla: SIAPDOCPENASI

Descripción: La presente tabla registra las asignaturas de un determinado pensum dadas por un docente en la institución

Nombre Largo:

SIAPDOCENTEPEPENSUMASIGNATURA

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	PENCOD	Foranea	Varchar	2	0	No	Codigo del pensum
3.	ASIGCOD	Foranea	Varchar	2	0	No	Codigo de la asignatura
4.	AÑO	Foranea	Varchar	4	0	No	Año lectivo

10.5.17 Tabla Docente _profesion

Nombre tabla: SIAPDOCPRO

Descripción: La presente tabla registra las asignaturas de un determinado peunsum dadas por un docente en la institucion

Nombre Largo: SIAPDOCENTEPROFESION

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	PROFCOD	Foranea	Varchar	2	0	No	Codigo de la profesion

10.5.18 Tabla Acudiente_estudiante

Nombre tabla: SIAPACUEST

Descripción: La presente tabla registra las asignaturas de un determinado peunsum dadas por un docente en la institucion

Nombre Largo: SIAPACUDIENETEESTUDIANTE

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ESTID	Foranea	Varchar	10	0	No	Identificación del estudiante
2.	ACUID	Foranea	Varchar	10	0	No	Identificación del acudiente
3.	AÑO		Varchar	4	0	No	Año de asignación

10.6 DICCIONARIO DE FLUJOS

ENTIDADES

ELEMENTOS DE DATOS	ESTUDIANTE
DESCRIPCION	Persona que desea ingresar a la institución y cumplir con un programa académico

ELEMENTOS DE DATOS	ACUDIENTE
DESCRIPCION	Persona que representa a uno o mas estudiantes

ELEMENTOS DE DATOS	SECRETARIA
DESCRIPCION	Persona que interactúa con el sistema(usuario)

ELEMENTOS DE DATOS	DOCENTE
DESCRIPCION	Persona que recibe listado de estudiantes para luego evaluarlos

ELEMENTOS DE DATOS	COORDINADOR
DESCRIPCION	Persona que se encarga de revisar y aprobar los reportes

ALMACENAMIENTO

ESTRUCTURAS DE DATOS	SIAPESTUDIANTE
DESCRIPCION	Persona que desea ingresar a la institución y cumplir con un programa académico
CONTENIDO	Información general del estudiante

ESTRUCTURAS DE DATOS	SIAPACUDIENTE
DESCRIPCION	Tabla donde se guardan los datos del acudiente
CONTENIDO	Información general del acudiente

ESTRUCTURAS DE DATOS	SIAPMATRICULA
DESCRIPCION	Es una tabla donde se guardan los datos relacionados con la matricula del estudiante
CONTENIDO	Información general de la matricula

ESTRUCTURAS DE DATOS	SIAPJORNADAS
DESCRIPCION	Persona que recibe listado de estudiantes para luego evaluarlos
CONTENIDO	Información general de jornadas de estudio

ESTRUCTURAS DE DATOS	SIAPCURSOS
DESCRIPCION	Tabla donde se guarda la información referente a los cursos
CONTENIDO	Información general de los cursos

ESTRUCTURAS DE DATOS	SIAPASIGNATURAS
DESCRIPCION	Tabla donde se guarda la información referente a Las asignaturas
CONTENIDO	Información general de las asignaturas

ESTRUCTURAS DE DATOS	SIAPDOCENTES
DESCRIPCION	Tabla donde se guarda la información referente a los datos del maestro
CONTENIDO	Información general del docente

11. DISEÑO DEL SISTEMA

11.1 ESTRUCTURA FUNCIONAL (Diagrama Jerárquico de Funciones (Menú))

Sistema de Información Académico-Administrativo (Institución Educativa Distrital San Salvador)				
Datos Básicos		Procesos		Sistema
				Ayuda
				Salir
Administradores		Evaluaciones		Generador de reportes
Área		Matriculas >		Auditoria
Asignatura				Impor / Export
Curso	>	>Creación >Asignación		
Docente				
Grupo				
Logro				
Pensum	>	> Creación		
Profesiones				
Estudiantes				

11.2 DESCRIPCION DE LOS COMPONENTES DE LA ESTRUCTURA FUNCIONAL

El diagrama funcional del sistema se encuentra dividido en tres submenús, que son: Datos Básicos, Procesos, Listados, Ayuda y Salir.

Datos Básicos:

- Administradores: Permite registrar los usuarios y el tipo de usuarios que pueden tener acceso al sistema.
 - Áreas: Permite registrar los campos de investigación y estudio en la institución.
 - Asignatura: Permite llevar registros de las asignaturas que se dictan en la institución pertenecientes a una determinada área.
 - Curso: Permite registrar los cursos que se van a abrir en la institución, y a su vez asignarlos a un grupo específico de una determinada jornada.
 - Grupo: Registra los grupos en la institución, llevando un conjunto de cursos pertenecientes a este.
-

- Logro: Se encarga de registrar los logros evaluativos de una determinada asignatura.
- Pensum: Registra el programa educativo a desarrollar en un curso en un año determinado.
- Profesiones: Se llevan registros de todos los campos de trabajo profesionales en la institución, de diferentes especialidades de los docentes por título.
- Estudiantes: Aquí se llevan registros de los estudiantes y/o aspirantes a ingresar en la institución. También se encarga de llevar los registros de los acudientes en general y aquellos que pertenecen a un determinado estudiante.

Procesos:

- Evaluaciones: permite llevar el seguimiento académico de un estudiante en una determinada asignatura, con respecto al programa que se lleva a cabo en esta, evaluado por logros.
-

- **Matriculas:** permite llevar registros de todas las matriculas de los estudiantes que hacen parte de la institución.

Sistema:

- **Generador de Reportes:** permite realizar consultas estaticas y dinamicas requeridas por los usuarios.
- **Auditoria:** permite mostrar información acerca de las transacciones realizadas por los diferentes usuarios del sistema (Solo en el caso de Administrador).
- **Impor / Export:** permite realizar exportaciones e importaciones de las tablas de la base de datos del sistema.

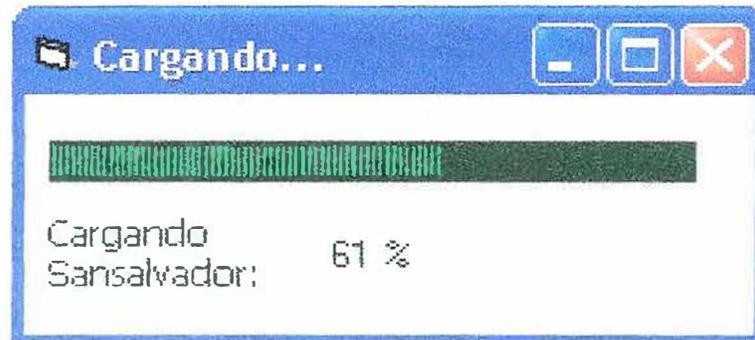
Ayudas:

- **Ayudas:** muestra las ayudas del sistema con su descripción para mayor comprensión al usuario.

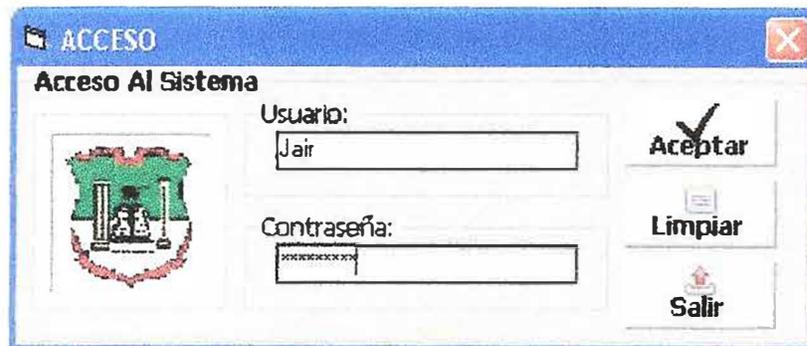
Salir:

- Permite cerrar la aplicación.
-

11.3 DISEÑO DE LAS INTERFACES DEL SISTEMA

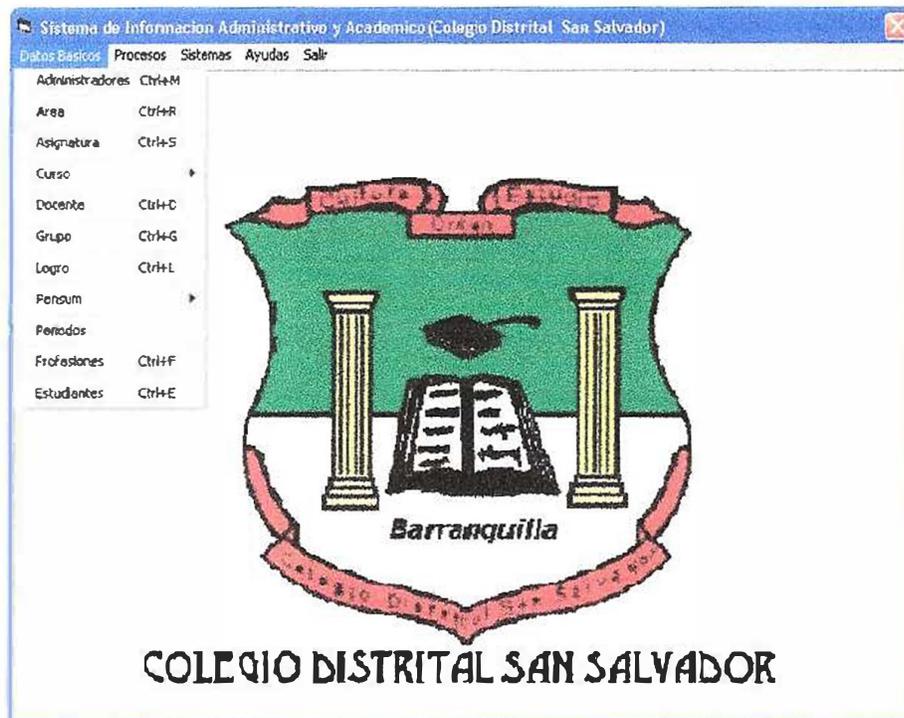


PANTALLA DE CARGA DEL SISTEMA



PANTALLA DE ACCESO AL SISTEMA





PANTALLA MENU PRINCIPAL CON EL SUB MENU DATOS BASICOS

PANTALLA ESTUDIANTES

SIAP (ESTUDIANTES)

Adicionar Actualizar Consultar Eliminar Limpiar Salir

ESTUDIANTES **ACUDIENTES**

Selección
 Ingresar Acudientes Asignación de Acudientes

Datos Básicos Del Acudiente

Id: Telefono:

Nombres: Fecha: 11/05/2005

Apellido: Dirección:

Sexo: Seleccione Parentesco: Seleccione

Estado: Activo

PANTALLA ACUDIENTES

Logros

Adicionar Actualizar Consultar Eliminar Limpiar Salir

Datos Generales del Logro

Código de La Asignatura

Código del Logro

Calificador

Descripción

Estado

[E]=Exelente
 [B]=Bueno
 [A]=Aceptable
 [D]=Deficiente
 [I]=Insuficiente

PANTALLA DE LOGROS

SIAP (MATRICULAS)

Adicionar Actualizar Consultar Eliminar Limpiar Salir

REGISTROS DE MATRICULAS

Proceso de Matricula

Matricula No:

Id del Estudiante:

Jornada:

Curso:

Grupo:

Fecha de Matricula: 01/05/2005 Estado

Observaciones:

PANTALLA MATRICULAS

CONSULTA DE ESTUDIANTES

Codigo	Nombres

Enviar Cancel

PANTALLA DE CONSULTA DE ESTUDIANTES

Valoración

 Adicionar
  Actualizar
  Consultar
  Eliminar
  Limpiar
  Salir
 

Busqueda de Estudiantes por Jornada-Curso-Grupo

Jornada: Curso: Grupo:

 Identificación
 Matricula: Año Lectivo:

Datos Generales del Periodo

Periodo Estado

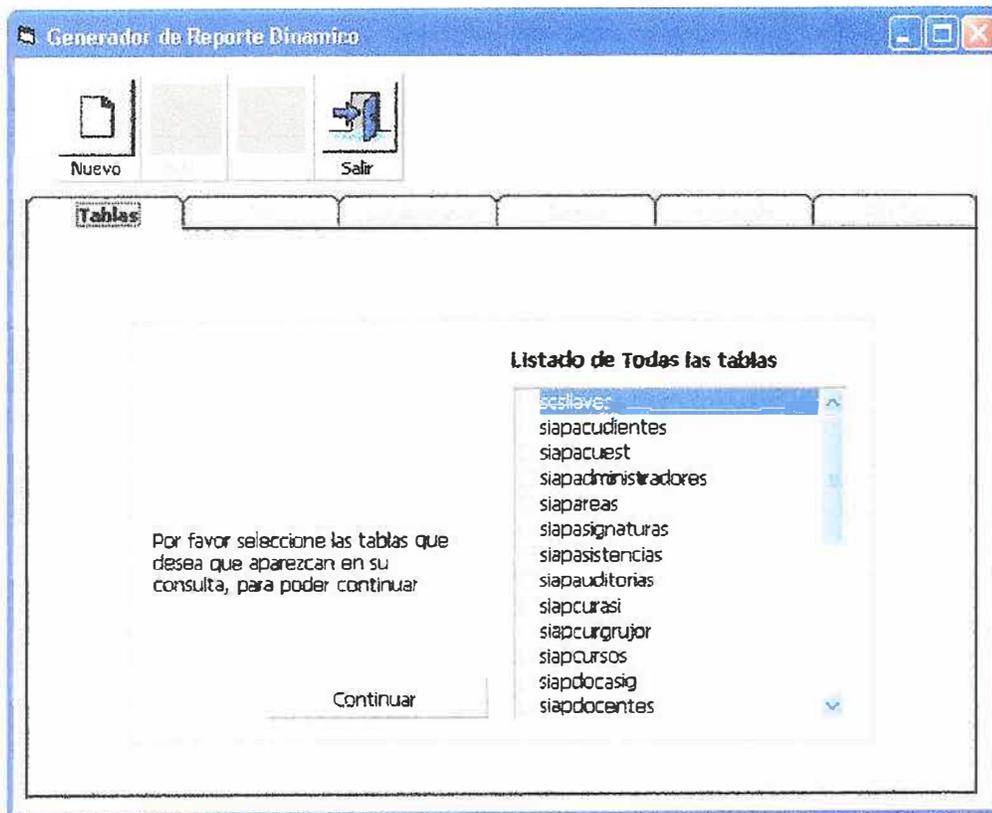
Datos de la Valoración (Logros por Asignaturas)

Asignaturas

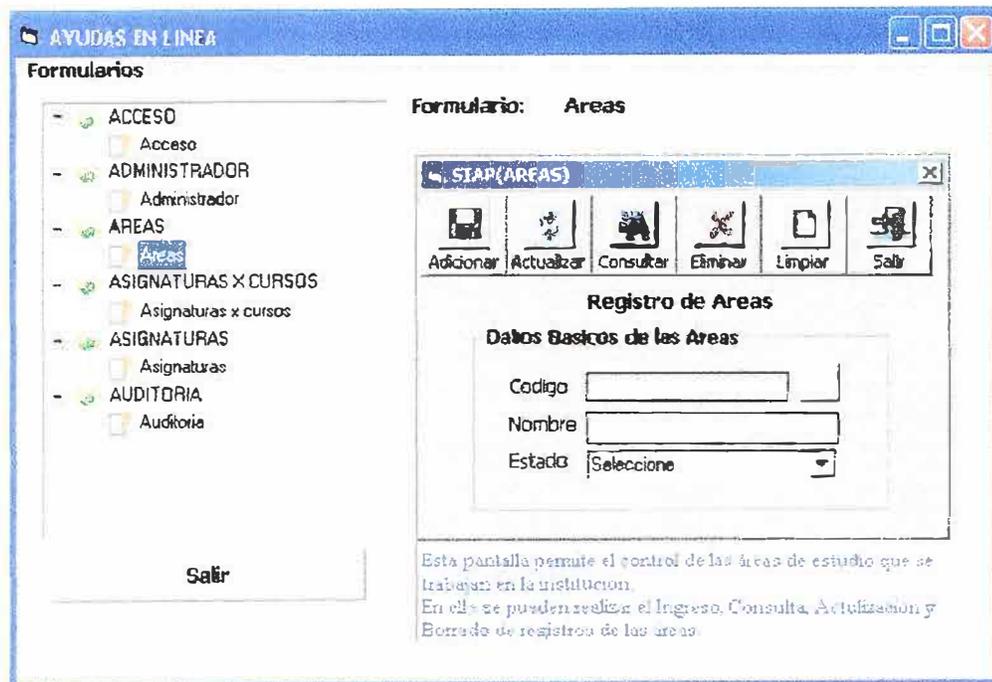
Cod Asig	Nombre	Cal	Descripcion

Observaciones

PANTALLA DE EVALUACIONES



PANTALLA DE GENERADOR DE REPORTES



PANTALLA DE AYUDAS EN LINEA

12. RECOMENDACIONES

CONCLUSION

La realización del presente proyecto nos ha permitido conocer la realidad de una comunidad educativa la cual estaba atravesando por una situación muy difícil. Gracias a los ingenieros asesores y el personal directivo del Colegio Distrital San Salvador, hemos podido darle una solución factible y óptima para que los procesos que se llevan a cabo en la institución se realicen en una mejor manera

El propósito del presente proyecto ha sido colaborar, en la medida de lo posible, con el mejoramiento de esta situación brindando una herramienta que permita automatizar los procesos académicos y de matrículas, llevando consigo todos los lineamientos y metodologías vistas en el transcurso del trabajo

Por otro lado, el proyecto nos ha permitido además de desarrollar y aplicar los conocimientos adquiridos en las aulas sobre todo en el área de Ingeniería del software.

BIBLIOGRAFIA

Stair, Ralph. Sistemas de información. Editorial MC Graw Hill

Vergel Cabrales, Gustavo. Metodología.

[http://www.google.com/sistemas de informaci3n.](http://www.google.com/sistemas%20de%20informaci3n)

[http://www.monografias.com/sistemas de informaci3n.](http://www.monografias.com/sistemas%20de%20informaci3n)

[http://www.altavista.com/sistemas de informaci3n.](http://www.altavista.com/sistemas%20de%20informaci3n)

ANEXOS

SCRIPT DE CREACION DE LA BASE DE DATOS

Table structure for table 'siapacuest'

#

```
CREATE TABLE siapacudientes (  
  ACUID varchar(10) NOT NULL default '0',  
  ACUNOMBRE varchar(30) NOT NULL default "",  
  ACUAPELLI varchar(30) NOT NULL default "",  
  ACUSEXO varchar(10) NOT NULL default "",  
  ACUTELE int(15) NOT NULL default '0',  
  ACUFECNAC varchar(10) NOT NULL default '0',  
  ACUDIR varchar(20) NOT NULL default "",  
  ACUPAREN varchar(15) NOT NULL default "",  
  ACUESTA varchar(10) NOT NULL default "",  
  PRIMARY KEY (ACUID)  
) TYPE=MyISAM;
```

Table structure for table 'siapacuest'

#

```
CREATE TABLE siapacuest (  
  ESTID varchar(10) NOT NULL default '0',  
  ACUID varchar(10) NOT NULL default '0',  
  PRIMARY KEY (ESTID,ACUID)  
) TYPE=MyISAM;
```

Table structure for table 'siapadministradores'

#

```
CREATE TABLE siapadministradores (  
  ADMCOD char(2) NOT NULL default '0',  
  ADMNOM varchar(20) NOT NULL default "",  
  ADMCON varchar(6) NOT NULL default "",
```

```
PRIMARY KEY (ADMCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapareas'

#

```
CREATE TABLE siapareas (
  AREACOD char(2) NOT NULL default '0',
  AREANOM varchar(20) NOT NULL default '0',
  AREAEST varchar(10) NOT NULL default '0',
  PRIMARY KEY (AREACOD)
) TYPE=MyISAM;
```

Table structure for table 'siapasignaturas'

#

```
CREATE TABLE siapasignaturas (
  ASIGCOD char(2) NOT NULL default "",
  ASIGNOM varchar(15) NOT NULL default "",
  ASIGESTA varchar(10) NOT NULL default "",
  AREANOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ASIGCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapcurrupen'

#

```
CREATE TABLE siapcurrupen (
  CURCOD char(2) NOT NULL default '0',
  GRUCOD char(2) NOT NULL default '0',
  PENCOD char(2) NOT NULL default '0',
  jornada varchar(15) NOT NULL default "",
  año varchar(15) NOT NULL default "",
  PRIMARY KEY (CURCOD,GRUCOD,PENCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapcursos'

#

```
CREATE TABLE siapcursos (  
  CURCOD char(2) NOT NULL default '0',  
  CURNOM varchar(15) NOT NULL default '0',  
  CURESTA varchar(10) NOT NULL default '0',  
  PRIMARY KEY (CURCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapdocasig'

#

```
CREATE TABLE siapdocasig (  
  DOCID varchar(10) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PRIMARY KEY (DOCID,ASIGCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapdocentes'

#

```
CREATE TABLE siapdocentes (  
  DOCID varchar(10) NOT NULL default '0',  
  DOCNOM varchar(25) NOT NULL default '0',  
  DOCAPELLI varchar(25) NOT NULL default '0',  
  DOCNUMESC char(2) NOT NULL default '0',  
  DOCSEXO varchar(10) NOT NULL default '0',  
  DOCFECNAC varchar(10) NOT NULL default '0',  
  DOCTELE int(15) NOT NULL default '0',  
  DOCTIPDOC varchar(20) NOT NULL default '0',  
  DOCESTA varchar(10) NOT NULL default '0',  
  DOCDIR varchar(20) NOT NULL default "",
```

```
PRIMARY KEY (DOCID)
) TYPE=MyISAM;
```

Table structure for table 'siapestudiantes'

#

```
CREATE TABLE siapestudiantes (
  ESTID varchar(10) NOT NULL default '0',
  ESTTIPDOC varchar(20) NOT NULL default "",
  ESTNOMBRE varchar(30) NOT NULL default "",
  ESTAPELLI varchar(30) NOT NULL default "",
  ESTSEXO varchar(15) NOT NULL default "",
  ESTFECNAC varchar(15) NOT NULL default "",
  ESTTELE int(15) unsigned NOT NULL default '0',
  ESTOBS varchar(200) NOT NULL default "",
  ESTTIPCLI varchar(15) NOT NULL default "",
  ESTESTA varchar(10) NOT NULL default "",
  ESTLUGAR varchar(25) NOT NULL default "",
  ESTDIR varchar(25) NOT NULL default "",
  PRIMARY KEY (ESTID)
) TYPE=MyISAM;
```

Table structure for table 'siapgrupos'

#

```
CREATE TABLE siapgrupos (
  GRUCOD char(2) NOT NULL default '0',
  GRUNOM varchar(20) NOT NULL default '0',
  GRUESTA varchar(10) NOT NULL default '0',
  PRIMARY KEY (GRUCOD)
) TYPE=MyISAM;
```

Table structure for table 'siaplogros'

#

```
CREATE TABLE siaplogros (  
  LOGCOD char(3) NOT NULL default '0',  
  LOGDES varchar(200) NOT NULL default "",  
  LOGEST varchar(10) NOT NULL default "",  
  PENCOD char(2) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PRIMARY KEY (LOGCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapmatriculas'

#

```
CREATE TABLE siapmatriculas (  
  MATNUM char(3) NOT NULL default '0',  
  ESTID varchar(10) NOT NULL default "",  
  CURCOD char(2) NOT NULL default "",  
  GRUNOM varchar(30) NOT NULL default "",  
  MATFEC date NOT NULL default '0000-00-00',  
  MATEST varchar(15) NOT NULL default "",  
  MATOBS varchar(100) NOT NULL default "",  
  MATJOR varchar(10) NOT NULL default "",  
  PRIMARY KEY (MATNUM)  
) TYPE=MyISAM;
```

Table structure for table 'siappenasig'

#

```
CREATE TABLE siappenasig (  
  PENCOD char(2) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PENAÑO varchar(10) NOT NULL default "",  
  PRIMARY KEY (PENCOD,ASIGCOD)
```

```
) TYPE=MyISAM;
```

Table structure for table 'siappensum'

#

```
CREATE TABLE siappensum (  
  PENCOD char(2) NOT NULL default '0',  
  PENNOM varchar(30) NOT NULL default "",  
  PENEST varchar(10) NOT NULL default "",  
  PRIMARY KEY (PENCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapprofesiones'

#

```
CREATE TABLE siapprofesiones (  
  PROFCOD char(3) NOT NULL default '0',  
  PROFNOM varchar(30) NOT NULL default "",  
  PROFEST varchar(10) NOT NULL default "",  
  PRIMARY KEY (PROFCOD)  
) TYPE=MyISAM;
```

**SISTEMA DE INFORMACION ACADEMICO-
ADMINISTRATIVO PARA LA GESTION DE DATOS EN LOS
DEPARTAMENTOS DE MATRICULAS Y PROCESOS
ACADEMICOS EN LA INSTITUCION EDUCATIVA
DISTRITAL SAN SALVADOR DE BARRANQUILLA**

MANUAL DEL SISTEMA

**CORPORACION EDUCATIVA MAYOR DEL DESARROLLO
SIMON BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
AREA DE INVESTIGACION FORMATIVA IV
2005**

INDICE

INTRODUCCIÓN
1. OBJETIVOS DEL SISTEMA
2. REQUERIMIENTOS MÍNIMOS
3. DEFINICIÓN DE LA BASE DE DATOS
4. DICCIONARIO DE DATOS
5. DIAGRAMA ENTIDAD RELACIÓN
6. DIAGRAMA RELACIONAL
7. DIAGRAMA FUNCIONAL
8. DIAGRAMA DE FLUJO DE DATOS
9. CODIGO FUENTE DEL SISTEMA

INTRODUCCIÓN

La Institución Educativa Distrital San Salvador en su afán de agilizar los procesos académicos y administrativos, ha solicitado la creación de un Sistema de Información, que gestione los datos en los departamentos de Matricula y Procesos Académicos.

El objetivo primordial del Sistema, será la administración de la Información en los departamentos de Matricula y Procesos académicos.

En este manual se especificará de forma detallada el funcionamiento del Sistema de Información (SIAP), su objetivo o finalidad y los requerimientos básicos y técnicos.

OBJETIVOS DEL SISTEMA

- ✓ Permitir el manejo seguro y efectivo de la información en los departamentos de Procesos Académicos y Administración.
- ✓ Registrar los datos solicitados por la institución y permitir las diferentes transacciones aplicadas a estos (Consulta, actualización, eliminación, etc).
- ✓ Generar reportes evaluativos de los estudiantes por periodos en un año lectivo.
- ✓ Generar Planillas de asistencia y evaluaciones de estudiantes por curso.
- ✓ Interactuar de manera fácil y sencilla con el usuario mediante el uso de ayudas implementadas en el sistema.



REQUERIMIENTOS MÍNIMOS

- ✓ Sistema Operativo: Windows 98, 2000 Professional, XP.
 - ✓ Hardware: Intel Pentium III, 750 Mhz, 128 Mb RAM.
 - ✓ Programa: Microsoft Visual Basic 6.0.
 - ✓ Base de Datos: MySQL-Front.
-

DEFINICION DE LA BASE DE DATOS

SCRIPT DE LAS TABLAS

Table structure for table 'siapacuest'

#

```
CREATE TABLE siapacudientes (  
  ACUID varchar(10) NOT NULL default '0',  
  ACUNOMBRE varchar(30) NOT NULL default "",  
  ACUAPELLI varchar(30) NOT NULL default "",  
  ACUSEXO varchar(10) NOT NULL default "",  
  ACUTELE int(15) NOT NULL default '0',  
  ACUFECNAC varchar(10) NOT NULL default '0',  
  ACUDIR varchar(20) NOT NULL default "",  
  ACUPAREN varchar(15) NOT NULL default "",  
  ACUESTA varchar(10) NOT NULL default "",  
  PRIMARY KEY (ACUID)  
) TYPE=MyISAM;
```

Table structure for table 'siapacuest'

#

```
CREATE TABLE siapacuest (  
  ESTID varchar(10) NOT NULL default '0',  
  ACUID varchar(10) NOT NULL default '0',  
  PRIMARY KEY (ESTID,ACUID)  
) TYPE=MyISAM;
```

Table structure for table 'siapadministradores'

#

```
CREATE TABLE siapadministradores (  
  ADMCOD char(2) NOT NULL default '0',
```

```
ADMNOM varchar(20) NOT NULL default "",
ADMCON varchar(6) NOT NULL default "",
PRIMARY KEY (ADMCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapareas'

#

```
CREATE TABLE siapareas (
  AREACOD char(2) NOT NULL default '0',
  AREANOM varchar(20) NOT NULL default '0',
  AREAEST varchar(10) NOT NULL default '0',
  PRIMARY KEY (AREACOD)
) TYPE=MyISAM;
```

Table structure for table 'siapasignaturas'

#

```
CREATE TABLE siapasignaturas (
  ASIGCOD char(2) NOT NULL default "",
  ASIGNOM varchar(15) NOT NULL default "",
  ASIGESTA varchar(10) NOT NULL default "",
  AREANOM varchar(20) NOT NULL default "",
  PRIMARY KEY (ASIGCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapcurgruPen'

#

```
CREATE TABLE siapcurgruPen (
  CURCOD char(2) NOT NULL default '0',
  GRUCOD char(2) NOT NULL default '0',
  PENCOD char(2) NOT NULL default '0',
  jornada varchar(15) NOT NULL default "",
  año varchar(15) NOT NULL default "",
  PRIMARY KEY (CURCOD,GRUCOD,PENCOD)
) TYPE=MyISAM;
```

Table structure for table 'siapcursos'

#

```
CREATE TABLE siapcursos (  
  CURCOD char(2) NOT NULL default '0',  
  CURNOM varchar(15) NOT NULL default '0',  
  CURESTA varchar(10) NOT NULL default '0',  
  PRIMARY KEY (CURCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapdocasig'

#

```
CREATE TABLE siapdocasig (  
  DOCID varchar(10) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PRIMARY KEY (DOCID,ASIGCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siapdocentes'

#

```
CREATE TABLE siapdocentes (  
  DOCID varchar(10) NOT NULL default '0',  
  DOCNOM varchar(25) NOT NULL default '0',  
  DOCAPELLI varchar(25) NOT NULL default '0',  
  DOCNUMESC char(2) NOT NULL default '0',  
  DOCSEXO varchar(10) NOT NULL default '0',  
  DOCFECNAC varchar(10) NOT NULL default '0',  
  DOCTELE int(15) NOT NULL default '0',  
  DOCTIPDOC varchar(20) NOT NULL default '0',  
  DOCESTA varchar(10) NOT NULL default '0',  
  DOCDIR varchar(20) NOT NULL default "",  
  PRIMARY KEY (DOCID)  
) TYPE=MyISAM;
```

Table structure for table 'siapeestudiantes'

#

```
CREATE TABLE siapestudiantes (  
  ESTID varchar(10) NOT NULL default '0',  
  ESTTIPDOC varchar(20) NOT NULL default "",  
  ESTNOMBRE varchar(30) NOT NULL default "",  
  ESTAPELLI varchar(30) NOT NULL default "",  
  ESTSEXO varchar(15) NOT NULL default "",  
  ESTFECNAC varchar(15) NOT NULL default "",  
  ESTTELE int(15) unsigned NOT NULL default '0',  
  ESTOBS varchar(200) NOT NULL default "",  
  ESTTIPCLI varchar(15) NOT NULL default "",  
  ESTESTA varchar(10) NOT NULL default "",  
  ESTLUGAR varchar(25) NOT NULL default "",  
  ESTDIR varchar(25) NOT NULL default "",  
  PRIMARY KEY (ESTID)  
) TYPE=MyISAM;
```

Table structure for table 'siapgrupos'

#

```
CREATE TABLE siapgrupos (  
  GRUCOD char(2) NOT NULL default '0',  
  GRUNOM varchar(20) NOT NULL default '0',  
  GRUESTA varchar(10) NOT NULL default '0',  
  PRIMARY KEY (GRUCOD)  
) TYPE=MyISAM;
```

Table structure for table 'siaplogros'

#

```
CREATE TABLE siaplogros (  
  LOGCOD char(3) NOT NULL default '0',  
  LOGDES varchar(200) NOT NULL default "",  
  LOGEST varchar(10) NOT NULL default "",  
  PENCOD char(2) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PRIMARY KEY (LOGCOD)
```

) TYPE=MyISAM;

Table structure for table 'siapmatriculas'

#

```
CREATE TABLE siapmatriculas (  
  MATNUM char(3) NOT NULL default '0',  
  ESTID varchar(10) NOT NULL default "",  
  CURCOD char(2) NOT NULL default "",  
  GRUNOM varchar(30) NOT NULL default "",  
  MATFEC date NOT NULL default '0000-00-00',  
  MATEST varchar(15) NOT NULL default "",  
  MATOBS varchar(100) NOT NULL default "",  
  MATJOR varchar(10) NOT NULL default "",  
  PRIMARY KEY (MATNUM)
```

) TYPE=MyISAM;

Table structure for table 'siappenasig'

#

```
CREATE TABLE siappenasig (  
  PENCOD char(2) NOT NULL default '0',  
  ASIGCOD char(2) NOT NULL default "",  
  PENASÑO varchar(10) NOT NULL default "",  
  PRIMARY KEY (PENCOD,ASIGCOD)
```

) TYPE=MyISAM;

Table structure for table 'siappensum'

#

```
CREATE TABLE siappensum (  
  PENCOD char(2) NOT NULL default '0',  
  PENNOM varchar(30) NOT NULL default "",  
  PENEST varchar(10) NOT NULL default "",  
  PRIMARY KEY (PENCOD)
```

) TYPE=MyISAM;

Table structure for table 'siapprofesiones'

#

```
CREATE TABLE siapprofesiones (  
  PROFCOD char(3) NOT NULL default '0',  
  PROFNOM varchar(30) NOT NULL default "",  
  PROFEST varchar(10) NOT NULL default "",  
  PRIMARY KEY (PROFCOD)  
) TYPE=MyISAM;
```

DICCIONARIO DE DATOS

Tabla Estudiantes

Nombre tabla: SIPESTUDIANTES

Descripción de la tabla:

Nombre Largo: SIPESTUDIANTES

Almacena los datos personales de los estudiantes o aspirantes que pertenezcan o que se encuentren interesados en formar parte de esta institución.

Aplicación: ACADÉMICO _
ADMINISTRATIVO

Nº	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ESTID	Primaria	Varchar	10	0	No	Identificación del estudiante
2.	ESTNOM		Varchar	30	0	No	Nombres del estudiante
3.	ESTAPELLI		Varchar	30	0	No	Apellidos del estudiante
4.	ESTDIR		Varchar	20	0	No	Dirección del estudiante
5.	ESTTEL		Varchar	10	0	No	Teléfono del estudiante
6.	ESTSEX		Varchar	10	0	No	Sexo del estudiante
7.	ESTOBS		Varchar	80	0	No	Observaciones del estudiante
8.	ESTTIPDOC		Varchar	20	0	No	Tipo de documento del estudiante
9.	ESTFECNAC		Fecha	10	0	No	Fecha de nacimiento del estudiante

Tabla Cursos

Nombre tabla:SIAPCURSOS

Nombre Largo: SIAPCURSOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla cursos es el archivo en el cual se van a mantener guardados los datos correspondientes a los cursos con que cuenta la institución por año

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	CURCOD	Primaria	Varchar	2	0	No	Codigo del curso
2.	CURNOM		Varchar	15	0	No	Nombre del curso
3.	CURESTA		Varchar	10	0	No	Estado del curso
4.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la jornada



Tabla Grupos

Nombre tabla:SIAPGRUPOS

Nombre Largo: SIAPGRUPOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla grupos es la encargada de guardar los datos de cada uno de los grupos que existen respectivamente por cada curso

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	GRUCOD	Primaria	Varchar	2	0	No	Codigo del grupo
2.	GRUNOM		Varchar	15	0	No	Nombre del grupo
3.	GRUESTA		Varchar	10	0	No	Estado del grupo

Tabla Pensum

Nombre tabla:SIAPPENSUM

Nombre Largo: SIAPPENSUM

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla siappensum posee los diferentes pensum o asignación de asignaturas a los diferentes cursos existentes en la institucion

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PENCOD	Primaria	Varchar	2	0	No	Codigo del pensun
2.	PENNOM		Varchar	15	0	No	Nombre del pensum
3.	PENESTA		Varchar	10	0	No	Estado del pensum

Tabla Asignaturas

Nombre tabla:SIAPASIGNATURAS

Nombre Largo: SIAPASIGNATURAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla cursos es el archivo en el cual se van a mantener guardados los datos correspondientes a los cursos con que cuenta la institución por año

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ASIGCOD	Primaria	Varchar	2	0	No	Codigo de la asignatura
2.	ASIGNOM		Varchar	20	0	No	Nombre de la asignatura
3.	ASIGESTA		Varchar	10	0	No	Estado de la asignatura

Tabla Areas

Nombre tabla: SIAPAREAS

Descripción: La tabla areas es la encargada de llevar las areas de trabajo academico en la institución divididas en ciencias

Nombre Largo: SIAPAREAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	AREACOD	Primaria	Varchar	2	0	No	Codigo del area
2.	AREANOM		Varchar	20	0	No	Nombre del area
3.	AREAESTA		Varchar	10	0	No	Estado del area

Tabla Logros

Nombre tabla: SIAPLOGROS

Nombre Largo: SIAPLOGROS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: En esta tabla almacenamos los diferentes logros estipulados por los docentes de la institución como medio de evaluación a los estudiantes, cada uno de ellos identificados por un único código

Nº	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	LOGCOD	Primaria	Varchar	2	0	No	Código del logro
2.	DESCRIPCION		Varchar	200	0	No	Descripción
3.	LOGESTA		Varchar	10	0	No	Estado del área
4.	ASIGCOD	Foranea	Varchar	2	0	No	Código de la asignatura
5.	PENCODE	Foranea	Varchar	2	0	No	Código del pensum

Tabla Acudientes

Nombre tabla: SIAPACUDIENES

Descripción de la tabla:

Nombre Largo: SIAPACUDIENES

La tabla acudiente es la encargada de guardar los datos personales del acudiente y su relacion con el estudiante.

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ACUID	Primaria	Varchar	10	0	No	Identificación del acudiente
2.	ACUNOM		Varchar	30	0	No	Nombres del acudiente
3.	ACUAPELLI		Varchar	30	0	No	Apellidos del acudiente
4.	ACUDIR		Varchar	20	0	No	Direccion del acudiente
5.	ACUTEL		Varchar	10	0	No	Telefono del acudiente
6.	ACUSEX		Varchar	10	0	No	Sexo del acudiente
7.	ACUPAREN		Varchar	20	0	No	Parentesco del acudiente
8.	ACUFECNAC		Fecha	20	0	No	Fecha de nacimiento del acudiente

Tabla Docentes

Nombre tabla: SIAPDOCENTES

Descripción de la tabla:

Nombre Largo: SIAPDOCENTES

En esta tabla se almacenaran los datos personales de los docentes que laboran en la institución.

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	DOCNOM		Varchar	30	0	No	Nombres del docente
3.	DOCAPELLI		Varchar	30	0	No	Apellidos del docente
4.	DOCDIR		Varchar	20	0	No	Dirección del docente
5.	DOCTEL		Varchar	10	0	No	Telefono del docente
6.	DOCSEX		Varchar	10	0	No	Sexo del docente
7.	DOCTIPDOC		Varchar	20	0	No	Tipo de documento del estudiante
8.	DOCFECNAC		Fecha	10	0	No	Fecha de nacimiento del docente
9.	DOCESTA		Varchar	10	0	No	Estado del docente
	DOCNUMESC		Varchar	2	0	No	Numero de escalafon del docente



Tabla Cursos_Grupos

Nombre tabla: SIAPCURGRU

Descripción: En esta tabla almacenamos la relacion entre los cursos y los diferentes grupos que hay en la institución y el año de creación del curso.

Nombre Largo: SIAPCURSOSGRUPOS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	CURCOD	Foranea	Varchar	2	0	No	Codigo del curso
2.	GRUCOD	Foranea	Varchar	2	0	No	Codigo del grupo
3.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la Jornada

Tabla Pensum_Asignatura

Nombre tabla: SIAPPENASIG

Descripción: En esta tabla se manejan directamente la relacion de las asignaturas con los diferentes pensum existentes por cursos

Nombre Largo: SIAPPENSUMASIGNATURA

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PENCOD	Foranea	Varchar	2	0	No	Codigo del pensum
2.	ASIGCOD	Foranea	Varchar	2	0	No	Codigo de la asignatura

Tabla Matriculas

Nombre tabla: SIAPMATRICULAS

Descripción de la tabla:

Nombre Largo: SIAPMATRICULAS

En esta tabla se guardaran los datos correspondientes a las matriculas realizadas anualmente por los estudiantes por cada curso

Aplicación: ACADÉMICO _
ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	MATNUM	Primaria	Varchar	6	0	No	Numero de matricula
2.	ESTID	Foranea	Varchar	10	0	No	Identificacion del estudiante
3.	CURCOD	Foranea	Varchar	2	0	No	Codigo del curso
4.	GRUCOD	Foranea	Varchar	10	0	No	Codigo del grupo
5.	MATFEC		Fecha		0	No	Fecha de matricula
6.	MATEST		Varchar	10	0	No	Estado de matricula
7.	MATOBS		Varchar	200	0	No	Observaciones de la matricula
8.	JORCOD	Foranea	Varchar	10	0	No	Codigo de la jornada
9.	MATAÑO		Varchar	4	0	No	Año de matricula

Tabla Profesiones

Nombre tabla: SIAPPROFESIONES

Nombre Largo: SIAPPROFESIONES

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla profesiones es la encargada de llevar las diferentes profesiones de trabajo que posee el personal que labora en la institución.

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	PROFCOD	Primaria	Varchar	2	0	No	Codigo de la profesion
2.	PROFNOM		Varchar	20	0	No	Nombre de la profesion
3.	PROFESTA		Varchar	10	0	No	Estado de la profesion

Tabla Jornadas

Nombre tabla: SIAPJORNADAS

Descripción: La tabla jornadas se encarga de llevar las jornadas de estudio en la institución.

Nombre Largo: SIAPJORNADAS

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	JORCOD	Primaria	Varchar	10	0	No	Codigo de la jornada
2.	JORNOM		Varchar	20	0	No	Nombre de la jornada
3.	JORFESTA		Varchar	10	0	No	Estado de la jornada

Tabla Logro _ estudiante

Nombre tabla: SIAPLOGEST

Nombre Largo: SIAPLOGEST

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La tabla logro _ estudiante se encarga de asignarle logros a cada estudiante.

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	LOGCOD	Primaria	Varchar	2	0	No	Codigo de la profesion
2.	ESTID		Varchar	20	0	No	Nombre de la profesion
3.	AÑO		Varchar	4	0	No	Año lectivo
4.	PERIODO		Varchar	10	0	No	Periodo de trabajo

Tabla Docente_Pensum_Asignatura

Nombre tabla: SIAPDOCPENASI

Descripción: La presente tabla registra las asignaturas de un determinado pensum dadas por un docente en la institución

Nombre Largo:
SIAPDOCENTEPENSUMASIGNATURA

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	PENCOD	Foranea	Varchar	2	0	No	Codigo del pensum
3.	ASIGCOD	Foranea	Varchar	2	0	No	Codigo de la asignatura
4.	AÑO	Foranea	Varchar	4	0	No	Año lectivo



Tabla Docente_profesion

Nombre tabla: SIAPDOCPRO

Descripción: La presente tabla registra las asignaturas de un determinado peunsum dadas por un docente en la institucion

Nombre Largo: SIAPDOCENTEPROFESION

Aplicación: ACADÉMICO _ ADMINISTRATIVO

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	DOCID	Primaria	Varchar	10	0	No	Identificación del docente
2.	PROFCOD	Foranea	Varchar	2	0	No	Codigo de la profesion

Tabla Acudiente_estudiante

Nombre tabla: SIAPACUEST

Nombre Largo: SIAPACUDIENETEESTUDIANTE

Aplicación: ACADÉMICO _ ADMINISTRATIVO

Descripción: La presente tabla registra las asignaturas de un determinado peunsum dadas por un docente en la institucion

N°	Nombre atributo	Llave	Tipo dato	long	Dec	Null	Descripción
1.	ESTID	Foranea	Varchar	10	0	No	Identificación del estudiante
2.	ACUID	Foranea	Varchar	10	0	No	Identificación del acudiente
3.	AÑO		Varchar	4	0	No	Año de asignación

DIAGRAMA ENTIDAD-RELACION

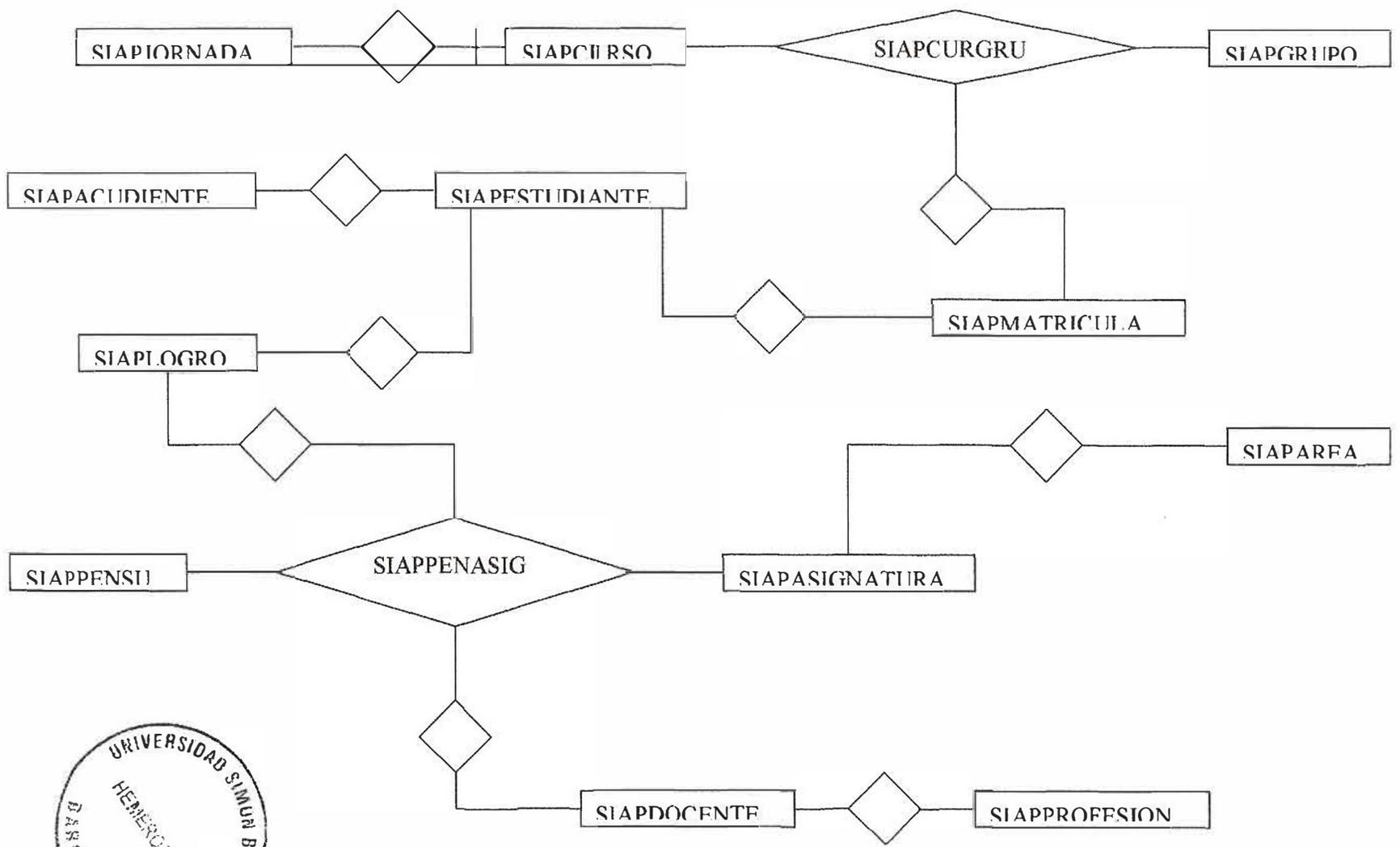


DIAGRAMA RELACIONAL

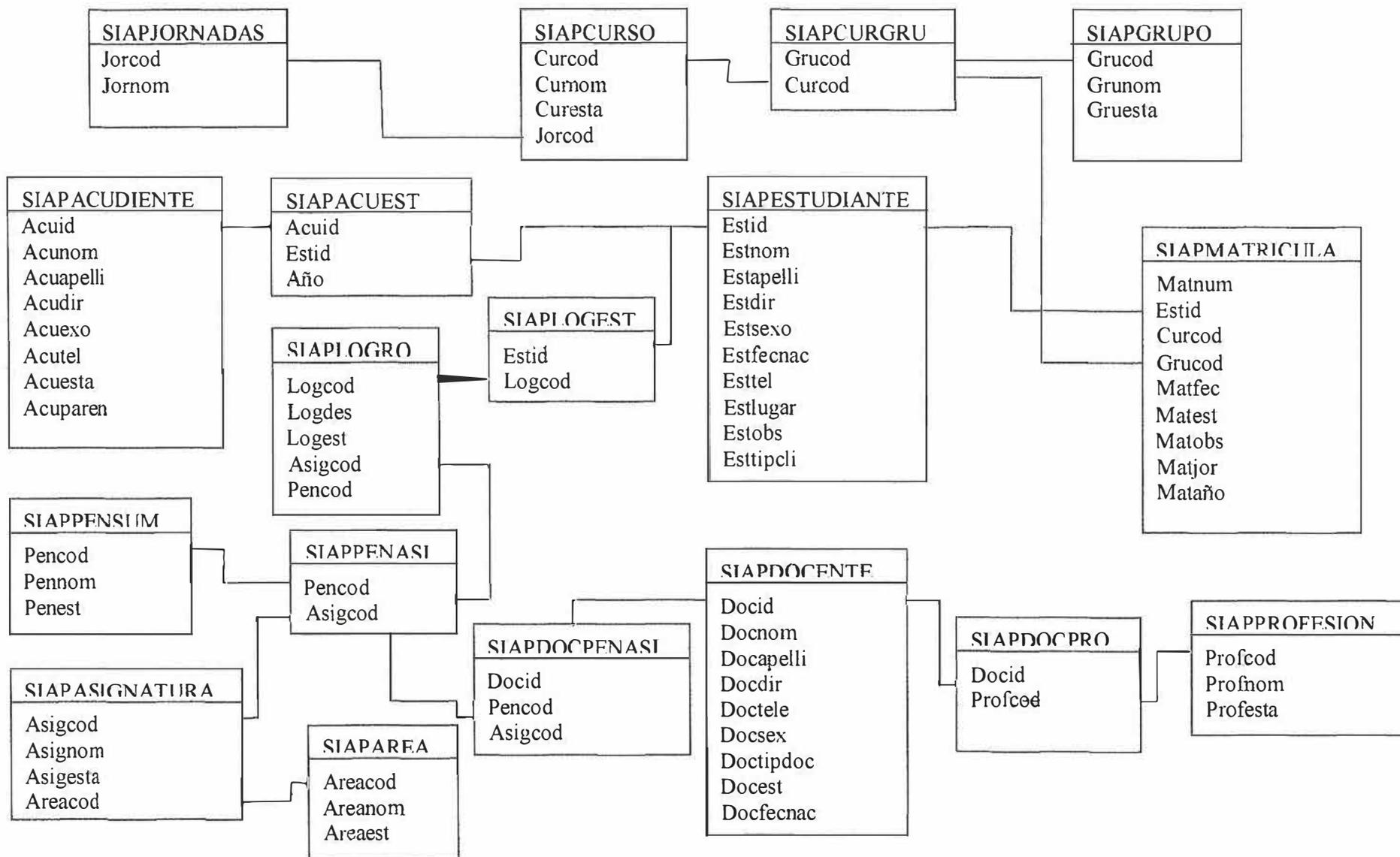


DIAGRAMA FUNCIONAL DEL SISTEMA

Sistema de Información Académico-Administrativo (Institución Educativa Distrital San Salvador)				
Datos Básicos	Procesos	Sistema	Ayuda	Salir
Administradores	Evaluaciones	Generador de Reporte		
Área	Matriculas	Auditoria		
Asignatura		IMPOR/EXPOR		
Curso	> > Creación > > Asignación			
Docente				
Grupo				
Logro				
Periodos				
Pensum	> > Creación			
Profesiones				
Estudiantes				

DIAGRAMA DE FLUJO DE DATOS DEL PROYECTO (EN DESIGNER)

Diagrama de Contexto

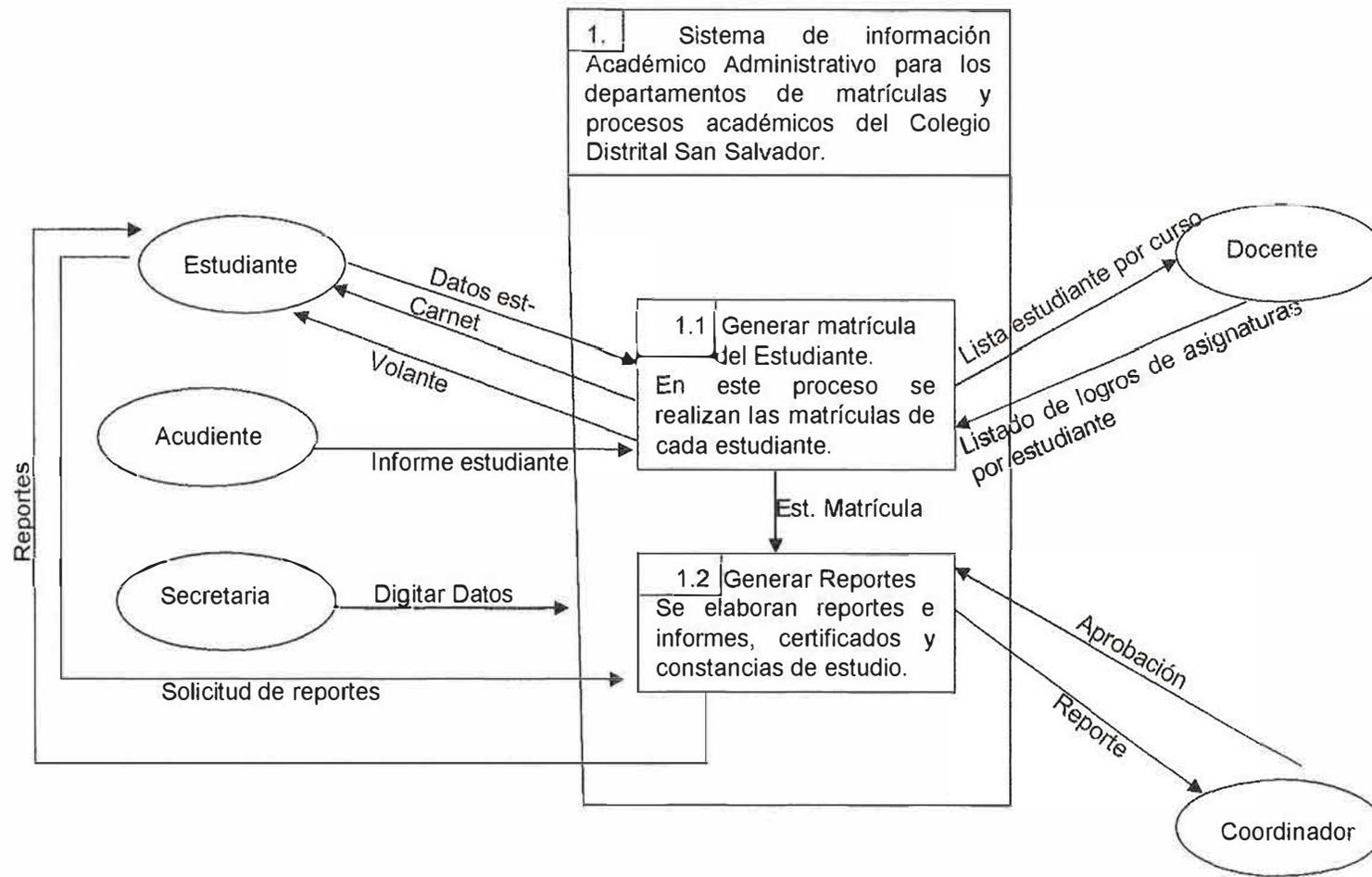


Diagrama de Nivel 1

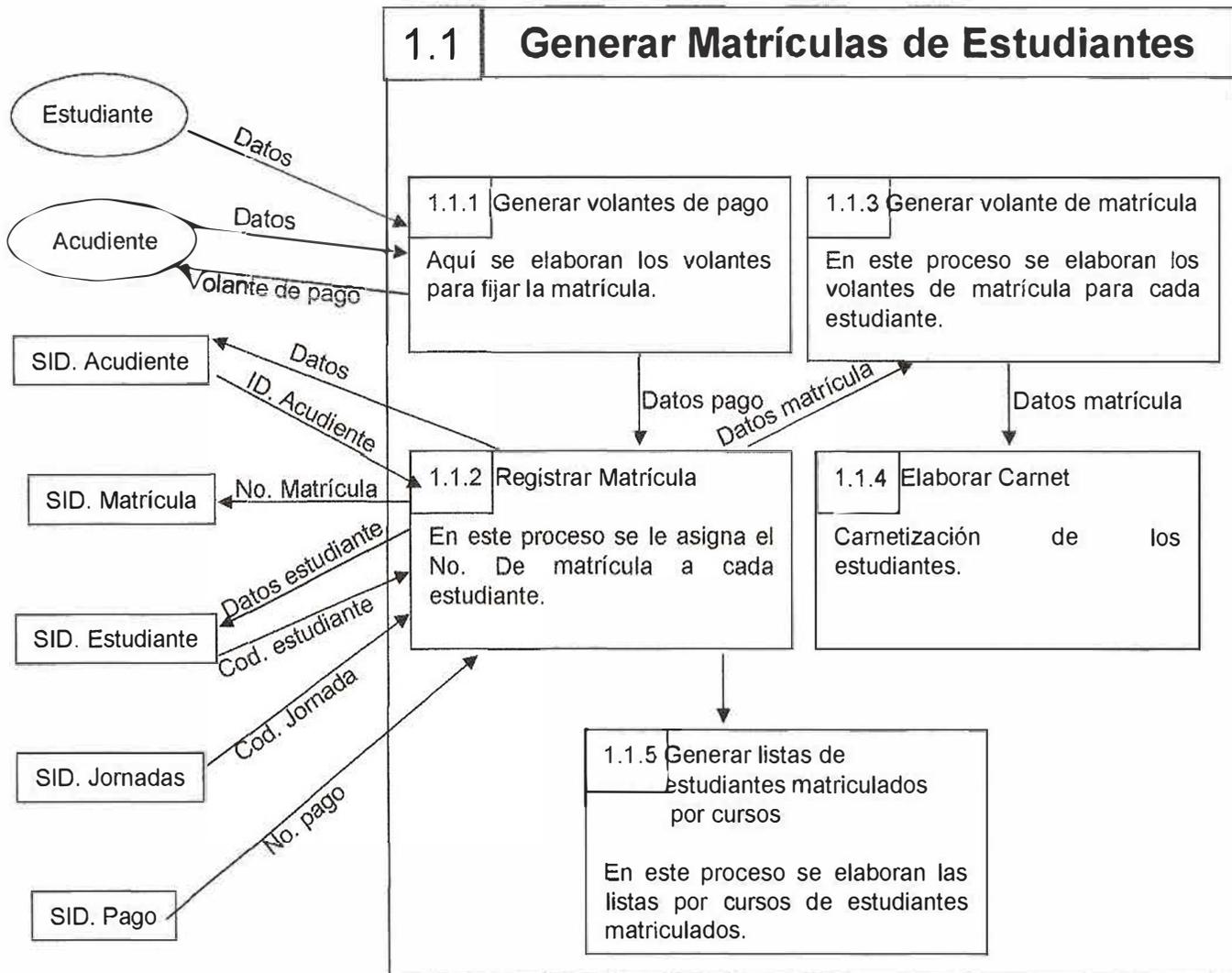
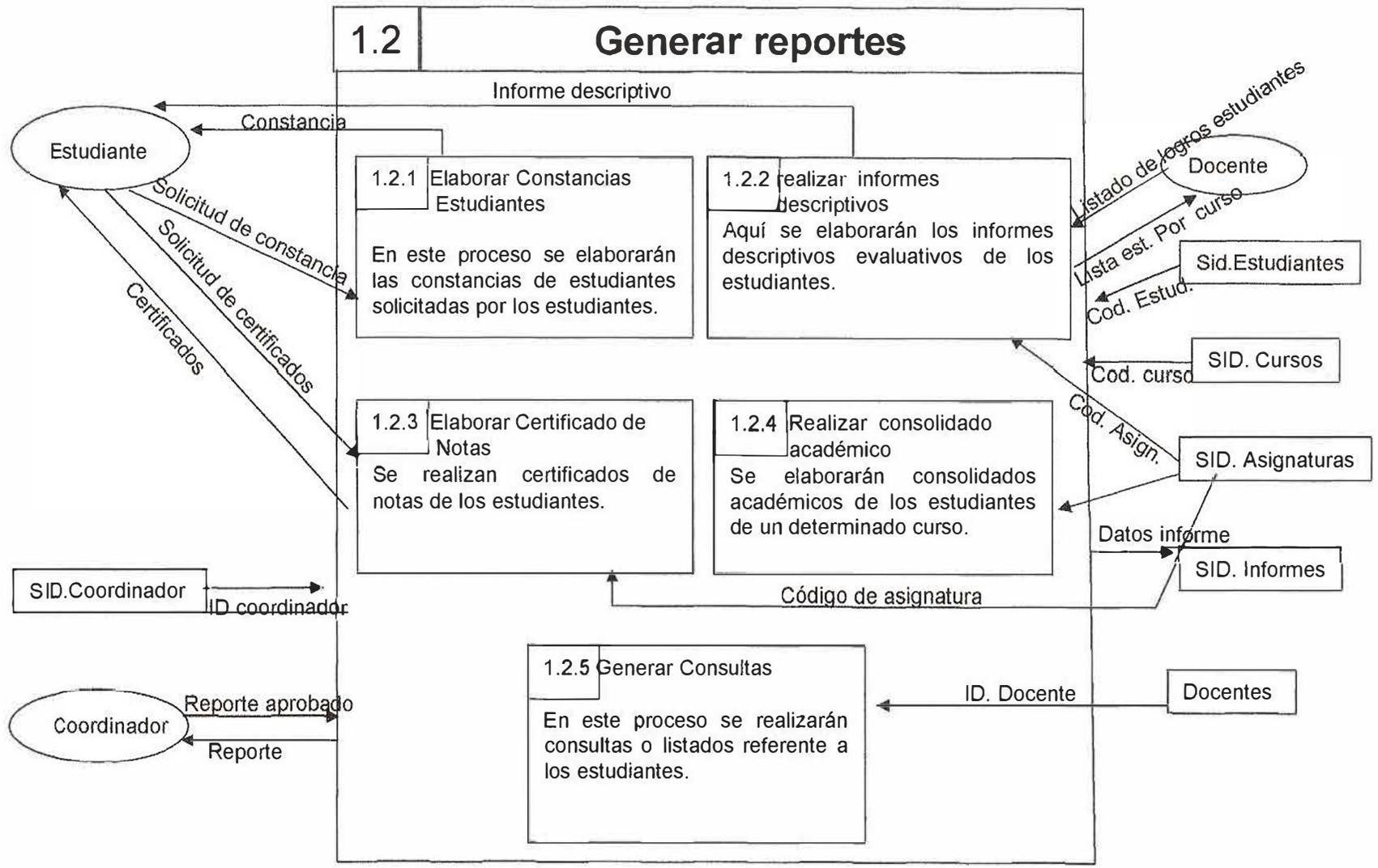


Diagrama de Nivel 2



DOCUMENTACION DE LAS PRINCIPALES RUTINAS DEL PROGRAMA CODIGO FUENTE

Cargar

'procedimiento para incrementar el componente timer de Uno en Uno

```
Private Sub Timer1_Timer()  
Label1.Caption = Label1.Caption + "|"  
Label3.Caption = Val(Label3.Caption) + 1  
If Label3.Caption = "100" Then  
Timer1.Enabled = False  
Label1.Visible = False  
Label2.Visible = False  
Label3.Visible = False  
Label4.Visible = False  
Unload Me  
ACCESO.Show  
  
End If  
End Sub
```

Acceso

'funcion validar los campos es decir que contengan datos

```
Private Function validar() As Boolean  
validar = True  
If Trim(usu.Text) = "" Then  
validar = False  
End If  
If Trim(con.Text) = "" Then  
validar = False  
End If  
  
If validar = False Then  
MsgBox "falta algun campo"  
End If  
End Function
```

'procedimiento aceptar

```
Private Sub Command1_Click()  
'conexion a la base de datos  
Set Conexion = New ADODB.Connection
```

```

Conexion.Open sansalvador.ConnectionString
If validar() Then
'seleccionar si el registro existe o no
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM SIAPADMINISTRADORES where admnom=" & usu & "" and admcon=" &
con & """)
If Not Registro.EOF Then
tadm = Registro!TIPADM
'verificar q tipo de usuario es para restringir las opciones en el menu
If (tadm = "Normal") Then
MENU.AUD.Enabled = False
MENU.admin.Enabled = False
End If
txtuser = con
txtname = usu
MENU.Show vbModal
Unload Me
Else
MsgBox "No se encuentra el usuario", vbInformation, "Defina el usuario"
End If
Registro.Close
End If
End Sub

```

```

'colocar un color al boton de aceptar
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command1.BackColor = &HFFFFFF
End Sub

```

```

'procedimiento limpiar
Private Sub Command2_Click()
jp.ForeColor = &H800000
usu = ""
con = ""
End Sub

```

```

'procedimiento salir
Private Sub Command3_Click()
Unload Me
End Sub

```

'colocar un color al boton de limpiar

```

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command2.BackColor = &HFFFFFF
End Sub

```



'colocar un color al boton de salir

```
Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
Command3.BackColor = &HFFFFFF
End Sub
```

'colocar el color original de los botones

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
Command1.BackColor = &H8000000A
Command3.BackColor = &H8000000A
Command2.BackColor = &H8000000A
End Sub
```

'procedimiento enter como aceptar

```
Private Sub con_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii = 13) And (con.Text = "") Then
```

```
MsgBox "Por favor digite su contraseña", vbCritical, "Atencion"
```

```
con.SetFocus
```

```
Else
```

```
If (KeyAscii = 13) Then
```

```
Set Conexion = New ADODB.Connection
```

```
Conexion.Open sansalvador.ConnectionString
```

```
If validar() Then
```

```
Set Registro = Conexion.Execute("SELECT" _
```

```
& "*" FROM SIAPADMINISTRADORES where admnom=" & usu & " and admcon=" & con & """)
```

```
    If Not Registro.EOF Then
```

```
        tadm = Registro!TIPADM
```

```
        If (tadm = "Normal") Then
```

```
            MENU.AUD.Enabled = False
```

```
            MENU.admin.Enabled = False
```

```
        End If
```

```
        txtuser = con
```

```
        txtname = usu
```

```
        MENU.Show vbModal
```

```
    Else
```

```
        MsgBox "No se encuentra el usuario", vbInformation, "Defina el usuario"
```

```
    End If
```

```
        Registro.Close
```

```
End If
```

```

End If
End If
End Sub
'procedimiento enter para ubicar el cursor en el textbox de contraseña o no dejar el campo
usuario en blanco
Private Sub usu_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) And (usu.Text = "") Then
MsgBox "Por favor digite nombre del usuario", vbCritical, "Atencion"
usu.SetFocus
Else
If (KeyAscii = 13) Then
con.SetFocus
End If
End If
End Sub

```

Menus

```

'procedimiento para visualizar el formulario administradores
Private Sub admin_Click()
ADMINISTRADOR.Show vbModal
End Sub
'procedimiento para visualizar el formulario areas
Private Sub area_Click()
AREAS.Show vbModal
End Sub
'procedimiento para visualizar el formulario asignacion de cursos
Private Sub asign_Click()
ASIG_CURSOS.Show vbModal
End Sub
'procedimiento para visualizar el formulario asignaturas
Private Sub Asignatura_Click()
ASIGNATURAS.Show vbModal
End Sub
'procedimiento para visualizar el formulario asignar pensum
Private Sub asina_Click()
ASIG_PEN.Show vbModal
End Sub

"procedimiento para visualizar el formulario auditoria
Private Sub AUD_Click()
If SWCLICK = 1 Then
AUDITORIA.Command2.Enabled = True
Else
AUDITORIA.Command2.Enabled = False

```

```

End If
AUDITORIA.Show vbModal

End Sub
'procedimiento para visualizar el formulario ayudas
Private Sub AYUDAS_Click()
AYUDASENLINEA.Show vbModal
End Sub
'procedimiento para visualizar el formulario cursos
Private Sub crea_Click()
CURSOS.Show vbModal
End Sub
'procedimiento para visualizar el formulario pensum
Private Sub creac_Click()
PENSUM.Show vbModal
End Sub
'procedimiento para visualizar el formulario docentes
Private Sub Docente_Click()
DOCENTES.Show vbModal
End Sub

'procedimiento para visualizar el formulario estudiantes
Private Sub estud_Click()
ESTUDIANTES.Show vbModal
End Sub
'procedimiento para visualizar el formulario valoraciones
Private Sub eval_Click()
VALORACIONES.Show vbModal
End Sub
'procedimiento para visualizar el formulario generador de reportes
Private Sub GRDs_Click()
GRD.Show vbModal
End Sub
'procedimiento para visualizar el formulario grupos
Private Sub Grupo_Click()
GRUPOS.Show vbModal
End Sub
'procedimiento para visualizar el formulario importar exportar
Private Sub IMEX_Click()
    IMPEXP.Show vbModal
End Sub
'procedimiento para visualizar el formulario logros
Private Sub log_Click()
LOGROS.Show vbModal
End Sub
'procedimiento para visualizar el formulario matriculas
Private Sub matr_Click()

```

```

MATRICULAS.Show vbModal
End Sub
'procedimiento para visualizar el formulario periodos
Private Sub per_Click()
PERIODOS.Show vbModal
End Sub
'procedimiento para visualizar el formulario profesiones
Private Sub prof_Click()
PROFESIONES.Show vbModal
End Sub
'procedimiento para visualizar el formulario salir
Private Sub Salir_Click()
Conexion.Close
End
End Sub
'procedimiento para la conexion a la base de datos
Private Sub Form_Load()
AUDITORIA.Command2.Enabled = False
Set Conexion = New ADODB.Connection
Conexion.Open sansalvador.ConnectionString
End Sub

```

Administrador

```

'procedimiento adicionar
Private Sub Adicionar()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

'CONSULTAMOS REGISTROS

Set Registro = Conexion.Execute("SELECT" _
& "*" FROM SIAPADMINISTRADORES WHERE ADMCOD" _
& "=" & cod & """)
If Registro.EOF Then
'capturar el nombre de la tabla para su auditoria
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect,
"ADMINISTRADORES", "Identificacion=" & cod)
If Option1.Value = True Then
tip.Text = "Administrador"
Else
tip.Text = "Normal"
End If

Conexion.Execute "INSERT INTO SIAPADMINISTRADORES" _
& "(ADM COD,ADM NOM,ADM CON,TIPADM) VALUES (" & cod _

```



```
& "," & nom & "," & con & "," & tip & ")"
```

```
'capturar el nombre de la tabla para su auditoria
```

```
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSInsert,  
"ADMINISTRADORES", "Identificacion=" & cod)
```

```
MsgBox "TRANSACCION REALIZADA", vbInformation, _  
"MENSAJE"
```

```
Else
```

```
MsgBox "YA EXISTE UN REGISTRO CON ESE" _  
& " NIT", vbExclamation, _  
"ATENCION"
```

```
End If
```

```
Registro.Close
```

```
End If
```

```
End Sub
```

```
Private Sub actualizar()
```

```
If validar Then
```

```
'CONSULTAMOS Registro
```

```
Set Registro = Conexion.Execute("SELECT" _  
& "* FROM SIAPADMINISTRADORES WHERE ADMCOD" _  
& "=" & cod & """)
```

```
If Not Registro.EOF Then
```

```
'capturar el nombre de la tabla para su auditoria
```

```
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect,  
"ADMINISTRADORES", "Identificacion=" & cod)
```

```
Conexion.Execute "update SIAPADMINISTRADORES set ADMNOM=" _  
& nom & "where ADMCOD=" & cod & """
```

```
Conexion.Execute "update SIAPADMINISTRADORES set ADMCON=" _  
& con & "where ADMCOD=" & cod & """
```

```
Conexion.Execute "update SIAPADMINISTRADORES set TIPADM=" _  
& tip & "where ADMCOD=" & cod & """
```

```
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSUpdate,  
"ADMINISTRADORES", "Identificacion=" & cod)
```

```
MsgBox "Registro actualizado", vbInformation, _  
"MENSAJE"
```

```
Limpiar
```

```

Else
MsgBox "No existe registro con" _
& "este NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub
Private Sub Consultar()
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM SIAPADMINISTRADORES WHERE ADMCOD" _
& "=" & cod & """)
If Not Registro.EOF Then

'capturar el nombre de la tabla para su auditoria
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect,
"ADMINISTRADORES", "Identificacion=" & cod)

nom = Registro!ADMNOM
con = Registro!ADMCON
tip = Registro!TIPADM

MsgBox "Registro consultado", vbInformation, _
"MENSAJE"
Else
MsgBox "No existe registro" _
& " con este NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End Sub
Private Sub Limpiar()
cod = ""
nom = ""
con = ""
tip = ""
Option1.Value = True
cod.SetFocus

```

```

End Sub
Private Sub ELIMINAR()
If validar Then

Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "* FROM SIAPADMINISTRADORES WHERE ADMCOD" _
& "=" & cod & """)
If Not Registro.EOF Then
'capturar el nombre de la tabla para su auditoria

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect,
"ADMINISTRADORES", "Identificacion=" & cod)

Conexion.Execute "delete from SIAPADMINISTRADORES where ADMCON=" & _
& con & "and ADMCOD=" & cod & ""
Conexion.Execute "delete from SIAPADMINISTRADORES where ADMNOM=" & _
& nom & "and ADMCOD=" & cod & ""
Conexion.Execute "delete from SIAPADMINISTRADORES where TIPADM=" & _
& tip & "and ADMCOD=" & cod & ""

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete,
"ADMINISTRADORES", "Identificacion=" & cod)

MsgBox "Registro eliminado", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "No existe este registro con este" _
& "NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub

'opciones de casos para el toolbar
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2

```

```
actualizar
Case 3
Consultar
Case 4
ELIMINAR
Case 5
Limpiar
Case 6
```

```
salir
End Select
End Sub
```

'procedimiento para consultar con enter o no dejar el codigo del administrador en blanco

```
Private Sub COD_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) And (cod.Text = "") Then
MsgBox "Deben existir datos", vbCritical, "Atencion"
cod.SetFocus
Else
If (KeyAscii = 13) Then
Consultar
End If
End If
End Sub
```

'procedimiento validar si hay datos en los campos

```
Private Function validar() As Boolean
validar = True
If Trim(cod.Text) = "" Then
validar = False
End If
If Trim(nom.Text) = "" Then
validar = False
End If
If Trim(con.Text) = "" Then
validar = False
End If
If validar = False Then
MsgBox "falta algun campo"
End If
End Function
'procedimiento para asignar el valor por defecto en administrador
Private Sub Form_Load()
Option1.Value = True
End Sub
```

```
'procedimiento salir
Private Sub salir()
Unload Me
End Sub
```

Area

```
Private Sub Adicionar()
If validar Then

l = Len(cod)
'verificar q la longitud del codigo del area
If (l <= 2) Then

'capturar el nombre de la tabla para su auditoria
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "AREAS",
"Identificacion=" & cod)
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapareas WHERE areacod" _
& "=" & cod & """)
If Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSInsert, "AREAS",
"Identificacion=" & cod)
Conexion.Execute "INSERT INTO siapareas" _
& "(areacod,areanom,areaest) VALUES (" & cod _
& ", " & nom & ", " & est & """)"
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
Else
MsgBox "Ya existe un registro con este" _
& " NIT", vbExclamation, _
"ATENCION"
End If
Registro.Close
Else
MsgBox "¡La longitud del codigo del area no es permitida!Longitud permitida 2 digitos ",
vbExclamation, _
"ATENCION"
End If
End If

End Sub
```

```
Private Sub actualizar()
If validar Then

'capturar el nombre de la tabla para su auditoria
```

```
d = ProcAuditoriaI(txtuser, txtname, vCodigoFormulario, SCSSelect, "AREAS",  
"Identificacion=" & cod)  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapareas WHERE areacod" _  
& "=" & cod & """)  
If Not Registro.EOF Then
```

```
d = ProcAuditoriaI(txtuser, txtname, vCodigoFormulario, SCSUpdate, "AREAS",  
"Identificacion=" & cod)  
Conexion.Execute "update siapareas set areanom=" _  
& nom & "where areacod=" & cod & ""  
Conexion.Execute "update siapareas set areaest=" _  
& est & "where areacod=" & cod & ""  
MsgBox "Registro actualizado correctamente", vbInformation, _  
"MENSAJE"  
Limpiar  
Else  
MsgBox "No existe registro con" _  
& "este NIT", vbExclamation, _  
"ATENCION"  
Limpiar  
End If  
Registro.Close
```

```
End If  
End Sub  
Private Sub Consultar()  
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```
Set Registro = New ADODB.Recordset  
'capturar el nombre de la tabla para su auditoria
```

```
d = ProcAuditoriaI(txtuser, txtname, vCodigoFormulario, SCSSelect, "AREAS",  
"Identificacion=" & cod)  
'CONSULTAMOS Registro  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapareas WHERE areacod" _  
& "=" & cod & """)  
If Not Registro.EOF Then  
nom = Registro!areanom  
est = Registro!areaest  
Else  
MsgBox "no existe registro" _  
& " con este NIT", vbExclamation, _  
"ATENCION"  
Limpiar
```

```

End If
Registro.Close

End Sub
Private Sub Limpiar()
cod.Enabled = True
cod = ""
nom = ""
est = ""
cod.SetFocus
End Sub
Private Sub ELIMINAR()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

Set Registro = New ADODB.Recordset
'capturar el nombre de la tabla para su auditoria

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "AREAS",
"Identificacion=" & cod)
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapareas WHERE areacod" _
& "=" & cod & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "AREAS",
"Identificacion=" & cod)
Conexion.Execute "delete from siapareas where areaest=" _
& est & ""and areacod=" & cod & ""
Conexion.Execute "delete from siapareas where areanom=" _
& nom & ""and areacod=" & cod & ""
MsgBox "Registro eliminado", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "no existe este registro" _
& "NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub
'procedimient para consultar con enter siempre y cuando exista dato en el codigo del area
Private Sub COD_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) And (cod.Text = "") Then

```

```
MsgBox "Deben existir datos", vbCritical, "Atencion"  
cod.SetFocus  
Else  
If (KeyAscii = 13) Then  
Consultar  
End If  
End If  
End Sub  
'llamado al formulario de consulta de areas  
Private Sub Command1_Click()  
MSAR.Show vbModal  
End Sub
```

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, x As Single, y  
As Single)  
Command1.BackColor = &HFFFFFFF  
End Sub
```

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As  
Single)  
Command1.BackColor = &H8000000F  
End Sub
```

'opciones de casos del toolbar

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)  
Select Case Button.Index  
Case 1  
Adicionar  
Case 2  
actualizar  
Case 3  
Consultar  
Case 4  
ELIMINAR  
Case 5  
Limpiar  
Case 6  
  
salir  
End Select  
End Sub
```

'procedimiento validar campos

```
Private Function validar() As Boolean  
validar = True
```



```
If Trim(cod.Text) = "" Then
validar = False
End If
If Trim(nom.Text) = "" Then
validar = False
End If
If Trim(est.Text) = "" Then
validar = False
End If
If validar = False Then
MsgBox "falta algun campo", 16
```

```
End If
End Function
```

```
'procedimiento salir
```

```
Private Sub salir()
Unload Me
End Sub
```

Asignaturas por curso

```
Private Sub Adicionar()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

'CONSULTAMOS REGISTROS
```

```
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcurgrujor WHERE curcod " _
& "=" & curcod & " And grucod " _
& "=" & grunom & " And jorcod " _
& "=" & Combo3 & """)
```

```
If Registro.EOF Then
Conexion.Execute "INSERT INTO siapcurgrujor" _
& "(curcod,grucod,jorcod) VALUES (" & curcod _
& "," & grunom & "," & Combo3 & """)"
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
Else
MsgBox "Ya existe un registro con este" _
& " NIT", vbExclamation, _
"ATENCIÓN"
End If
Registro.Close
```

```

End If
End Sub
Private Sub actualizar()
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcurgrujor WHERE curcod " _
& "=" & curcod & " And grucod " _
& "=" & grunom & " And jorcod " _
& "=" & Combo3 & """)

If Not Registro.EOF Then
Conexion.Execute "update siapcurgrujor set curcod=" _
& curcod & "where curcod=" & curcod & ""
Conexion.Execute "update siapcurgrujor set grucod=" _
& grunom & "where grucod=" & grunom & ""
Conexion.Execute "update siapcurgrujor set jorcod=" _
& Combo3 & "where jorcod=" & Combo3 & ""
MsgBox "Registro actualizado correctamente", vbInformation, _
"MENSAJE"

Else
MsgBox "No existe registro con" _
& "este NIT", vbExclamation, _
"ATENCION"

End If
Registro.Close
End Sub
Private Sub ELIMINAR()

```

```

'HACEMOS LA CONEXION CON LA BASE DE DATOS'

```

```

Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcurgrujor WHERE curcod " _
& "=" & curcod & " And grucod " _
& "=" & grunom & " And jorcod " _
& "=" & Combo3 & """)

If Not Registro.EOF Then

Conexion.Execute "delete from siapcurgrujor where curcod=" _
& curcod & "and curcod=" & curcod & "and grucod=" & grunom & "and jorcod=" &
Combo3 & ""
'Conexion.Execute "delete from siapcurgrujor where grucod=" _
'& grunom & "and curcod=" & curcod & "and grucod=" & grunom & "and jorcod=" &
Combo3 & ""

```

```
'Conexion.Execute "delete from siapcurgrujor where jorcod=" & _  
'& Combo3 & "'and curcod=" & curcod & "'and grucod=" & grunom & "'and jorcod=" &  
Combo3 & "'
```

```
MsgBox "Registro eliminado", vbInformation, _  
"MENSAJE"
```

```
Else
```

```
MsgBox "No existe este registro con este" & _  
& "NIT", vbExclamation, _  
"ATENCION"
```

```
End If
```

```
Registro.Close
```

```
End Sub
```

```
Private Sub Consultar()
```

```
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```
Set Registro = New ADODB.Recordset
```

```
'CONSULTAMOS Registro
```

```
Set Registro = Conexion.Execute("SELECT" & _  
& "*" FROM siapcurgrujor WHERE curcod " & _  
& "=" & curcod & "' And grucod " & _  
& "=" & grunom & "' And jorcod " & _  
& "=" & Combo3 & "'")
```

```
If Not Registro.EOF Then
```

```
Set Registro = Conexion.Execute("SELECT" & _  
& "*" FROM siapcursos")
```

```
While Not Registro.EOF
```

```
tem = Registro!curcod
```

```
If (curcod = tem) Then
```

```
nomcur = Registro!curnom
```

```
End If
```

```
Registro.MoveNext
```

```
Wend
```

```
Set Registro = Conexion.Execute("SELECT" & _  
& "*" FROM siapgrupos")
```

```
While Not Registro.EOF
```

```
TEM1 = Registro!grucod
```

```
If (grunom = TEM1) Then
```

```
nom = Registro!grunom
```

```
estg = Registro!gruesta
```

```
End If
```

```
Registro.MoveNext
```

```
Wend
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapjornadas")
While Not Registro.EOF
tem2 = Registro!jorcod
If (Combo3 = tem2) Then
nombre1 = Registro!jornombre
End If
Registro.MoveNext
Wend
```

```
Else
MsgBox "No existe registro" _
& " con este NIT", vbExclamation, _
"ATENCIÓN"
End If
Registro.Close
```

```
End Sub
```

```
Private Sub Combo3_Change()
Set Registro = New ADODB.Recordset
```

```
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapjornadas")
While Not Registro.EOF
temar2 = Registro!jorcod
If (Combo3 = temar2) Then
nomjor = Registro!jornombre
End If
Registro.MoveNext
Wend
End Sub
```

```
Private Sub Combo3_Click()
Set Registro = New ADODB.Recordset
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapjornadas")
While Not Registro.EOF
TEM1 = Registro!jorcod
If (Combo3 = TEM1) Then
nomjor = Registro!jornombre
```

```
End If
Registro.MoveNext
Wend
End Sub
```

```
Private Sub Command1_Click()
Aux1 = Combo3.Text
MSCUR.Show vbModal
sw = yes
End Sub
```

```
Private Sub Form_Load()
'Var9 = ""
'curcod = ""
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapgrupos")
While Not Registro.EOF
grunom.AddItem Registro!grucod
Registro.MoveNext
Wend
```

```
nomjor.Enabled = False
curcod.Enabled = False
nomcur.Enabled = False
estcur.Enabled = False
estg.Enabled = False
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapjornadas")
While Not Registro.EOF
Combo3.AddItem Registro!jorcod
Registro.MoveNext
Wend
nom.Enabled = False
```

```
End Sub
Private Sub curcod_CHANGE()
'If Combo3 <> "" Then
Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcursos")
While Not Registro.EOF
temcur = Registro!curcod
If (curcod = temcur) Then
```

```
nomcur = Registro!curnom
estcur = Registro!curesta
End If
Registro.MoveNext
Wend
```

```
'CONSULTAMOS el grupo por medio del curso
Set Registro = New ADODB.Recordset
Set Registro = Conexion.Execute("SELECT * FROM siapcurgrujor WHERE curcod " _
& "=" & curcod & """)
If Not Registro.EOF Then
grunom = Registro!grucod
End If
End Sub
```

```
Private Sub grunom_Click()
Set Registro = New ADODB.Recordset
```

```
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapgrupos")
While Not Registro.EOF
tem = Registro!grucod
If (grunom = tem) Then
nom = Registro!grunom
estg = Registro!gruesta
End If
Registro.MoveNext
Wend
End Sub
```

```
Private Sub Limpiar()
Combo3 = ""
curcod = ""
grunom = ""
nombre1 = ""
nomcur = ""
estcur = ""
nom = ""
estg = ""
nomjor = ""
Combo3.SetFocus
End Sub
```

```
Private Function validar() As Boolean
```



```
validar = True
If Trim(Combo3) = "" Then
validar = False
End If
If Trim(nomjor.Text) = "" Then
validar = False
End If
If Trim(curcod.Text) = "" Then
validar = False
End If
If Trim(nomcur.Text) = "" Then
validar = False
End If
If Trim(estcur.Text) = "" Then
validar = False
End If
If Trim(grunom.Text) = "" Then
validar = False
End If
If Trim(nom.Text) = "" Then
validar = False
End If
If Trim(estg.Text) = "" Then
validar = False
End If
If validar = False Then
MsgBox "falta algun campo", 16

End If
End Function
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2
'Actualizar
Case 3
Consultar
Case 4
'Eliminar
Case 5
Limpiar
Case 6
salir
End Select
End Sub
```

```
Private Sub salir()  
Unload Me  
End Sub  
asignatura
```

```
Private Sub Adicionar()  
If validar Then  
'CONSULTAMOS REGISTROS  
l = Len(cod)  
If (l <= 4) Then  
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "ASIGNATURAS",  
" - Identificacion=" & cod)
```

```
Set Registro = Conexion.Execute("SELECT"  
& "* FROM siapasignaturas WHERE ASIGCOD" _  
& "=" & cod & """)  
If Registro.EOF Then
```

```
'capturar el nombre de la tabla para su auditoria
```

```
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSInsert, "ASIGNATURAS",  
" - Identificacion=" & cod)
```

```
Conexion.Execute "INSERT INTO siapasignaturas" _  
& "(ASIGCOD,ASIGNOM,ASIGESTA,AREACOD,CURCOD) VALUES (" & cod _  
& ", " & nom & ", " & est & ", " & Combo1 & ", " & Combo2 & """)  
MsgBox "Datos registrados correctamente", vbInformation, _  
"MENSAJE"
```

```
Else  
MsgBox "Ya existe un registro con este" _  
& " NIT", vbExclamation, _  
"ATENCION"
```

```
End If
```

```
Registro.Close
```

```
Else
```

```
MsgBox "¡La longitud del codigo de la asignatura no es permitida!Longitud permitida 4  
digitos ", vbExclamation, _  
"ATENCION"
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub actualizar()
```

```
If validar Then
```

```
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```

'CONSULTAMOS Registro
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ASIGNATURAS",
" - Identificacion=" & cod)
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapasignaturas WHERE ASIGCOD" _
& "=" & cod & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSUpdate,
"ASIGNATURAS", " - Identificacion=" & cod)
Conexion.Execute "update siapasignaturas set ASIGNOM=" _
& nom & "where ASIGCOD=" & cod & ""

Conexion.Execute "update siapasignaturas set ASIGESTA=" _
& est & "where ASIGCOD=" & cod & ""

Conexion.Execute "update siapasignaturas set AREACOD=" _
& Combo1 & "where ASIGCOD=" & cod & ""

Conexion.Execute "update siapasignaturas set CURCOD=" _
& Combo2 & "where ASIGCOD=" & cod & ""

MsgBox "Registro actualizado correctamente", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "No existe registro con" _
& "este NIT", vbExclamation, _
"ATENCIÓN"
Limpiar
End If
Registro.Close

End If
End Sub

```

```

Private Sub ELIMINAR()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

```

```

Set Registro = New ADODB.Recordset

```

```

'CONSULTAMOS Registro
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ASIGNATURAS",
" - Identificacion=" & cod)

```

```

Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapasignaturas WHERE asigcod" _

```



```

& "=" & cod & "")
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "ASIGNATURAS",
" - Identificacion=" & cod)
Conexion.Execute "delete from siapasignaturas where asigcod=" & _
& cod & ""
MsgBox "Registro eliminado", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "no existe este registro" _
& "NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub

```

```
Private Sub Consultar()
```

```
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```

Set Registro = New ADODB.Recordset
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSSelect, "ASIGNATURAS",
" - Identificacion=" & cod)
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapasignaturas WHERE ASIGCOD" _
& "=" & cod & "")
If Not Registro.EOF Then
nom = Registro!asignom
est = Registro!ASIGESTA
Combo2 = Registro!curcod
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSSelect, "AREAS", " -
Identificacion=" & Combo1)
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapareas WHERE AREACOD" _
& "=" & Combo1 & "")
If Not Registro.EOF Then
nomarea = Registro!areanom
End If
Else
MsgBox "No existe registro" _
& " con este NIT", vbExclamation, _
"ATENCION"

```

```
Limpiar  
End If  
Registro.Close  
End Sub
```

```
Private Sub Limpiar()  
cod = ""  
nom = ""  
est = ""  
Combo1 = ""  
nomarea = ""  
Combo2 = ""  
nomcur = ""  
cod.SetFocus  
End Sub
```

```
Private Function validar() As Boolean  
validar = True  
If Trim(cod.Text) = "" Then  
validar = False  
End If  
If Trim(nom.Text) = "" Then  
validar = False  
End If  
If Trim(est.Text) = "" Then  
validar = False  
End If  
If Trim(Combo1) = "" Then  
validar = False  
End If  
If Trim(Combo2) = "" Then  
validar = False  
End If  
If validar = False Then  
MsgBox "falta algun campo", 16  
End If  
End Function
```

```
Private Sub salir()  
Unload Me  
End Sub
```

```
Private Sub Combo1_Change()  
Set Registro = New ADODB.Recordset
```

```
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "AREAS", " -  
Identificacion=" & Combo1)  
'CONSULTAMOS el nombre del area  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapAREAS")  
While Not Registro.EOF  
temar = Registro!areacod  
If (Combo1 = temar) Then  
nomarea = Registro!areanom  
End If  
Registro.MoveNext  
Wend  
End Sub
```

```
Private Sub Combo1_Click()
```

```
Set Registro = New ADODB.Recordset
```

```
'CONSULTAMOS Registro  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapAREAS")  
While Not Registro.EOF  
tem = Registro!areacod  
If (Combo1 = tem) Then  
nomarea = Registro!areanom  
End If  
Registro.MoveNext  
Wend  
End Sub
```

```
Private Sub Command1_Click()
```

```
con = Combo1.Text  
MSASIG.Show vbModal
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
MSCURSOS.Show vbModal  
End Sub
```

```
Private Sub Form_Load()
```

```
nomarea.Enabled = False
```



```
nomcur.Enabled = False
Combo2.Enabled = False
```

```
'llenar commbo de areas
Set Registro = New ADODB.Recordset
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapareas")
While Not Registro.EOF
Combo1.AddItem Registro!areacod
Registro.MoveNext
Wend
End Sub
```

```
Private Sub COD_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) And (cod.Text = "") Then
MsgBox "Deben existir datos", vbCritical, "Atencion"
cod.SetFocus
Else
If (KeyAscii = 13) Then
Consultar
End If
End If
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2
actualizar
Case 3
Consultar
Case 4
ELIMINAR
Case 5
Limpiar
Case 6
salir
End Select
End Sub
```

Auditoria

```
Dim i As Integer
Dim e As Integer
Dim cod As Variant
```

```

Sub Grabar()
  If Trim(txtDescripcion) = vbNullString Then
    Beep
    txtDescripcion.SetFocus
  Else
    Call CON1
    ds.Open conex
    If ChInsert.Value Then
      cmd.CommandText = "insert into SIAPFORM values('" & CmbrForma &
        "' , 'INSERTAR', '" & txtDescripcion & "', '" & CmbEstado & "')"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    Else
      cmd.CommandText = "delete from SIAPFORM where opciones='" & CmbrForma &
        "' and trans='INSERTAR'"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    End If

    If ChDelete.Value Then
      cmd.CommandText = "insert into SIAPFORM values('" & CmbrForma &
        "' , 'ELIMINAR', '" & txtDescripcion & "', '" & CmbEstado & "')"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    Else
      cmd.CommandText = "delete from SIAPFORM where opciones='" & CmbrForma &
        "' and trans='ELIMINAR'"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    End If

    If ChUpdate.Value Then
      cmd.CommandText = "insert into SIAPFORM values('" & CmbrForma &
        "' , 'MODIFICAR', '" & txtDescripcion & "', '" & CmbEstado & "')"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    Else
      cmd.CommandText = "delete from SIAPFORM where opciones='" & CmbrForma &
        "' and trans='MODIFICAR'"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    End If

    If ChSelect.Value Then
      cmd.CommandText = "insert into SIAPFORM values('" & CmbrForma &
        "' , 'CONSULTAR', '" & txtDescripcion & "', '" & CmbEstado & "')"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    Else
      cmd.CommandText = "delete from SIAPFORM where opciones='" & CmbrForma &
        "' and trans=' CONSULTAR '"
      rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    End If
  End If

```

```
ds.Close
Limpiar
End If
End Sub
```

```
Sub ELIMINAR()
On Error Resume Next
If Trim(txtCodFormulario) = vbNullString Then
    Beep
    txtide.SetFocus
Else
    Call con
    ds.Open conex
    cmd.CommandText = "delete from SIAPFORM where opciones=" & CmbrForma & ""
and trans=' CONSULTAR '"
    rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    Limpiar
    ds.Close
End If
End Sub
```

```
Sub Consultar()
Call CON1
ds.Open conex

cmd.CommandText = "select * from SIAPFORM where opciones=" & CmbrForma & ""
rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
If Not rs.EOF Then
    txtDescripcion = rs!Desc
    While Not rs.EOF
        If rs!trans = "INSERTAR" Then
            ChInsert.Value = 1
            ChInsert.Value = 2
        ElseIf rs!trans = "ELIMINAR" Then
            ChDelete.Value = 1
            ChDelete.Value = 2
        ElseIf rs!trans = "MODIFICAR" Then
            ChUpdate.Value = 1
            ChUpdate.Value = 2
        ElseIf rs!trans = "CONSULTAR" Then
            ChSelect.Value = 1
            ChSelect.Value = 2
        End If
        rs.MoveNext
    Wend
End If
```

```
ds.Close  
End Sub
```

```
Sub Limpiar()
```

```
  Select Case SSTab.Caption
```

```
    Case "Formularios":
```

```
      Me.txtDescripcion = ""  
      Me.ChInsert.Value = 0  
      Me.ChUpdate.Value = 0  
      Me.ChSelect.Value = 0  
      Me.ChDelete.Value = 0  
      Me.CmbEstado.ListIndex = 0  
      Me.CmbrForma.ListIndex = -1  
      CmbrForma.Enabled = True
```

```
    Case "Seguimiento":
```

```
      FechaIni.Visible = False  
      FechaFin.Visible = False  
      Label8.Visible = False  
      CmdTodosUsu.Visible = False  
      CmdUnUsu.Visible = False  
      CmdTodosUsu.Value = False  
      CmdUnUsu.Value = False  
      CmdUsuarios.Visible = False  
      CmdUnUsu.Top = 600  
      CmdTodosUsu.Top = 360  
      CmdUsuarios.Clear  
  
      Me.FechaIni.Clear  
      Me.FechaIni.ListIndex = -1  
      Me.FechaFin.Clear  
      Me.FechaFin.ListIndex = -1  
      Me.CmdTipoConsulta.ListIndex = -1
```

```
      CmdProcesar.Enabled = False
```

```
      InitFlex
```

```
    End Select
```

```
End Sub
```

```
Sub InitFlex()
```

```
  MSFResultados.Clear  
  MSFResultados.Cols = 7  
  MSFResultados.Rows = 2  
  MSFResultados.TextMatrix(0, 0) = "Fecha"
```

```
MSFResultados.TextMatrix(0, 1) = "Hora"  
MSFResultados.TextMatrix(0, 2) = "Usuario"  
MSFResultados.TextMatrix(0, 3) = "Formulario"  
MSFResultados.TextMatrix(0, 4) = "Transaccion"  
MSFResultados.TextMatrix(0, 5) = "Tabla"  
MSFResultados.TextMatrix(0, 6) = "Descripcion"
```

```
MSFResultados.ColWidth(0) = 1300  
MSFResultados.ColWidth(1) = 1300  
MSFResultados.ColWidth(2) = 3000  
MSFResultados.ColWidth(3) = 1500  
MSFResultados.ColWidth(4) = 3000  
MSFResultados.ColWidth(5) = 1500  
MSFResultados.ColWidth(6) = 3000
```

End Sub

```
Sub addWhere(ByRef Cad As String, ByVal Cad2 As String)  
    If (Trim(Cad) = vbNullString) Then  
        Cad = Cad2  
    Else  
        Cad = Cad + " AND " + Cad2  
    End If  
End Sub
```

```
Private Sub CmbrForma_Click()  
    Call CON1  
    ds.Open conex  
  
    cmd.CommandText = "select * from SIAPFORM where opciones=" & CmbrForma &  
    """"  
    rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic  
    If Not rs.EOF Then  
        txtDescripcion = rs!Desc  
        While Not rs.EOF  
            If rs!trans = "INSERTAR" Then  
                ChInsert.Value = 1  
                ChInsert.Value = 2  
            ElseIf rs!trans = "ELIMINAR" Then  
                ChDelete.Value = 1  
                ChDelete.Value = 2  
            ElseIf rs!trans = "MODIFICAR" Then  
                ChUpdate.Value = 1  
                ChUpdate.Value = 2  
            ElseIf rs!trans = "CONSULTAR" Then
```

```

        ChSelect.Value = 1
        ChSelect.Value = 2
    End If
    rs.MoveNext
Wend
    CmbrForma.Enabled = False
End If
ds.Close
End Sub

```

```

Private Sub CmdProcesar_Click()

```

```

    Dim vSql As String
    Dim strWhere As String
    Dim fec

```

```

    CmdProcesar.Enabled = False

```

```

    InitFlex

```

```

    strWhere = ""

```

```

    vSql = " select * from SIAPAUDITORIAS, SIAPADMINISTRADORES "

```

```

    If CmdTipoConsulta = "FECHA" Then

```

```

        If (FechaIni <= FechaFin) Then

```

```

            addWhere strWhere, " FEC_TRANS between " & Year(FechaIni) & "-" &
Month(FechaIni) & "-" & Day(FechaIni) & " and " & Year(FechaFin) & "-" &
Month(FechaFin) & "-" & Day(FechaFin) & ""

```

```

        Else

```

```

            MsgBox "LA FECHA INICIAL DEBE SER MENOR O IGUAL A LA FECHA
FINAL"

```

```

            Exit Sub

```

```

        End If

```

```

    End If

```

```

    If CmdTipoConsulta = "USUARIO" Then

```

```

        If (CmdUnUsu.Value = True) Then

```

```

            addWhere strWhere, " usuario=" & CmdUsuarios & ""

```

```

        End If

```

```

        If (CmdTodosUsu.Value = True) Then

```

```

            vSql = vSql & " where ADMCOD=usuario order by fec_trans, Horas"

```

```

        End If

```

```

    End If

```

```

    If Trim(strWhere) <> vbNullString Then

```

```

        vSql = vSql & " where ADMCOD=usuario and " & strWhere & " GROUP BY
fec_trans,horas"

```

```

End If

Call CON1
ds.Open conex

Set rs = ds.Execute(vSql)

While Not rs.EOF
    Horas = rs!Horas
    cod = Split(Horas, " ")
    Horas = cod(1)
    MSFResultados.AddItem ""
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 0) = rs!FEC_TRANS
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 1) = Horas
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 2) = rs!Usuario & " - " &
rs!ADMNOM
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 3) = rs!Tabla
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 5) = rs!Tabla
    MSFResultados.TextMatrix(MSFResultados.Rows - 2, 6) = rs!descripcion

    Select Case rs!Tip_trans
        Case "INSERTAR": MSFResultados.TextMatrix(MSFResultados.Rows - 2, 4) =
"Ingreso de registro"
        Case "ELIMINAR": MSFResultados.TextMatrix(MSFResultados.Rows - 2, 4) =
"Eliminacion de registro"
        Case "CONSULTAR": MSFResultados.TextMatrix(MSFResultados.Rows - 2, 4) =
"Busqueda de registro"
        Case "MODIFICAR": MSFResultados.TextMatrix(MSFResultados.Rows - 2, 4) =
"Actualizacion de registro"
    End Select

    rs.MoveNext
Wend
rs.Close
ds.Close
End Sub

Private Sub CmdTipoConsulta_Click()
Dim fec
Call CON1
ds.Open conex
If CmdTipoConsulta = "USUARIO" Then
    FechaIni.Visible = False
    FechaFin.Visible = False
    Label8.Visible = False
    CmdTodosUsu.Visible = True
    CmdUnUsu.Visible = True

```



```

    CmdTodosUsu.Value = False
    CmdUnUsu.Value = False
    CmdUsuarios.Visible = False
    CmdUnUsu.Top = 600
    CmdTodosUsu.Top = 360
ElseIf CmdTipoConsulta = "FECHA" Then
    CmdTodosUsu.Visible = False
    CmdUnUsu.Visible = False
    CmdUsuarios.Visible = False
    FechaIni.Visible = True
    FechaFin.Visible = True
    Label8.Visible = True
    Set rs = ds.Execute("select fec_trans from SIAPAUDITORIAS group by fec_trans ")
    While Not rs.EOF
        fec = rs!FEC_TRANS
        FechaIni.AddItem FormatFecha2(fec)
        FechaFin.AddItem FormatFecha2(fec)
        rs.MoveNext
    Wend
End If
ds.Close
End Sub

```

```

Private Sub CmdTodosUsu_Click()
    CmdUsuarios.Visible = False
    CmdUnUsu.Visible = False
    CmdTodosUsu.Top = 480
    CmdProcesar.Enabled = True
End Sub

```

```

Private Sub CmdUnUsu_Click()
    Call CON1
    ds.Open conex
    cmd.CommandText = "select * from SIAPAUDITORIAS group by usuario"
    rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
    CmdUsuarios.Visible = True
    CmdTodosUsu.Visible = False
    CmdUnUsu.Top = 480
    While Not rs.EOF
        CmdUsuarios.AddItem rs!Usuario
        rs.MoveNext
    Wend
    ds.Close
End Sub

```

```

Private Sub CmdUsuarios_click()
    CmdProcesar.Enabled = True

```

End Sub

Private Sub Combo1_Click()

Call CON1

ds.Open conex

cmd.CommandText = "select * from SIAPFORM where opciones=" & CmbForma & ""

rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic

If Not rs.EOF Then

txtDescripcion = rs!Desc

While Not rs.EOF

If rs!trans = "INSERTAR" Then

ChInsert.Value = 1

ChInsert.Value = 2

ElseIf rs!trans = "ELIMINAR" Then

ChDelete.Value = 1

ChDelete.Value = 2

ElseIf rs!trans = "MODIFICAR" Then

ChUpdate.Value = 1

ChUpdate.Value = 2

ElseIf rs!trans = "CONSULTAR" Then

ChSelect.Value = 1

ChSelect.Value = 2

End If

rs.MoveNext

Wend

CmbForma.Enabled = False

End If

ds.Close

End Sub

Private Sub Command1_Click()

Command2.Enabled = False

Unload Me

End Sub

Private Sub Command2_Click()

Set Registro = New ADODB.Recordset

Set Registro = Conexion.Execute("SELECT" & "* FROM siapauditorias")

If Not Registro.EOF Then

Conexion.Execute "delete from siapauditorias"

End If

Command2.Enabled = False

SWCLICK = ●

End Sub

Private Sub FechaFin_Click()

If (FechaIni <= FechaFin) Then
 CmdProcesar.Enabled = True

Else

 MsgBox "LA FECHA INICIAL DEBE SER MENOR O IGUAL A LA FECHA
FINAL"

 FechaIni.Clear

 FechaIni.ListIndex = -1

 FechaFin.Clear

 FechaFin.ListIndex = -1

 CmdTipoConsulta.ListIndex = -1

 CmdProcesar.Enabled = False

 FechaIni.SetFocus

End If

End Sub

Private Sub FechaFin_KeyDown(KEYCODE As Integer, Shift As Integer)

If KEYCODE = 13 Then

 CmbTipo.SetFocus

End If

End Sub

Private Sub FechaIni_KeyDown(KEYCODE As Integer, Shift As Integer)

If KEYCODE = 13 Then

 FechaFin.SetFocus

End If

End Sub

Private Sub Form_Load()

 CmbEstado.AddItem "ACTIVO"

 CmbEstado.AddItem "INACTIVO"

 CmbEstado.ListIndex = 0

 CmdTipoConsulta.AddItem "USUARIO"

 CmdTipoConsulta.AddItem "FECHA"

 CmdTipoConsulta.ListIndex = -1

 CmbrForma.AddItem "ADMINISTRADORES"

 CmbrForma.AddItem "AREAS"

 CmbrForma.AddItem "ASIG_CURSOS"

 CmbrForma.AddItem "ASIGNATURAS"

 CmbrForma.AddItem "CURSOS"



```
CmbrForma.AddItem "DOCENTES"  
CmbrForma.AddItem "ESTUDIANTES"  
CmbrForma.AddItem "EVALUACIONES"  
CmbrForma.AddItem "GRUPOS"  
CmbrForma.AddItem "LOGROS"  
CmbrForma.AddItem "MATRICULAS"  
CmbrForma.AddItem "PENSUM"  
CmbrForma.AddItem "PROFESIONES"  
CmbrForma.ListIndex = -1
```

```
InitFlex  
SSTab_Click 0
```

```
End Sub
```

```
Private Sub imprimir_Click()  
limpiar_reporauditoria  
Dim j, sw As Integer  
If MSFResultados <> "" Then  
For j = 1 To MSFResultados.Rows - 2  
MSFResultados.Row = j  
MSFResultados.COL = 0  
fec = MSFResultados.Text  
MSFResultados.COL = 1  
hor = MSFResultados.Text  
MSFResultados.COL = 2  
usu = MSFResultados.Text  
MSFResultados.COL = 3  
formu = MSFResultados.Text  
MSFResultados.COL = 4  
trans = MSFResultados.Text  
MSFResultados.COL = 5  
Tabla = MSFResultados.Text  
MSFResultados.COL = 6  
Desc = MSFResultados.Text  
  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapreporauditoria")  
  
'If Registro.EOF Then  
Conexion.Execute "INSERT INTO siapreporauditoria" _  
& "(fecaud,horaud,usuaud,foraud,traaud,tabaud,desaud) VALUES (" & fec _  
& "," & hor & "," & usu & "," & formu & "," & trans & "," & Tabla & "," & Desc &  
")"  
sw = 0  
'End If
```

```

Next j
If sw = 0 Then
MsgBox "Reporte elaborado correctamente", vbInformation, _
"MENSAJE"
'imprimir auditoria
abccadena = ("select fecaud,horaud,usuaud,foraud,traud,tabaud,desaud from
siapreporauditoria")
MsgBox abccadena
Set Registro = Conexion.Execute(abccadena)
If Not Registro.EOF Then
Set DataRauditoria.DataSource = Registro
DataRauditoria.Show vbModal
End If
End If
Else
MsgBox "Debe existir datos " _
& "en la auditoria", vbExclamation, _
"ATENCION"
End If
End Sub
Private Sub limpiar_reporauditoria()
Set Registro = New ADODB.Recordset

Set Registro = Conexion.Execute("SELECT" _
& "* FROM siapreporauditoria")
If Not Registro.EOF Then
Conexion.Execute "delete from siapreporauditoria"
End If

End Sub
Private Sub SSTab_Click(PreviousTab As Integer)
Select Case SSTab.Caption
Case "Formularios":
Toolbar1.Buttons(1).Enabled = True
Toolbar1.Buttons(2).Enabled = True
Toolbar1.Buttons(3).Enabled = True
Case "Seguimiento":
Toolbar1.Buttons(1).Enabled = False
Toolbar1.Buttons(2).Enabled = False
Toolbar1.Buttons(3).Enabled = False
End Select
End Sub

Private Sub txtCodForm_GotFocus()
With txtCodForm
.SelStart = 0
.SelLength = Len(txtCodForm)

```

```
End With  
End Sub
```

```
Private Sub txtCodForm_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    FechaIni.SetFocus  
End If  
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)  
    Select Case Button.Index  
        Case 1: Grabar  
        Case 2: ELIMINAR  
        Case 3: Grabar  
        Case 4:  
            If SSTab.Caption = "Formularios" Then  
                Consultar  
            Else  
                CmdProcesar_Click  
            End If  
        Case 5: Limpiar  
        'Case 6: imprimir  
        Case 7: Unload Me  
    End Select  
End Sub
```

```
Private Sub txtDescripcion_GotFocus()  
With txtDescripcion  
    .SelStart = 0  
    .SelLength = Len(txtDescripcion)  
End With  
End Sub
```

```
Private Sub txtIde_GotFocus()  
With txtide  
    .SelStart = 0  
    .SelLength = Len(txtide)  
End With  
End Sub
```

```
Private Sub txtIde_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
    txtCodForm.SetFocus  
End If  
End Sub
```

Ayudas en linea

```
Private m_hwndTV As Long
```

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim Node1 As Node
```

```
Dim Node2 As Node
```

```
Dim Node3 As Node
```

```
Dim Node4 As Node
```

```
Dim Node5 As Node
```

```
Dim Node6 As Node
```

```
Dim Node7 As Node
```

```
TreeView1.HideSelection = False
```

```
TreeView1.LabelEdit = tvwManual
```

```
TreeView1.LineStyle = tvwRootLines
```

```
m_hwndTV = TreeView1.hWnd
```

```
Set Node1 = TreeView1.Nodes.Add(, , "ACCESO", 1)
```

```
Set Node2 = TreeView1.Nodes.Add(, , "ADMINISTRADOR", 1)
```

```
Set Node3 = TreeView1.Nodes.Add(, , "AREAS", 1)
```

```
Set Node4 = TreeView1.Nodes.Add(, , "ASIGNATURAS X CURSOS", 1)
```

```
Set Node5 = TreeView1.Nodes.Add(, , "ASIGNATURAS", 1)
```

```
Set Node6 = TreeView1.Nodes.Add(, , "AUDITORIA", 1)
```

```
Set Node7 = TreeView1.Nodes.Add(, , "CURSOS", 1)
```

```
Set Node1 = TreeView1.Nodes.Add(Node1, tvwChild, , "Acceso", 2)
```

```
Set Node2 = TreeView1.Nodes.Add(Node2, tvwChild, , "Administrador", 2)
```

```
Set Node3 = TreeView1.Nodes.Add(Node3, tvwChild, , "Areas", 2)
```

```
Set Node4 = TreeView1.Nodes.Add(Node4, tvwChild, , "Asignaturas x cursos", 2)
```

```
Set Node5 = TreeView1.Nodes.Add(Node5, tvwChild, , "Asignaturas", 2)
```

```
Set Node6 = TreeView1.Nodes.Add(Node6, tvwChild, , "Auditoria", 2)
```

```
Set Node7 = TreeView1.Nodes.Add(Node7, tvwChild, , "cursos", 2)
```

```
Call ShowSelectedNode(TreeView1.Nodes(1), False)
```

```
End Sub
```

```
Private Sub ShowSelectedNode(sNode As String, flsChecked As Boolean)
```

```
Label1 = sNode
```

```
If Label1 = "Acceso" Then
```

```
'MsgBox ("hola acceso")
```

```
Picture1.Width = 5940
```

```
Picture1.Height = 3490
```

```
'App.Path + "\imagenes\Acceso01.bmp"
```

```
Picture1.Picture = LoadPicture("ACCESO.bmp")
```

```
End If
```

```
If Label1 = "Administrador" Then
```

```
'MsgBox ("hola administrador")
```

```
Picture1.Width = 6380
```

```
Picture1.Height = 4680
```

```
Picture1.Picture = LoadPicture("ADMINISTRADOR.bmp")
```

```
End If
```

```
If Label1 = "Areas" Then
```

```
'MsgBox ("hola areas")
```

```
Picture1.Width = 5150
```

```
Picture1.Height = 4760
```

```
Picture1.Picture = LoadPicture("AREAS.bmp")
```

```
End If
```

```
If Label1 = "Asignaturas x cursos" Then
```

```
MsgBox ("hola asignaturas x cursos")
```

```
End If
```

```
If Label1 = "Asignaturas" Then
```

```
'MsgBox ("hola asignaturas")
```

```
Picture1.Width = 5250
```

```
Picture1.Height = 6350
```

```
Picture1.Picture = LoadPicture("ASIGNATURAS.bmp")
```

```
End If
```

```
If Label1 = "Auditoria" Then
```

```
MsgBox ("hola auditoria")
```

```
End If
```

```
End Sub
```

```
Private Sub TreeView1_NodeClick(ByVal Node As ComctlLib.Node)
```

```
Dim fChecked As Boolean
```

```
Dim hItem As Long
```

```
Dim pt As POINTAPI
```

If GetAsyncKeyState(vbKeyLButton) Then

Call GetCursorPos(pt)

Call ScreenToClient(m_hwndTV, pt)

fChecked = IsTVItemCheckedFromClick(m_hwndTV, pt.x, pt.y)

Call ShowSelectedNode(Node.Text, fChecked)

End If

End Sub

Curso

Private Sub Adicionar()

If validar Then

'HACEMOS LA CONEXION CON LA BASE DE DATOS'

'CONSULTAMOS REGISTROS'

l = Len(cod)

If (l <= 2) Then

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "CURSOS", " -
Identificacion=" & cod)

Set Registro = Conexion.Execute("SELECT " _

& "*" FROM siapcursos WHERE curcod" _

& "=" & cod & " AND jorcod" _

& "=" & Combo1 & """)

If Registro.EOF Then

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSInsert, "CURSOS", " -
Identificacion=" & cod)

Conexion.Execute "INSERT INTO siapcursos" _

& "(curcod,curnom,curesta,jorcod) VALUES (" & cod _

& ", " & nom & ", " & est & ", " & Combo1 & """)

MsgBox "Datos registrados correctamente", vbInformation, _

"MENSAJE"

Else

MsgBox "Ya existe un registro con este" _

& " NIT", vbExclamation, _

"ATENCION"

End If

Else

MsgBox "¡La longitud del codigo del curso no es permitida!Longitud permitida 2 digitos ",
vbExclamation, _

"ATENCION"

```

End If
Registro.Close
End If
End Sub
Private Sub actualizar()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "CURSOS", " -
Identificacion=" & cod)
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcursos WHERE curcod" _
& "=" & cod & " AND jorcod" _
& "=" & Combo1 & "")
If Not Registro.EOF Then
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSUpdate, "CURSOS", " -
Identificacion=" & cod)
Conexion.Execute "update siapcursos set curnom=" _
& nom & "where curcod=" & cod & " and jorcod=" & Combo1 & ""
Conexion.Execute "update siapcursos set curesta=" _
& est & "where curcod=" & cod & " and jorcod=" & Combo1 & ""
MsgBox "Registro actualizado correctamente", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "No existe registro con" _
& "este NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub
Private Sub Consultar()
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapcursos WHERE curcod" _
& "=" & cod & " AND JORCOD" _
& "=" & Combo1 & "")
If Not Registro.EOF Then

```

```

nom = Registro!curnom
est = Registro!curesta
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "CURSOS", " -
Identificacion=" & cod)
Else
MsgBox "No existe registro" _
& " con este NIT", vbExclamation, _
"ATENCIÓN"
Limpiar
End If
Registro.Close

```

```
End Sub
```

```
Private Sub Limpiar()
```

```
cod.Enabled = True
```

```
cod = ""
```

```
nom = ""
```

```
est = ""
```

```
Combo1 = ""
```

```
nomjor = ""
```

```
Combo1.SetFocus
```

```
End Sub
```

```
Private Sub ELIMINAR()
```

```
If validar Then
```

```
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```
Set Registro = New ADODB.Recordset
```

```
'CONSULTAMOS Registro
```

```
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "CURSOS", " -
Identificacion=" & cod)
```

```
Set Registro = Conexion.Execute("SELECT" _
```

```
& "*" FROM siapcursos WHERE curcod" _
```

```
& "=" & cod & " and jorcod" _
```

```
& "=" & Combo1 & """)
```

```
If Not Registro.EOF Then
```

```
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "CURSOS", " -
Identificacion=" & cod)
```

```
Conexion.Execute "delete from siapcursos where curesta=" _
```

```
& est & "and curcod=" & cod & " and jorcod=" & Combo1 & ""
```

```
Conexion.Execute "delete from siapcursos where curnom=" _
```

```
& nom & "and curcod=" & cod & " and jorcod=" & Combo1 & ""
```

```
MsgBox "Registro eliminado", vbInformation, _
```

```
"MENSAJE"
```

```
Limpiar
```

```
Else
```

```
MsgBox "No existe este registro con este" _
```

```
& "NIT", vbExclamation, _  
"ATENCION"  
Limpiar  
End If  
Registro.Close
```

```
End If  
End Sub
```

```
Private Sub Combo1_Change()
```

```
Set Registro = New ADODB.Recordset  
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "JORNADAS", " -  
Identificacion=" & cod)  
'CONSULTAMOS Registro  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapjornadas")  
While Not Registro.EOF  
temar1 = Registro!jorcod  
If (Combo1 = temar1) Then  
nomjor = Registro!jornombre  
End If  
Registro.MoveNext  
Wend
```

```
End Sub
```

```
Private Sub Combo1_Click()
```

```
Set Registro = New ADODB.Recordset  
  
'CONSULTAMOS Registro  
Set Registro = Conexion.Execute("SELECT" _  
& "*" FROM siapjornadas")  
While Not Registro.EOF  
tem3 = Registro!jorcod  
If (Combo1 = tem3) Then  
nomjor = Registro!jornombre  
End If  
Registro.MoveNext  
Wend  
End Sub
```

```
Private Sub Command1_Click()
```

```
Aux1 = Combo1.Text  
swjp = 1
```

```
MSCUR.Show vbModal
sw = yes
End Sub
```

```
Private Sub Form_Load()
nomjor.Enabled = False
Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapjornadas")
While Not Registro.EOF
Combo1.AddItem Registro!jorcod
Registro.MoveNext
Wend
```

```
End Sub
```

```
Private Sub COD_KeyPress(KeyAscii As Integer)
If (KeyAscii = 13) And (cod.Text = "") Then
MsgBox "Deben existir datos", vbCritical, "Atencion"
cod.SetFocus
Else
If (KeyAscii = 13) Then
Consultar
End If
End If
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2
actualizar
Case 3
Consultar
Case 4
ELIMINAR
Case 5
Limpiar
Case 6
salir
```

```
End Select
End Sub
```

```
Private Function validar() As Boolean
validar = True
If Trim(Combo1) = "" Then
validar = False
End If
If Trim(nomjor.Text) = "" Then
validar = False
End If
If Trim(cod.Text) = "" Then
validar = False
End If
If Trim(nom.Text) = "" Then
validar = False
End If
If Trim(est.Text) = "" Then
validar = False
End If
If validar = False Then
MsgBox "falta algun campo", 16
End If
End Function
```

```
Private Sub salir()
Unload Me
swjp = 0
End Sub
```

Docentes

```
Private Sub Adicionar()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

'CONSULTAMOS REGISTROS
l = Len(id)
If (l <= 10) Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "DOCENTES", " -
Identificacion=" & id)
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapdocentes WHERE DOCID" _
& "=" & id & """)
If Registro.EOF Then
```



```

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSInsert, "DOCENTES", " -
Identificacion=" & id)
Conexion.Execute "INSERT INTO siapdocentes" _
&
"(DOCID,DOCNOM,DOCAPELLI,DOCNUMESC,DOCSEXO,DOCFECNAC,DOCDIR,
DOCTELE,DOCTIPDOC,DOCESTA) VALUES (" & id _
& "," & nom & "," & ape & "," & numesc & "," & sex & "," & fec & "," & dir & ","
& tel & "," & tipdoc & "," & est & " )"
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
Else
MsgBox "Ya existe un registro con este" _
& " NIT", vbExclamation, _
"ATENCION"
End If
Registro.Close
Else
MsgBox "¡La longitud del código del docente no es permitida! Longitud permitida 10
dígitos ", vbExclamation, _
"ATENCION"
End If
End If
End Sub
Private Sub actualizar()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

```

```

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "DOCENTES", " -
Identificacion=" & id)
'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapdocentes WHERE DOCID" _
& "=" & id & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSUpdate, "DOCENTES", " -
Identificacion=" & id)
Conexion.Execute "update siapdocentes set DOCNOM=" _
& nom & "where DOCID=" & id & ""
Conexion.Execute "update siapdocentes set DOCAPELLI=" _
& ape & "where DOCID=" & id & ""
Conexion.Execute "update siapdocentes set DOCNUMESC=" _
& numesc & "where DOCID=" & id & ""
Conexion.Execute "update siapdocentes set DOCSEXO=" _
& sex & "where DOCID=" & id & ""
Conexion.Execute "update siapdocentes set DOCFECNAC=" _
& fec & "where DOCID=" & id & ""

```

```

Conexion.Execute "update siapdocentes set DOCDIR="" _
& dir & "where DOCID="" & id & ""
Conexion.Execute "update siapdocentes set DOCTELE="" _
& tel & "where DOCID="" & id & ""
Conexion.Execute "update siapdocentes set DOCTIPDOC="" _
& tipdoc & "where DOCID="" & id & ""
Conexion.Execute "update siapdocentes set DOCESTA="" _
& est & "where DOCID="" & id & ""
MsgBox "Registro actualizado correctamente", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "No existe registro con" _
& "este NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub

```

```

Private Sub Command1_Click()
MSASIGNATURAS.Show vbModal
End Sub

```

```

Private Sub Command2_Click()
Dim j, sw As Integer
sw = 0
MSdoc.Row = 1
MSdoc.COL = 0
If MSdoc.Text = "" Then
MsgBox "Debe seleccionar " _
& "una asignatura", vbExclamation, _
"ATENCION"
Else
If id.Text <> "" Then
For j = 1 To MSdoc.Rows - 2
MSdoc.Row = j
MSdoc.COL = 0
idasig = MSdoc.Text
Set Registro = Conexion.Execute("SELECT" _
& "* FROM siapdocasig WHERE docid " _
& "=" & id & " And asigcod " _
& "=" & idasig & "")

```

```

If Registro.EOF Then
Conexion.Execute "INSERT INTO siapdocasig" _
& "(docid,asigcod) VALUES (" & id _
& ", " & idasig & ")"
sw = 0
End If
Registro.Close
Next j
If sw = 0 Then
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
End If
Else
MsgBox "Debe existir el Id " _
& "del docente", vbExclamation, _
"ATENCIÓN"
End If
End If
End Sub

```

```

Private Sub Command3_Click()
Set Registro = New ADODB.Recordset

d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "DOCENTES", " -
Identificacion=" & id)
Set Registro = Conexion.Execute("SELECT" _
& "* FROM siapdocasig WHERE asigcod" _
& "=" & dateli & """)
If Not Registro.EOF Then
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSDelete, "DOCENTES", " -
Identificacion=" & id)
Conexion.Execute "delete from siapdocasig where asigcod=" & _
& dateli & "and asigcod=" & dateli & ""
MsgBox "Asignatura eliminado", vbInformation, _
"MENSAJE"
End If
Registro.Close

End Sub

```

```

Private Sub Command4_Click()
MSDOCENTES.Show vbModal
End Sub

```

```
Private Sub Frame2_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
Command1.BackColor = &H8000000A
Command3.BackColor = &H8000000A
Command2.BackColor = &H8000000A
End Sub
```

```
Private Sub MSdoc_Click()
MSdoc.COL = 0
dateli = MSdoc.Text
End Sub
```

```
Private Sub tel_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
If IsNumeric(tel.Text) Then
numesc.SetFocus
Else
MsgBox "Solo caracteres numericos", 0 + 48, "Error"
End If
End If
End Sub
```

```
Private Sub Consultar()
'HACEMOS LA CONEXION CON LA BASE DE DATOS'
```

```
Set Registro = New ADODB.Recordset

'CONSULTAMOS Registro
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapdocentes WHERE DOCID" _
& "=" & id & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "DOCENTES", " -
Identificacion=" & id)
nom = Registro!docnom
ape = Registro!docapelli
numesc = Registro!DOCNUMESC
sex = Registro!DOCSEXO
fec = Registro!DOCFECNAC
dir = Registro!DOCDIR
tel = Registro!DOCTELE
tipdoc = Registro!DOCTIPDOC
est = Registro!DOCESTA
Dim strSQL As String
Dim k As Double
strSQL = "SELECT asigcod FROM siapdocasig where docid = " + id.Text + ""
MsgBox strSQL
Set Registro = Conexion.Execute(strSQL)
```

```

k = 1
Do Until Registro.EOF

' Deben contar cuantos registros tiene
MSdoc.Rows = MSdoc.Rows + 1
MSdoc.Row = k
MSdoc.COL = 0
MSdoc.Text = Registro!asigcod
Registro.MoveNext
k = k + 1
Loop
filas = 1
While MSdoc.Text <> ""
MSdoc.Row = filas
MSdoc.COL = 0
tem = MSdoc.Text
Set Registro = New ADODB.Recordset
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ASIGNATURAS",
" - Identificacion=" & id)
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapasignaturas WHERE asigcod" _
& "=" & tem & """)
If Not Registro.EOF Then
MSdoc.COL = 1
MSdoc.Text = Registro!asignom
End If
filas = filas + 1
Wend
MsgBox "Registro consultado", vbInformation, _
"MENSAJE"
Else
MsgBox "No existe registro" _
& " con este NIT", vbExclamation, _
"ATENCIÓN"
Limpiar
End If
Registro.Close

End Sub
Private Sub ELIMINAR()
If validar Then
'HACEMOS LA CONEXION CON LA BASE DE DATOS'

Set Registro = New ADODB.Recordset
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "DOCENTES", " -
Identificacion=" & id)
'CONSULTAMOS Registro

```

```

Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapdocentes WHERE DOCID" _
& "=" & id & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "DOCENTES", " -
Identificacion=" & id)
Conexion.Execute "delete from siapdocentes where DOCNOM=" & nom & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCAPELLI=" & ape & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCNUMESC=" & numesc & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCSEXO=" & sex & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCFECNAC=" & fec & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCDIR=" & dir & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCTELE=" & tel & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCTIPDOC=" & tipdoc & "and DOCID=" & id & ""
Conexion.Execute "delete from siapdocentes where DOCESTA=" & est & "and DOCID=" & id & ""
MsgBox "Registro eliminado", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "No existe este registro con este" _
& "NIT", vbExclamation, _
"ATENCION"
Limpiar
End If
Registro.Close

End If
End Sub

```

```

Private Sub Limpiar()
Dim y As Integer
y = 1
While y < MSdoc.Rows
MSdoc.Rows = MSdoc.Rows - 1
Wend
If MSdoc.Rows = 0 Then
MSdoc.Rows = MSdoc.Rows + 1
'MSdoc.Row = i

```

```
MSdoc.COL = 0
MSdoc.Text = ""
MSdoc.COL = 1
MSdoc.Text = ""
MSdoc.COL = 2
MSdoc.Text = ""
End If
MSdoc.Rows = MSdoc.Rows + 1
id.Enabled = True
id = ""
nom = ""
ape = ""
numesc = ""
sex = ""
dir = ""
tel = ""
tipdoc = ""
est = ""
id.SetFocus
End Sub
Private Function validar() As Boolean
validar = True
If Trim(id.Text) = "" Then
validar = False
End If
If Trim(nom.Text) = "" Then
validar = False
End If
If Trim(ape.Text) = "" Then
validar = False
End If
If Trim(numesc.Text) = "" Then
validar = False
End If
If Trim(sex.Text) = "" Then
validar = False
End If
If Trim(fec) = "" Then
validar = False
End If
If Trim(tel.Text) = "" Then
validar = False
End If
If Trim(tipdoc.Text) = "" Then
validar = False
End If
If Trim(est.Text) = "" Then
```

```

validar = False
End If
If validar = False Then
MsgBox "falta algun campo", 16
End If
End Function

Private Sub id_keypress(KeyAscii As Integer)
If (KeyAscii = 13) And (id.Text = "") Then
MsgBox "Deben existir datos", vbCritical, "Atencion"
id.SetFocus
Else
If (KeyAscii = 13) Then
Consultar
End If
End If
End Sub

Private Sub salir()
Unload Me
End Sub

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command1.BackColor = &HFFFFFF
End Sub

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command2.BackColor = &HFFFFFF
End Sub

Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command3.BackColor = &HFFFFFF
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2
actualizar
Case 3
Consultar
Case 4
ELIMINAR
Case 5
Limpiar
Case 6
salir

```

End Select
End Sub

Estudiantes

```
Private Sub Adicionar()  
    Set Registro = New ADODB.Recordset  
    If SSTab1.Tab = 0 Then  
        l = Len(cod)  
        If (l <= 10) Then  
            Set Registro = Conexion.Execute("SELECT" _  
            & "*" FROM SIAPESTUDIANTES WHERE ESTID" _  
            & "=" & cod & """)  
            If Registro.EOF Then  
                d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ESTUDIANTES", "  
                - Identificacion=" & cod)  
                d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSInsert, "ESTUDIANTES", "  
                - Identificacion=" & cod)  
                Conexion.Execute "INSERT INTO SIAPESTUDIANTES" _  
                &  
                "(ESTID,ESTTIPDOC,ESTNOMBRE,ESTAPELLI,ESTSEXO,ESTFECNAC,ESTTELE,  
                ESTOBS,ESTTIPCLI,ESTESTA,ESTDIR) VALUES (" & cod _  
                & ", " & ti & ", " & n & ", " & ape & ", " & s & ", " & f & ", " & t & ", " & o & ", " &  
                tip & ", " & es & ", " & di & """)  
                MsgBox "Datos registrados correctamente", vbInformation, _  
                "MENSAJE"  
            Else  
                MsgBox "Ya existe un registro con ese" _  
                & "id", vbExclamation, _  
                "ATENCION"  
            End If  
            Registro.Close  
        Else  
            MsgBox "¡La longitud del codigo del estudiante no es permitida!Longitud permitida 10  
            digitos ", vbExclamation, _  
            "ATENCION"  
        End If  
    Else  
        l = Len(acuid)  
        If (l <= 10) Then  
            Set Registro = Conexion.Execute("SELECT" _  
            & "*" FROM SIAPACUDIENES WHERE ACUID" _  
            & "=" & acuid & """)  
            If Registro.EOF Then  
                d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ACUDIENES", "  
                - Identificacion=" & acuid)
```



```

d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSInsert, "ACUDIENES", "
- Identificacion=" & acuid)
Conexion.Execute "INSERT INTO SIAPACUDIENES" _
&
"(ACUID,ACUNOMBRE,ACUPELLI,ACUSEXO,ACUTELE,ACUFECNAC,ACUDIR,
ACUPAREN,ACUESTA) VALUES (" & acuid _
& "," & acunom & "," & acuape & "," & acusex & "," & acutel & "," & acufec & ","
& acudir & "," & acupar & "," & acuest & ")"
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
Else
MsgBox "Ya existe un registro con ese" _
& " ID", vbExclamation, _
"ATENCION"
End If
Registro.Close
Else
MsgBox "¡La longitud del codigo del acudiente no es permitida!Longitud permitida 10
digitos ", vbExclamation, _
"ATENCION"
End If
End If
End Sub
Private Sub Consultar()
Set Registro = New ADODB.Recordset
If SSTab1.Tab = 0 Then
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapestudiantes WHERE ESTID" _
& "=" & cod & """)
If Not Registro.EOF Then
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "ESTUDIANTES",
" - Identificacion=" & cod)
idest = Registro!estid
ti = Registro!ESTTIPDOC
n = Registro!ESTNOMBRE
nomest = n
ape = Registro!ESTAPELLI
apeest = ape
s = Registro!ESTSEXO
f = Registro!ESTFECNAC
t = Registro!ESTTELE
o = Registro!ESTOBS
tip = Registro!ESTTIPCLI
es = Registro!ESTESTA
di = Registro!ESTDIR
Dim strSQL As String
Dim k As Double

```

```

strSQL = "SELECT ACUID FROM siapacuest where ESTID = "" + cod.Text + ""
MsgBox strSQL
Set Registro = Conexion.Execute(strSQL)
k = 1
Do Until Registro.EOF

' Deben contar cuantos registros tiene
MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1
MSFlexGrid1.Row = k
MSFlexGrid1.COL = 0
MSFlexGrid1.Text = Registro!acuid
Registro.MoveNext
k = k + 1
Loop
filas = 1
While MSFlexGrid1.Text <> ""
MSFlexGrid1.Row = filas
MSFlexGrid1.COL = 0
tem = MSFlexGrid1.Text
Set Registro = New ADODB.Recordset
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapacudientes WHERE ACUID" _
& "=" & tem & """)
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ACUDIENES", "
- Identificacion=" & acuid)
MSFlexGrid1.COL = 1
MSFlexGrid1.Text = Registro!acunombre
MSFlexGrid1.COL = 2
MSFlexGrid1.Text = Registro!acuapelli
End If
filas = filas + 1
Wend

MsgBox "Registro consultado", vbInformation, _
"MENSAJE"
Else
MsgBox "no existe registro" _
& " con este id", vbExclamation, _
"ATENCION"
End If
Else
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapacudientes WHERE ACUID" _
& "=" & acuid & """)
If Not Registro.EOF Then

```

```

d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "ACUDIEN-
tes", "
- Identificacion=" & acuid)
acunom = Registro!acunombre
acuape = Registro!acuapelli
acusex = Registro!ACUSEXO
acutel = Registro!ACUTELE
acufec = Registro!ACUFECNAC
acudir = Registro!acudir
acupar = Registro!ACUPAREN
acuest = Registro!ACUESTA
Else
MsgBox "no existe registro" _
& " con este ID", vbExclamation, _
"ATENCION"
End If
End If
Registro.Close
End Sub
Private Sub actualizar()
Set Registro = New ADODB.Recordset
If SSTab1.Tab = 0 Then
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapestudiantes WHERE ESTID" _
& "=" & cod & """)
If Not Registro.EOF Then
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "ESTUDIANTES",
" - Identificacion=" & cod)
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSUpdate, "ESTUDIANTES",
" - Identificacion=" & cod)
Conexion.Execute "update siapestudiantes set ESTTIPDOC=" _
& ti & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTNOMBRE=" _
& n & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTAPELLI=" _
& ape & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTSEXO=" _
& s & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTFECNAC=" _
& f & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTTELE=" _
& t & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTOBS=" _
& o & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTTIPCLI=" _
& tip & ""where ESTID=" & cod & ""
Conexion.Execute "update siapestudiantes set ESTESTA=" _
& es & ""where ESTID=" & cod & ""

```

```

Conexion.Execute "update siapestudiantes set ESTDIR="" _
& di & "where ESTID="" & cod & ""
MsgBox "Registro actualizado", vbInformation, _
"MENSAJE"
Else
MsgBox "no existe registro" _
& "con este id", vbExclamation, _
"ATENCION"
End If
Else
Set Registro = Conexion.Execute("SELECT" _
& "*" FROM siapacudientes WHERE ACUID" _
& "=" & acuid & """)
If Not Registro.EOF Then
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSSelect, "ACUDIENES", "
- Identificacion=" & acuid)
d = ProcAuditoria1(txtuser, txtname, vCodigoFormulario, SCSUpdate, "ACUDIENES",
" - Identificacion=" & acuid)
Conexion.Execute "update siapacudientes set ACUNOMBRE="" _
& acunom & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUPELLI="" _
& acuape & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUSEXO="" _
& acusex & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUTELE="" _
& acusex & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUFECNAC="" _
& acufec & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUDIR="" _
& acudir & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUPAREN="" _
& acupar & "where ACUID="" & acuid & ""
Conexion.Execute "update siapacudientes set ACUESTA="" _
& acuest & "where ACUID="" & acuid & ""
MsgBox "Registro actualizado correctamente", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "no existe registro con " _
& "este ID", vbExclamation, _
"ATENCION"
End If
End If
Registro.Close
End Sub
Private Sub ELIMINAR()
Set Registro = New ADODB.Recordset

```

```

If SSTab1.Tab = 0 Then
  Set Registro = Conexion.Execute("SELECT" _
  & "* FROM siapestudiantes WHERE ESTID=" _
  & "=" & cod & """)
  If Not Registro.EOF Then
    d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ESTUDIANTES",
    " - Identificacion=" & cod)
    d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "ESTUDIANTES",
    " - Identificacion=" & cod)
    Conexion.Execute "delete from siapestudiantes where ESTID=" _
    & i & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTTIPDOC=" _
    & ti & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTNOMBRE=" _
    & n & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTAPELLI=" _
    & ape & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTSEXO=" _
    & s & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTFECNAC=" _
    & f & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTTELE=" _
    & t & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTOBS=" _
    & o & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTTIPCLI=" _
    & tip & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTESTA=" _
    & es & ""and ESTID=" & cod & ""
    Conexion.Execute "delete from siapestudiantes where ESTDIR=" _
    & di & ""and ESTID=" & cod & ""

  MsgBox "se ha eliminado este registro", vbInformation, _
  "MENSAJE"
  Else
  MsgBox "no existe este registro" _
  & "NIT", vbExclamation, _
  "ATENCION"
  End If
  Else
  Set Registro = Conexion.Execute("SELECT" _
  & "* FROM siapacudientes WHERE ACUID" _
  & "=" & acuid & """)
  If Not Registro.EOF Then
    d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ACUDIENES", "
  - Identificacion=" & acuid)

```

```

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSDelete, "ACUDIENES", "
- Identificacion=" & acuid)
Conexion.Execute "delete from siapacudientes where ACUID=" & acuid & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUNOMBRE=" & acunom & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUAPELLI=" & acuape & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUSEXO=" & acusex & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUTELE=" & acutel & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUFECNAC=" & acufec & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUDIR=" & acudir & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUPAREN=" & acupar & ""and ACUID=" & acuid & ""
Conexion.Execute "delete from siapacudientes where ACUESTA=" & acuest & ""and ACUID=" & acuid & ""
MsgBox "Registro eliminado", vbInformation, _
"MENSAJE"
Limpiar
Else
MsgBox "no existe este registro con este" & "ID", vbExclamation, _
"ATENCION"
Limpiar
End If
End If
Registro.Close
End Sub
Private Sub salir()
Unload Me
End Sub
Private Sub Limpiar()
cod.Enabled = True
acuid.Enabled = True

If SSTab1.Tab = 0 Then
cod = ""
ti = ""
n = ""
ape = ""
s = ""
t = ""
o = ""

```



```

tip = ""
es = ""
di = ""
cod.SetFocus
Else
If Option1.Value = True Then
acuid = ""
acunom = ""
acuape = ""
acusex = ""
acutel = ""
acudir = ""
acupar = ""
acuest = ""
acuid.SetFocus
Else
Dim y As Integer
y = 1
While y < MSFlexGrid1.Rows
MSFlexGrid1.Rows = MSFlexGrid1.Rows - 1
Wend
If MSFlexGrid1.Rows = 0 Then
MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1
MSFlexGrid1.Row = i
MSFlexGrid1.COL = 0
MSFlexGrid1.Text = ""
MSFlexGrid1.COL = 1
MSFlexGrid1.Text = ""
MSFlexGrid1.COL = 2
MSFlexGrid1.Text = ""
End If
MSFlexGrid1.Rows = MSFlexGrid1.Rows + 1
idest = ""
nomest = ""
apeest = ""
End If
End If
End Sub
Private Function validar() As Boolean
If SSTab1.Tab = 0 Then
validar = True
If Trim(cod.Text) = "" Then
validar = False
End If
If Trim(ti.Text) = "" Then
validar = False
End If

```

```
If Trim(n.Text) = "" Then
validar = False
End If
If Trim(ape.Text) = "" Then
validar = False
End If
If Trim(s) = "" Then
validar = False
End If
If Trim(f) = "" Then
validar = False
End If
If Trim(t.Text) = "" Then
validar = False
End If
If Trim(o.Text) = "" Then
validar = False
End If
If Trim(tip) = "" Then
validar = False
End If
If Trim(es.Text) = "" Then
validar = False
End If
If Trim(di.Text) = "" Then
validar = False
End If
If validar = False Then
MsgBox "falta algun campo"
End If
Else
validar1 = True
If Trim(acuid.Text) = "" Then
validar1 = False
End If
If Trim(acunom.Text) = "" Then
validar1 = False
End If
If Trim(acuape.Text) = "" Then
validar1 = False
End If
If Trim(acusex.Text) = "" Then
validar1 = False
End If
If Trim(acutel.Text) = "" Then
validar1 = False
End If
```

```

If Trim(acufec) = "" Then
validar1 = False
End If
If Trim(acudir.Text) = "" Then
validar1 = False
End If
If Trim(acupar) = "" Then
validar1 = False
End If
If Trim(acuest) = "" Then
validar1 = False
End If
If validar1 = False Then
MsgBox "falta algun campo", 16
End If
End If
End Function

```

```

Private Sub Command1_Click()
MSACUDI.Show vbModal
End Sub

```

```

Private Sub Command2_Click()
MSACUDIENTES.Show vbModal
End Sub

```

```

Private Sub Command3_Click()
Dim i, sw As Integer
sw = 0
MSFlexGrid1.Row = 1
MSFlexGrid1.COL = 0
If MSFlexGrid1.Text = "" Then
MsgBox "Debe seleccionar " _
& "los acudientes", vbExclamation, _
"ATENCION"
Else
If idest.Text <> "" Then
For i = 1 To MSFlexGrid1.Rows - 2
MSFlexGrid1.Row = i
MSFlexGrid1.COL = 0
idl = MSFlexGrid1.Text
Set Registro = Conexion.Execute("SELECT " _
& "*" FROM siapacuest WHERE ESTID " _
& "=" & idest & " And acuid " _
& "=" & idl & """)

```

```

If Registro.EOF Then
Conexion.Execute "INSERT INTO siapacuest" _
& "(estid,acuid) VALUES (" & idest _
& "," & id1 & ")"
sw = 0
End If
Registro.Close
Next i
If sw = 0 Then
MsgBox "Datos registrados correctamente", vbInformation, _
"MENSAJE"
End If
Else
MsgBox "Debe existir el Id " _
& "del Estudiante", vbExclamation, _
"ATENCION"
End If
End If
End Sub
Private Sub COD_KeyPress(Key Ascii As Integer)
If (KeyAscii = 13) And (cod.Text = "") Then
MsgBox "Deben existir datos", vbCritical, "Atencion"
cod.SetFocus
Else
If (KeyAscii = 13) Then
Consultar
End If
End If
End Sub

Private Sub Command4_Click()
Set Registro = New ADODB.Recordset

d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSelect, "ESTUDIANTES",
" - Identificacion=" & idest)
Set Registro = Conexion.Execute("SELECT" _
& "* FROM siapacuest WHERE acuid" _
& "=" & dateli2 & "")
If Not Registro.EOF Then
d = ProcAuditorial(txtuser, txtname, vCodigoFormulario, SCSSDelete, "ESTUDIANTES",
" - Identificacion=" & idest)
Conexion.Execute "delete from siapacuest where acuid=" & _
& dateli2 & "and acuid=" & dateli2 & ""
MsgBox "Acudiente eliminado", vbInformation, _
"MENSAJE"
End If

```

```
Registro.Close  
End Sub
```

```
Private Sub Command5_Click()  
MSESTUDIANTES1.Show vbModal
```

```
End Sub
```

```
Private Sub MSFlexGrid1_Click()  
MSFlexGrid1.COL = 0  
dateli2 = MSFlexGrid1.Text  
End Sub
```

```
Private Sub SSTab1_GotFocus()  
SSTab1.Tab = 0  
End Sub
```

```
Private Sub Form_Load()  
Frameacu.Visible = False  
Frameest.Visible = False  
Frameasigacu.Visible = False
```

```
End Sub
```

```
Private Sub Label14_Click()  
Option1.Value = True  
End Sub
```

```
Private Sub Label15_Click()  
Option2.Value = True  
End Sub
```

```
Private Sub Option1_Click()  
Frameacu.Visible = True  
Frameest.Visible = False  
Frameasigacu.Visible = False
```

```
End Sub
```

```
Private Sub Option2_Click()  
Frameest.Visible = True  
Frameacu.Visible = False  
Frameest.Enabled = False  
Frameasigacu.Visible = True  
End Sub
```

```
Private Sub Frameasigacu_MouseMove(Button As Integer, Shift As Integer, x As Single, y  
As Single)  
Command4.BackColor = &H8000000A
```

```

Command3.BackColor = &H8000000A
Command2.BackColor = &H8000000A
End Sub
Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command4.BackColor = &HFFFFFF
End Sub
Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command2.BackColor = &HFFFFFF
End Sub
Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, x As Single, y
As Single)
Command3.BackColor = &HFFFFFF
End Sub
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
Select Case Button.Index
Case 1
Adicionar
Case 2
actualizar
Case 3
Consultar
Case 4
ELIMINAR
Case 5
Limpiar
Case 6
salir
End Select
End Sub

```

Generador de reporte dinamico

Option Explicit

Dim CmpAgr As Boolean

Private Sub btnExport_Click()

On Error GoTo ErrHandler

DLG.Filter = "Excel File(*.xls)|*.xls|Text File (*.txt)|*.txt"

DLG.FilterIndex = 1

DLG.ShowSave

ExportarGrid MSFDatos, DLG.FileName, DLG.FilterIndex

ErrHandler:

End Sub

```

*****
*****
*****
*****
Private Sub CmbAtab_Click()
    Dim tStrAlias, i, vPosA
    tStrAlias = CmbAlias & " AS " & UCase(TxtAliaTab)
    vPosA = FindTab(CmbAlias)
    For i = 1 To ListAlias.ListCount
        If (ListAlias.List(i - 1) = tStrAlias Or ListAlias.List(i) = tStrAlias) Then
            MsgBox "No puede agregar el nuevo Alias, esta ya se encuentra en la lista",
vbInformation, "Atencion!"
            Exit Sub
        End If
    Next
    TxtAliaTab = ""
    ListAlias.AddItem tStrAlias
    If vPosA = CmbAlias.ListCount Then
        CmbConAlia.Enabled = True
    End If
End Sub

```

```

*****
*****
*****
*****
Private Sub CmbCampC_Click()
    Dim vPosT, vPosC, Func, Oper, i
    vPosT = FindTab(CmbTabC)
    vPosC = FindCmp(CmbTabC, CmbCampC)
    With Tablas(vPosT).tCampo(vPosC)
        Select Case UCase(.cTipo)
            Case "INT"
                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Valor Entero - Longitud
Maxima: " + .cLongitud
                Oper = "Between,Like,=,>,<,>=,<=,<>"
            Case "VARCHAR", "CHAR"
                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Valor Alfanumerico -
Longitud Maxima: " + .cLongitud
                Oper = "Like,=,<>"
            Case "ENUM"
                InfoCondicion = "Informacion: " + vbCrLf + "Campo: Lista - Valores : " +
.cValores
                Oper = "Like,=,<>"
            Case "DATE"

```

```

        InfoCondicion = "Informacion: " + vbCrLf + "Campo: Fecha - Formato:
YYYYY/MM/DD"
        Oper = "Between,=,>,<,>=,<=,<>"
        Case "TIME"
            InfoCondicion = "Informacion: " + vbCrLf + "Campo: Tiempo - Formato:
HH:MM:SS"
            Oper = "Between,=,>,<,>=,<=,<>"
        Case Else
            InfoCondicion = ""
        End Select
    End With
    CmbOperC.Clear
    Oper = Split(Oper, ",")
    For i = 0 To UBound(Oper)
        CmbOperC.AddItem Oper(i)
    Next
    If CmbOperC.ListCount > 0 Then
        CmbOperC.ListIndex = 0
    End If

```

End Sub

```

Private Sub CmbCampF_Click()
    Dim vPosT, vPosC, Func, i
    vPosT = FindTab(CmbTabF)
    vPosC = FindCmp(CmbTabF, CmbCampF)
    With Tablas(vPosT).tCampo(vPosC)
        Select Case UCase(.cTipo)
            Case "INT"
                InfoCampo = "Informacion: " + vbCrLf + "Campo: Valor Entero - Longitud
Maxima: " + .cLongitud
                Func = "NINGUNO,COUNT,MAX,MIN,AVG,SUM"
            Case "VARCHAR", "CHAR"
                InfoCampo = "Informacion: " + vbCrLf + "Campo: Valor Alfanumerico - Longitud
Maxima: " + .cLongitud
                Func = "NINGUNO,COUNT"
            Case "ENUM"
                InfoCampo = "Informacion: " + vbCrLf + "Campo: Lista - Valores : " + .cValores
                Func = "NINGUNO,COUNT"
            Case "DATE"
                InfoCampo = "Informacion: " + vbCrLf + "Campo: Fecha - Formato:
YYYYY/MM/DD"
                Func = "NINGUNO,COUNT,MAX,MIN"
            Case "TIME"
                InfoCampo = "Informacion: " + vbCrLf + "Campo: Tiempo - Formato:
HH:MM:SS"

```

```

        Func = "NINGUNO,COUNT,MAX,MIN"
    Case Else
        InfoCampo = ""
    End Select
End With

CmbFuncF.Clear
Func = Split(Func, ",")
For i = 0 To UBound(Func)
    CmbFuncF.AddItem Func(i)
Next
CmbFuncF.ListIndex = 0
End Sub

Private Sub CmbCmp1_Click()
    Dim tPos1, cPos1, tPos2, cPos2
    Dim tab1, tab2 As String
    Dim sw As Boolean

    tPos1 = FindATab(CmbTab1)
    tPos2 = FindATab(CmbTab2)
    cPos1 = FindACmp(CmbTab1, CmbCmp1)
    cPos2 = FindACmp(CmbTab2, CmbCmp2)

    Call CON1
    ds.Open conex
    Set rs = ds.Execute("select * from scsllaves where (llav_nom1 = "" +
    Tablas(tPos1).tNombre + "" and llav_nom2="" + Tablas(tPos2).tNombre + "" )")
    If Not rs.EOF Then
        tab1 = rs!llav_cod_tab1
        tab2 = rs!llav_cod_tab2
        sw = True
    Else
        rs.Close
        Set rs = ds.Execute("select * from scsllaves where (llav_nom2 = "" +
    Tablas(tPos1).tNombre + "" and llav_nom1 = "" + Tablas(tPos2).tNombre + "" )")
        If Not rs.EOF Then
            tab1 = rs!llav_cod_tab1
            tab2 = rs!llav_cod_tab2
            sw = True
        Else
            sw = False
        End If
    End If
    rs.Close
    ds.Close

```

```

InfoJoin = "Informacion: " + vbCrLf

If (Tablas(tPos1).tCampo(cPos1).cTipo <> Tablas(tPos2).tCampo(cPos2).cTipo) Then
    InfoJoin = InfoJoin + "No se puede realizar la union por estos campos, por no ser del
mismo tipo" + _
        vbCrLf + CmbTab1 + "." + CmbCmp1 + ": " +
Tablas(tPos1).tCampo(cPos1).cTipo + _
        vbCrLf + CmbTab2 + "." + CmbCmp2 + ": " +
Tablas(tPos2).tCampo(cPos2).cTipo
    CmdAddJoin.Enabled = False
    ElseIf ((CmbCmp1 = UCase(tab1) And CmbCmp2 <> UCase(tab2))) Then
        InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no tener
Relacion la Relacion existente es:" + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)
        CmdAddJoin.Enabled = False
    ElseIf ((CmbCmp2 = UCase(tab1) And CmbCmp1 <> UCase(tab2))) Then
        InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no tener
Relacion la Relacion existente es:" + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)
        CmdAddJoin.Enabled = False
    ElseIf ((CmbCmp1 <> UCase(tab1) And CmbCmp2 <> UCase(tab2)) And (CmbCmp1
<> UCase(tab2) And CmbCmp2 <> UCase(tab1))) Then
        InfoJoin = InfoJoin + " No se puede realizar la union por estos campos, por no tener
Relacion la Relacion existente es:" + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab1 + ": El Campo " + UCase(tab1) + _
            vbCrLf + _
            vbCrLf + "De la tabla " + CmbTab2 + ": El Campo " + UCase(tab2)
        CmdAddJoin.Enabled = False
    If Not (sw) Then
        MsgBox "Seleccione Tablas con Relacion", vbInformation, " Informacion"
        CmbTab1.ListIndex = 0
        CmbTab2.ListIndex = 1
    End If
    ElseIf Not (sw) Then
        MsgBox "Seleccione Tablas con Relacion", vbInformation, " Informacion"
        CmbTab1.ListIndex = 0
        CmbTab2.ListIndex = 1
    Else
        InfoJoin = InfoJoin + vbCrLf + "Puede continuar con la union."
        CmdAddJoin.Enabled = True

```

```
End If  
End Sub
```

```
Private Sub CmbCmp2_Click()  
    CmbCmp1_Click  
End Sub
```

```
Private Sub CmbConAlia_Click()  
    Dim i As Long  
    Dim Txt  
    InitTab  
    If (ListAlias.ListCount > 0) Then  
        For i = 1 To ListAlias.ListCount  
            Txt = Split(ListAlias.List(i - 1), " ")  
            TxtAliaTab = Txt(2)  
            addAlias (TxtAliaTab)  
        Next  
    End If  
    If (Pos > 1) Then  
        SSTab.TabEnabled(1) = True  
        SSTab.Tab = 1  
    Else  
        SSTab.TabEnabled(2) = True  
        SSTab.Tab = 2  
    End If  
End Sub
```

```
Private Sub CmbLimAlia_Click()  
    ListAlias.Clear  
    TxtAliaTab = ""  
End Sub
```

```
Private Sub CmbLimDato_Click()  
    MSFDatos.Clear  
    MSFDatos.Rows = 2  
    MSFDatos.Cols = 1  
End Sub
```

```
Private Sub CmbOperC_Click()  
    Select Case UCase(CmbOperC)  
        Case "BETWEEN": ValC2.Enabled = True  
        Case Else: ValC2.Enabled = False  
    End Select  
End Sub
```

```
Private Sub CmbRegAlia_Click()  
    SSTab.Tab = 0
```

End Sub

```
Private Sub CmbRegRes_Click()  
    SSTab.Tab = 3  
End Sub
```

```
Private Sub CmbTab1_Click()  
    Dim vPos, i  
    If (CmbTab1 <> CmbTab2) Then  
        vPos = FindATab(CmbTab1)  
        CmbCmp1.Clear  
        For i = 1 To Tablas(vPos).tNumCamp  
            CmbCmp1.AddItem Tablas(vPos).tCampo(i).cNombre  
        Next  
        CmbCmp1.ListIndex = 0  
    Else  
        MsgBox "La union debe ser entre dos tablas diferentes, seleccione otra tabla para poder  
continuar", vbInformation, "Atencion"  
        If (CmbTab1.ListIndex > 0) Then  
            CmbTab1.ListIndex = 0  
        Else  
            CmbTab1.ListIndex = 1  
        End If  
        CmbTab1_Click  
    End If  
End Sub
```

```
Private Sub CmbTab2_Click()  
    Dim vPos, i  
    If (CmbTab1 <> CmbTab2) Then  
        vPos = FindATab(CmbTab2)  
        CmbCmp2.Clear  
        For i = 1 To Tablas(vPos).tNumCamp  
            CmbCmp2.AddItem Tablas(vPos).tCampo(i).cNombre  
        Next  
        CmbCmp2.ListIndex = 0  
    Else  
        MsgBox "La union debe ser entre dos tablas diferentes, seleccione otra tabla para poder  
continuar", vbInformation, "Atencion"  
        If (CmbTab2.ListIndex > 0) Then  
            CmbTab2.ListIndex = 0  
        Else  
            CmbTab2.ListIndex = 1  
        End If  
        CmbTab2_Click  
    End If  
End Sub
```

```

Private Sub CmbTabC_Click()
    Dim vPos, i
    vPos = FindATab(CmbTabC)
    CmbCampC.Clear
    For i = 1 To Tablas(vPos).tNumCamp
        CmbCampC.AddItem Tablas(vPos).tCampo(i).cNombre
    Next
    CmbCampC.ListIndex = 0
End Sub

Private Sub CmbTabF_Click()
    Dim vPos, i
    vPos = FindATab(CmbTabF)
    CmbCampF.Clear
    For i = 1 To Tablas(vPos).tNumCamp
        CmbCampF.AddItem Tablas(vPos).tCampo(i).cNombre
    Next
    CmbCampF.ListIndex = 0
End Sub

Private Sub CmdAddCamp_Click()
    Dim KeyWord, tStrCmp, ListaAgr, xCmp, i, sw As Boolean
    KeyWord = " ' SELECT AS FROM WHERE GROUP BY ORDER HAVING SUM
MAX MIN AVG COUNT INSERT INTO VALUES UPDATE SET UNION
INTERSECTION EXCEPT EXECUTE DELETE DROP ALTER TABLE CREATE
MODIFY COLUMN ADD CONSTRAINT PRIMARY KEY RENAME VARCHAR
CHAR NUMBER INT ENUM TIME DATE SEQUENCE JOIN LEFT RIGH "
    tStrCmp = CmbTabF + "." + CmbCampF

    If UCase(CmbFuncF) <> "NINGUNO" Then
        tStrCmp = CmbFuncF + "(" + tStrCmp + ")"
    End If
    If Trim(txtAlias) <> vbNullString Then
        If (InStr(1, KeyWord, UCase(Trim(txtAlias)))) = 0 Then
            tStrCmp = tStrCmp + " AS " + txtAlias
        Else
            MsgBox "Los alias no pueden contener las siguientes palabras" + vbCrLf +
KeyWord
            Exit Sub
        End If
    End If

    For i = 1 To ListCondiciones.ListCount
        If (ListCampos.List(i - 1) = tStrCmp Or ListCampos.List(i) = tStrCmp) Then
            MsgBox "No puede agregar la nueva union, esta ya se encuentra en la lista",
vbInformation, "Atencion!"

```

```

Exit Sub
End If
Next

If CmbFuncF <> "NINGUNO" Then
If CmpAgr Or ListCampos.ListCount = 0 Then
ListCampos.AddItem tStrCmp
Else
ListaAgr = ""
For i = 1 To ListCampos.ListCount
If (InStr(1, ListCampos.List(i - 1), "(")) Then
xCmp = Split(ListCampos.List(i - 1), "(")
xCmp = Split(xCmp(0), "(")
xCmp = xCmp(1)
Else
xCmp = Split(ListCampos.List(i - 1), " AS ")
xCmp = xCmp(0)
End If

If ListaAgr = vbNullString Then
ListaAgr = xCmp
Else
ListaAgr = ListaAgr + "," + xCmp
End If
Next

If (MsgBox("Atencion al insertar esta funcion de agregacion " + vbCrLf + "usted
debe agrupar la consulta por los siguientes campos: " + vbCrLf + ListaAgr + vbCrLf +
"Desea hacerlo?", vbYesNo + vbInformation, "Atencion!") = vbYes) Then
ListCampos.AddItem tStrCmp
CmpAgr = True
ListaAgr = Split(ListaAgr, ",")
For i = 0 To UBound(ListaAgr)
ListGroup.AddItem Trim(ListaAgr(i))
Next
Else
Exit Sub
End If
End If
Else
If (Not CmpAgr) Then
sw = False
xCmp = CmbTabF + "." + CmbCampF
For i = 1 To ListCampos.ListCount
If (InStr(1, ListCampos.List(i - 1), "COUNT") Or InStr(1, ListGroup.List(i -
1), "MIN") Or InStr(1, ListGroup.List(i - 1), "MAX") Or InStr(1, ListGroup.List(i - 1),
"SUM") Or InStr(1, ListGroup.List(i - 1), "AVG")) Then
sw = True

```

```

        Exit For
    End If
Next
If (Not sw) Then
    ListCampos.AddItem tStrCmp
    ElseIf (MsgBox("Atencion para adicionar este campo, usted debe agrupar la
consulta por el mismo desea hacerlo?", vbYesNo + vbInformation, "Atencion!") = vbYes)
Then
        ListCampos.AddItem tStrCmp
        ListGroup.AddItem xCmp
        CmpAgr = True
    End If
Else
    sw = False
    xCmp = CmbTabF + "." + CmbCampF
    For i = 1 To ListGroup.ListCount
        If xCmp = Trim(ListGroup.List(i - 1)) Then
            sw = True
            Exit For
        End If
    Next
    If (sw) Then
        ListCampos.AddItem tStrCmp
    Else
        If (MsgBox("Atencion para adicionar este campo, usted debe agrupar la
consulta por el mismo desea hacerlo?", vbYesNo + vbInformation, "Atencion!") = vbYes)
Then
            ListCampos.AddItem tStrCmp
            ListGroup.AddItem xCmp
        Else
            Exit Sub
        End If
    End If
End If
End Sub

```

```

Private Sub CmdAddCond_Click()
    Dim tStrCond, Msg, tPos, cPos, i
    tPos = FindTab(CmbTabC)
    cPos = FindCmp(CmbTabC, CmbCampC)
    If (UCase(CmbOperC) = "LIKE") Then
        tStrCond = CmbTabC + "." + CmbCampC + " " + CmbOperC + " " + ValC1 + "%"
+ ""
    Else
        tStrCond = CmbTabC + "." + CmbCampC + " " + CmbOperC + " " + ValC1 + ""
    End If

```

```

For i = 1 To ListCondiciones.ListCount
    If (ListCondiciones.List(i - 1) = tStrCond Or ListCondiciones.List(i) = tStrCond) Then
        MsgBox "No puede agregar la nueva union, esta ya se encuentra en la lista",
vbInformation, "Atencion!"
        Exit Sub
    End If
Next

```

```

With Tablas(tPos).tCampo(cPos)

```

```

    If Not ValidarDato(.cTipo, ValC1, .cValores) Then
        ValC1.SetFocus
        Exit Sub
    End If
    If (UCase(CmbOperC) = "BETWEEN") Then
        If Not ValidarDato(.cTipo, ValC2, .cValores) Then
            ValC2.SetFocus
            Exit Sub
        End If
        tStrCond = tStrCond + " AND " + ValC2 + ""
    End If

```

```

    If (CmbOpeLC.Enabled) Then
        tStrCond = CmbOpeLC + " " + tStrCond
    End If
    ListCondiciones.AddItem tStrCond
    If (ListCondiciones.ListCount > 1) Then
        CmbOpeLC.Enabled = True
        CmbOpeLC.ListIndex = 0
    End If

```

```

End With
End Sub

```

```

Private Sub CmdAddJoin_Click()

```

```

    Dim tStrJoin, tStrJoin2, i
    tStrJoin = CmbTab1 & "." & CmbCmp1 & "=" & CmbTab2 & "." & CmbCmp2
    tStrJoin2 = tStrJoin
    For i = 1 To ListJoin.ListCount
        If (ListJoin.List(i - 1) = tStrJoin Or ListJoin.List(i) = tStrJoin2) Then
            MsgBox "No puede agregar la nueva union, esta ya se encuentra en la lista",
vbInformation, "Atencion!"
            Exit Sub
        End If
    Next
    ListJoin.AddItem tStrJoin
End Sub

```

```
Private Sub CmdBackCampo_Click()  
    SSTab.Tab = 2  
End Sub
```

```
Private Sub CmdBackCond_Click()  
    If (Pos > 1) Then  
        SSTab.Tab = 1  
    Else  
        SSTab.Tab = 0  
    End If  
End Sub
```

```
Private Sub CmdBackJoin_Click()  
    SSTab.Tab = 0  
End Sub
```

```
Private Sub CmdClearCampo_Click()  
    ListCampos.Clear  
    ListGroup.Clear  
    CmpAgr = False  
End Sub
```

```
Private Sub CmdClearCon_Click()  
    CmbOpeLC.Enabled = False  
    ListCondiciones.Clear  
End Sub
```

```
Private Sub CmdClearJoin_Click()  
    ListJoin.Clear  
End Sub
```

```
Private Sub CmdDatos_Click()  
    Dim vSql, i, j, a As Field  
    On Error GoTo EtError  
    Call CON1  
    ds.Open conex  
    cmd.CommandText = Replace(Trim(TxtSql), vbCrLf, " ")  
    rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic  
    i = 0: j = 0  
    MSFDatos.Clear  
    MSFDatos.Rows = 2  
    MSFDatos.Cols = 1  
    If Not rs.EOF Then  
        While Not rs.EOF  
            MSFDatos.Rows = MSFDatos.Rows + 1  
            For Each a In rs.Fields
```

```

        j = j + 1
        If i = 0 Then
            MSFDatos.Cols = MSFDatos.Cols + 1
            MSFDatos.TextMatrix(0, j - 1) = a.Name
        End If
        MSFDatos.TextMatrix(MSFDatos.Rows - 2, j - 1) = IIf(IsNull(a.Value), " ",
a.Value)
        Next
        rs.MoveNext
        i = i + 1
        j = 0
    Wend
Else
    MsgBox "No se hayo ningun registro para la consulta establecida", vbInformation,
"Atencion!"
End If
rs.Close
ds.Close
MSFDatos.Cols = MSFDatos.Cols - 1
Exit Sub
EtError:
ds.Close
MsgBox "Error en su consulta SQL", vbInformation, "Atencion!"
End Sub

Private Sub CmdNext1_Click()
    Dim i As Long
    InitTab
    If (ListTablas.SelCount > 0) Then
        ds.Open conex
        For i = 0 To ListTablas.ListCount - 1
            If (ListTablas.Selected(i)) Then
                addTab (ListTablas.List(i))
                rs.Open SqlColumns & ListTablas.List(i), ds, adOpenDynamic,
adLockBatchOptimistic
                While Not rs.EOF
                    addCmp ListTablas.List(i), rs.Fields(0), rs.Fields(1), rs.Fields(2), rs.Fields(3)
                    rs.MoveNext
                Wend
                rs.Close
            End If
        Next
        ds.Close

        If (Pos >= 1) Then
            SSTab.TabEnabled(5) = True
            SSTab.Tab = 5
        End If
    End If
End Sub

```

```
End If
Else
    MsgBox "Para poder continuar, por lo menos debe seleccionar una tabla",
vbInformation, "Atencion!"
End If
```

```
End Sub
```

```
Private Sub CmdNextCampo_Click()
    SSTab.Tab = 4
    SSTab.TabEnabled(4) = True
End Sub
```

```
Private Sub CmdNextCond_Click()
    SSTab.Tab = 3
    SSTab.TabEnabled(3) = True
End Sub
```

```
Private Sub CmdNextJoin_Click()
    SSTab.Tab = 2
    SSTab.TabEnabled(2) = True
End Sub
```

```
Private Sub Form_Load()
    Dim i As Long
    Load_Sql_To_List ListTablas, SqlTab, 0
    For i = 1 To 5
        SSTab.TabEnabled(i) = False
    Next
    CmdClearJoin_Click
    CmdClearCon_Click
    CmdClearCampo_Click
    Toolbar1.Buttons(2).Enabled = False
    Toolbar1.Buttons(3).Enabled = False
    CmbConAlia.Enabled = False
End Sub
```

```
Private Sub ListTablas_Click()
    Dim i, d, a, f As Long
    Dim j, e, c As Long
    Dim Ntab1, Ctab1, Ntab2, Ctab2 As String

    If (ListTablas.SelCount > 0) Then
        Pos = 0
        ds.Open conex
```

```

For i = 0 To ListTablas.ListCount - 1
  If (ListTablas.Selected(i)) Then
    addTab (ListTablas.List(i))
    rs.Open SqlColumns & ListTablas.List(i), ds, adOpenDynamic,
adLockBatchOptimistic
    While Not rs.EOF
      addCmp ListTablas.List(i), rs.Fields(0), rs.Fields(1), rs.Fields(2), rs.Fields(3)
      rs.MoveNext
    Wend
    rs.Close
  End If
Next
ds.Close
ComTab
End If
End Sub

```

```

Private Sub SSTab_Click(PreviousTab As Integer)

```

```

  Dim i, j
  Select Case SSTab.Tab
    Case 0: 'TABLAS
      Form_Load
    Case 1: 'JOIN
      CmbTab1.Clear
      CmbTab2.Clear
      If Pos > 1 Then
        For i = 1 To Pos
          CmbTab1.AddItem Alias(i).tNombre
          CmbTab2.AddItem Alias(i).tNombre
        Next
        CmbTab1.ListIndex = 0
        CmbTab2.ListIndex = 1
      End If
    Case 2: 'WHERE
      CmbTabC.Clear
      If Pos > 0 Then
        For i = 1 To Pos
          CmbTabC.AddItem Alias(i).tNombre
        Next
        CmbTabC.ListIndex = 0
      End If
    Case 3: 'CAMPOS
      CmpAgr = False
      CmbTabF.Clear
      If Pos > 0 Then
        For i = 1 To Pos
          CmbTabF.AddItem Alias(i).tNombre
        Next
      End If
    End Select

```

```

Next
CmbTabF.ListIndex = 0
End If
Case 4: 'RESULTADOS
TxtSql = "SELECT" + vbCrLf
'***** CAMPOS
If (ListCampos.ListCount > 0) Then
    For i = 1 To ListCampos.ListCount
        TxtSql = TxtSql + ListCampos.List(i - 1) + ", "
    Next
    TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf
Else
    TxtSql = TxtSql + "*" + vbCrLf
End If

'***** TABLAS
TxtSql = TxtSql + "FROM" + vbCrLf
If (ListAlias.ListCount > 0) Then
    For i = 1 To ListAlias.ListCount
        TxtSql = TxtSql + ListAlias.List(i - 1) + ", "
    Next
Else
    For i = 1 To Pos
        TxtSql = TxtSql + Tablas(i).tNombre + ", "
    Next
End If
TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf

'***** JOIN
If (ListJoin.ListCount > 0) Then
    TxtSql = TxtSql + "WHERE ( " + vbCrLf
    For i = 1 To ListJoin.ListCount
        TxtSql = TxtSql + ListJoin.List(i - 1) + vbCrLf + "AND "
    Next
    TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 4) + ")"
End If

'***** WHERE
If (ListCondiciones.ListCount > 0) Then
    If (ListJoin.ListCount = 0) Then
        TxtSql = TxtSql + "WHERE" + vbCrLf
    Else
        TxtSql = TxtSql + "AND "
    End If
    For i = 1 To ListCondiciones.ListCount
        TxtSql = TxtSql + ListCondiciones.List(i - 1) + " "
    Next

```

```

        'TxtSql = TxtSql + ")"
    End If
    '***** GROUP BY
    If (ListGroup.ListCount > 0) Then
        TxtSql = TxtSql + vbCrLf + "GROUP BY"
        For i = 1 To ListGroup.ListCount
            TxtSql = TxtSql + vbCrLf + ListGroup.List(i - 1) + ", "
        Next
        TxtSql = Mid(TxtSql, 1, Len(TxtSql) - 2) + vbCrLf
    End If

    Toolbar1.Buttons(2).Enabled = True
    Toolbar1.Buttons(3).Enabled = True
    Toolbar1.Buttons(4).Enabled = True
Case 5: 'WHERE
    CmbAlias.Clear
    If Pos > 0 Then
        For i = 1 To Pos
            CmbAlias.AddItem Tablas(i).tNombre
        Next
        CmbAlias.ListIndex = 0
    End If

End Select
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    Dim f, i, j, Linea, Abrir
    Select Case Button.Index
        Case 1: SSTab.Tab = 0
            CmdClearJoin_Click
            CmdClearCon_Click
            CmdClearCampo_Click
            CmbLimDato_Click
            CmbLimAlia_Click
            CmbConAlia.Enabled = False
        Case 2:
            'Guardar Datos
            DLG.Filter = "Archivo Texto|*.txt"
            DLG.ShowSave
            If Trim(DLG.FileName) <> vbNullString Then
                Abrir = DLG.FileName
                f = FreeFile
                Open DLG.FileName For Output As f
                Print #f, vbCrLf
                Print #f, Tab(3); "Sistema de Informacion "
                Print #f, Tab(3); "Generador de Reportes Dinamico"
            End If
        End Select
    End Sub

```

```

' Print #f, Tab(3); "Reporte creado por: " + txtuserw
' Print #f, Tab(3); "Fecha      : " + FormatDateTime(Date, vbLongDate)
' Print #f, Tab(3); "Hora      : " + CStr(Time) + vbCrLf
' With Me.MSFDatos
'   For i = 0 To .Rows - 1
'     Linea = ""
'     If i = 1 Then
'       For j = 0 To .Cols - 1
'         Linea = Linea + StrRepeat("_", MAXLONGCOL(Me.MSFDatos, j) +
1) + " "
'       Next
'       Print #f, Tab(3); Linea
'       Linea = ""
'     End If
'     For j = 0 To .Cols - 1
'       Linea = Linea + .TextMatrix(i, j) + StrRepeat(" ",
MAXLONGCOL(Me.MSFDatos, j) + 2 - Len(.TextMatrix(i, j)))
'     Next
'     Print #f, Tab(3); Linea
'   Next
' End With
' Close f
' If MsgBox("Desea ver el contenido de este archivo .TXT?", vbQuestion +
vbYesNo, "Atencion") = vbYes Then
'   Shell "notepad " + Abrir, vbNormalFocus
' End If
' End If
' Case 3:
'   'Guardar SQL
'   DLG.Filter = "Archivo Sql|*.sql"
'   DLG.ShowSave
'   If Trim(DLG.FileName) <> vbNullString Then
'     Abrir = DLG.FileName
'     f = FreeFile
'     Open DLG.FileName For Output As f
'     Print #f, TxtSql
'     Close f
'     If MsgBox("Desea ver el contenido de este archivo .SQL?", vbQuestion +
vbYesNo, "Atencion") = vbYes Then
'       Shell "notepad " + Abrir, vbNormalFocus
'     End If
'   End If
' Case 4: Unload Me
' End Select
'End Sub

```

```

Function MAXLONGCOL(MSF As MSFlexGrid, ByVal COL As Long) As Long
    Dim i
    MAXLONGCOL = 0
    With MSF
        For i = 0 To .Rows - 1
            If Len(.TextMatrix(i, COL)) > MAXLONGCOL Then
                MAXLONGCOL = Len(.TextMatrix(i, COL))
            End If
        Next
    End With
End Function

```

```

Function StrRepeat(ByVal Caracter As String, ByVal Cantidad As Long) As String
    Dim i
    StrRepeat = ""
    For i = 1 To Cantidad
        StrRepeat = StrRepeat + Caracter
    Next
End Function

```

```

Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
    Dim f, i, j, Linea, Abrir
    Select Case Button.Index
        Case 1: SSTab.Tab = 0
            CmdClearJoin_Click
            CmdClearCon_Click
            CmdClearCampo_Click
            CmbLimDato_Click
            CmbLimAlia_Click
            CmbConAlia.Enabled = False
        Case 2:
            'Guardar Datos
            DLG.Filter = "Archivo Texto|*.txt"
            DLG.ShowSave
            If Trim(DLG.FileName) <> vbNullString Then
                Abrir = DLG.FileName
                f = FreeFile
                Open DLG.FileName For Output As f
                Print #f, vbCrLf
                Print #f, Tab(3); "Sistema de Informacion "
                Print #f, Tab(3); "Generador de Reportes Dinamico"
                Print #f, Tab(3); "Reporte creado por: LUIS L. BARROS"
                Print #f, Tab(3); "Fecha          : " + FormatDateTime(Date, vbLongDate)
            End If
        End Select
End Sub

```

```

Print #f, Tab(3); "Hora          : " + CStr(Time) + vbCrLf
With Me.MSFDatos
  For i = 0 To .Rows - 1
    Linea = ""
    If i = 1 Then
      For j = 0 To .Cols - 1
        Linea = Linea + StrRepeat("_", MAXLONGCOL(Me.MSFDatos, j) +
1) + " "
      Next
      Print #f, Tab(3); Linea
      Linea = ""
    End If
    For j = 0 To .Cols - 1
      Linea = Linea + .TextMatrix(i, j) + StrRepeat(" ",
MAXLONGCOL(Me.MSFDatos, j) + 2 - Len(.TextMatrix(i, j)))
    Next
    Print #f, Tab(3); Linea
  Next
End With
Close f
If MsgBox("Desea ver el contenido de este archivo .TXT?", vbQuestion +
vbYesNo, "Atencion") = vbYes Then
  Shell "notepad " + Abrir, vbNormalFocus
End If
End If
Case 3:
'Guardar SQL
DLG.Filter = "Archivo Sql|*.sql"
DLG.ShowSave
If Trim(DLG.FileName) <> vbNullString Then
  Abrir = DLG.FileName
  f = FreeFile
  Open DLG.FileName For Output As f
  Print #f, TxtSql
  Close f
  If MsgBox("Desea ver el contenido de este archivo .SQL?", vbQuestion +
vbYesNo, "Atencion") = vbYes Then
    Shell "notepad " + Abrir, vbNormalFocus
  End If
End If
Case 4: Unload Me
End Select
End Sub

```

IMPOR/EXPOR

```
Sub Limpiar()
```

```
    Dim i
```

```
    For i = 1 To ListTablas.ListCount
```

```
        ListTablas.Selected(i - 1) = False
```

```
    Next
```

```
    PBar.Value = 0.0001
```

```
End Sub
```

```
Private Sub CmbTab_Click()
```

```
    Load_Sql_To_List ListCmp, SqlColumns + " " + CmbTab, 0
```

```
End Sub
```

```
Private Sub CmdExport_Click()
```

```
    Dim i, total, indice, vCmp As Field, Linea, f
```

```
    Dim a As Field
```

```
    Call CON1
```

```
    ds.Open conex
```

```
    total = 0
```

```
    PBar.Value = 0.0001
```

```
    InitTab
```

```
    If ListTablas.SelCount > 0 Then
```

```
        For i = 1 To ListTablas.ListCount
```

```
            If ListTablas.Selected(i - 1) Then
```

```
                addTab (ListTablas.List(i - 1))
```

```
                cmd.CommandText = "select count(*) from " & ListTablas.List(i - 1)
```

```
                rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
```

```
                total = total + rs.Fields(0)
```

```
                rs.Close
```

```
            End If
```

```
        Next
```

```
        indice = 0
```

```
        For i = 1 To Pos
```

```
            Set rs = ds.Execute("select * from " & Tablas(i).tNombre & " ")
```

```
            If Not rs.EOF Then
```

```
                Dlg.Filter = "Archivo Imex|*.imex"
```

```
                Dlg.FileName = Tablas(i).tNombre
```

```
                AUDITORIA.Command2.Enabled = False
```

```

If Tablas(i).tNombre = "SIAPAUDITORIAS" Then
SWCLICK = 1
AUDITORIA.Command2.Enabled = True
End If

Dlg.ShowSave
If Trim(Dlg.FileName) <> vbNullString Then
f = FreeFile
Open Dlg.FileName For Output As f
While Not rs.EOF
Linea = ""
For Each a In rs.Fields
Linea = Linea & "" & IIf(IsNull(a.Value), " ", a.Value) & ","
Next
Print #f, Mid(Linea, 1, Len(Linea) - 1)
indice = indice + 1
rs.MoveNext
Wend
Close f
PBar.Value = Cint((indice / total) * 100)
MsgBox "Proceso de la tabla " + Tablas(i).tNombre + " ha terminado
satisfactoriamente ", vbInformation, "Atencion"
If MsgBox("Desea ver el CONItenido de este archivo?", vbQuestion +
vbYesNo, "Atencion") = vbYes Then
Shell "notepad " + Dlg.FileName, vbNormalFocus
End If
End If

End If
rs.Close
Next
Limpiar
Else
MsgBox "Debe selecciona por lo menos una tabla para CONItinuar CONI la
operacion", vbInformation, "Atencion"
End If
ds.Close

End Sub

Private Sub CmdFind_Click()
Dlg.Filter = "Archivos de Datos Import Export|*.imex"
Dlg.FileName = ""
Dlg.ShowOpen

```

```

If Trim(Dlg.FileName) <> vbNullString Then
    If (Trim(Dlg.FileName) = Trim(TxtFileOut)) Then
        MsgBox "No puede seleccionar este archivo ya lo esta usando para almacenar los
errores"
    Else
        TxtFileIn = Dlg.FileName
    End If
End If
End Sub

```

```

Private Sub CmdImport_Click()

```

```

*****

```

```

*****
' verificamos la existencia de los archivos
If Trim(TxtFileOut) = vbNullString Then
    MsgBox "No se ha definido el archivo para almacenar los errores en caso de haberlo",
vbInformation, "Atencion"
    Exit Sub
End If
If Trim(TxtFileIn) = vbNullString Then
    MsgBox "No se ha definido el archivo para cargar los datos", vbInformation,
"Atencion"
    Exit Sub
End If
CmdImport.Enabled = False
FileOut = TxtFileOut
If ExistFile(FileOut) Then
    Kill FileOut
End If
If Not ExistFile(TxtFileIn) Then
    MsgBox "No se encontro el archivo especificado para cargar los datos",
vbInformation, "Atencion"
    Exit Sub
End If

```

```

*****

```

```

*****

```

```

' ** cargamos los datos de la tabla seleccionada
InitTab
addTab CmbTab

Call CON1
ds.Open conex
cmd.CommandText = SqlColumns + " " + CmbTab
rs.Open cmd.CommandText, ds, adOpenDynamic, adLockBatchOptimistic
While Not rs.EOF

```

```

        addCmp CmbTab, rs.Fields(0), rs.Fields(1), rs.Fields(2), rs.Fields(3)
        rs.MoveNext
    Wend
    ds.Close

' verificar cantidad de campos de la tabla y sus valores
If VerificarCmp(TxtFileIn, FileOut) Then
    LoadDB
    MsgBox "Los datos han sido cargados satisfactoriamente.", vbInformation, "Atencion"
    PIE.Value = 0.0001
ElseIf MsgBox("Se han encontrado errores en el archivo estos han sido almacenados
en:" + vbCrLf + TxtFileOut + vbCrLf + "Desea ver el contenido de este archivo?",
vbQuestion + vbYesNo, "Atencion") = vbYes Then
    Shell "notepad " + TxtFileOut, vbNormalFocus
End If
CmdImport.Enabled = True
TxtFileOut = ""
TxtFileIn = ""
CmbTab.ListIndex = 0
End Sub

Private Sub CmdSave_Click()
    Dlg.Filter = "Archivos de Errores Import Export|*.err"
    Dlg.FileName = ""
    Dlg.ShowSave
    If Trim(Dlg.FileName) <> vbNullString Then
        If (Trim(Dlg.FileName) = Trim(TxtFileIn)) Then
            MsgBox "No puede seleccionar este archivo ya lo esta usando extraer los datos"
        Else
            TxtFileOut = Dlg.FileName
        End If
    End If
End Sub

Private Sub Form_Load()
    Load_Sql_To_List ListTablas, SqlTab, 0
    Load_Sql_To_Cmb CmbTab, SqlTab, 0
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    Unload Me
End Sub

Private Sub LoadDB()
    Dim f, Linea, lista, nLinea
    f = FreeFile
    nLinea = 0

```

```

PIE.Value = 0.0001
Open TxtFileIn For Input As #f
While Not EOF(f)
    nLinea = nLinea + 1
    Line Input #f, Linea
    Linea = Trim(Linea)
    If Linea <> vbNullString Then
        ds.Open conex
        ds.Execute ("insert into " + CmbTab + " values(" + Linea + ")")
        ds.Close
    End If
    PIE.Value = CInt((nLinea / NumeroLineas) * 100)
Wend
Close #f
End Sub

```

```

Function NewLinea(ByVal Linea As String) As String
    Dim cLinea, lista, i
    cLinea = Mid(Linea, 2, Len(Linea) - 2)
    lista = Split(cLinea, ",")
    For i = 0 To UBound(lista)
        If IsDate(lista(i)) Then
            NewLinea = "" + FormatFecha2(lista(i)) + ","
        Else
            NewLinea = DateTime(lista(i))
        End If
    Next
    NewLinea = Mid(NewLinea, 1, Len(NewLinea) - 1)
End Function

```

```

Private Sub Toolbar1_Click()
Unload Me
End Sub

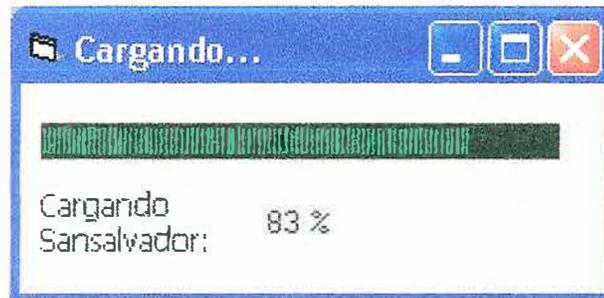
```

**SISTEMA DE INFORMACION ACADEMICO-
ADMINISTRATIVO PARA LA GESTION DE LOS DATOS EN
LOS DEPARTAMENTOS DE MATRICULAS Y PROCESOS
ACADEMICOS EN LA INSTITUCION EDUCATIVA
DISTRITAL SAN SALVADOR DE BARRANQUILLA**

MANUAL DEL USUARIO

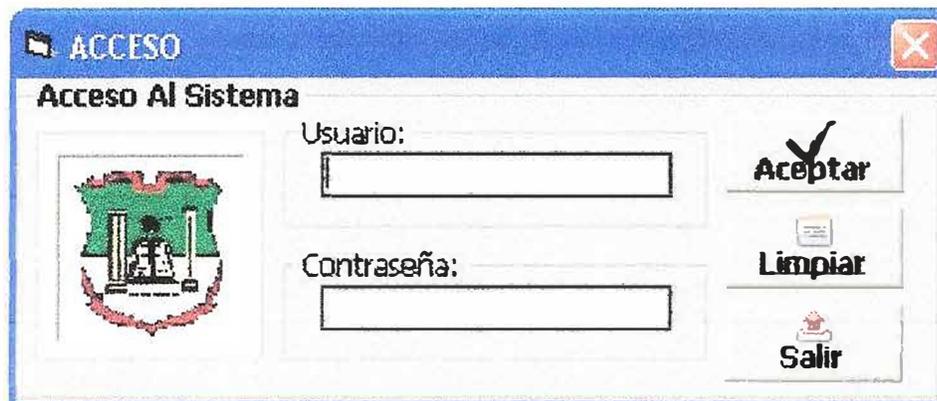
**CORPORACION EDUCATIVA MAYOR DEL DESARROLLO
SIMON BOLIVAR
FACULTAD DE INGENIERIA DE SISTEMAS
AREA DE INVESTIGACION FORMATIVA
2005**





PANTALLA DE CARGA DEL SISTEMA

Esta pantalla nos permite mostrar la carga del sistema con su conexión a la base de datos, toda su información para luego ingresar al sistema.



PANTALLA DE ACCESO AL SISTEMA

INDICE

PANTALLA DE CARGA DEL SISTEMA -----	1
PANTALLA DE ACCESO AL SISTEMA -----	1
PANTALLA SUB MENU DATOS BASICOS -----	3
PANTALLA DE USUARIOS -----	4
PANTALLA DE AREAS -----	5
PANTALLA DE ASIGNATURAS -----	6
PANTALLA SUB MENU DEL ITEM CURSOS -----	8
PANTALLA DE CURSOS -----	9
PANTALLA DE ASIGNACION DE CURSOS -----	10
PANTALLA DE REGISTROS DE DOCENTES Y ASIGNACION DE ASIGNATURAS -----	12
PANTALLA DE CREACION DE GRUPOS -----	14
PANTALLA DE LOGROS -----	16
PANTALLA DE PENSUM Y ASIGNACION DE ASIGNATURAS -----	18
PANTALLA DE PROFESIONES -----	19



PANTALLA DE ESTUDIANTES EN LA PESTAÑA ESTUDIANTES	21
PANTALLA DE ESTUDIANTES EN LA PESTAÑA ACUDIENES	22
PANTALLA SUB MENU PROCESOS	25
PANTALLA DE EVALUACIONES	26
PANTALLA DE MATRICULAS	27
PANTALLA SUB MENU SISTEMAS	29
PANTALLA DE GENERADOR DE REPORTE	30
PANTALLA DE AUDITORIAS	36
PANTALLA DE EXPORT / IMPORT	39
PANTALLA DE AYUDAS EN LINEA	41

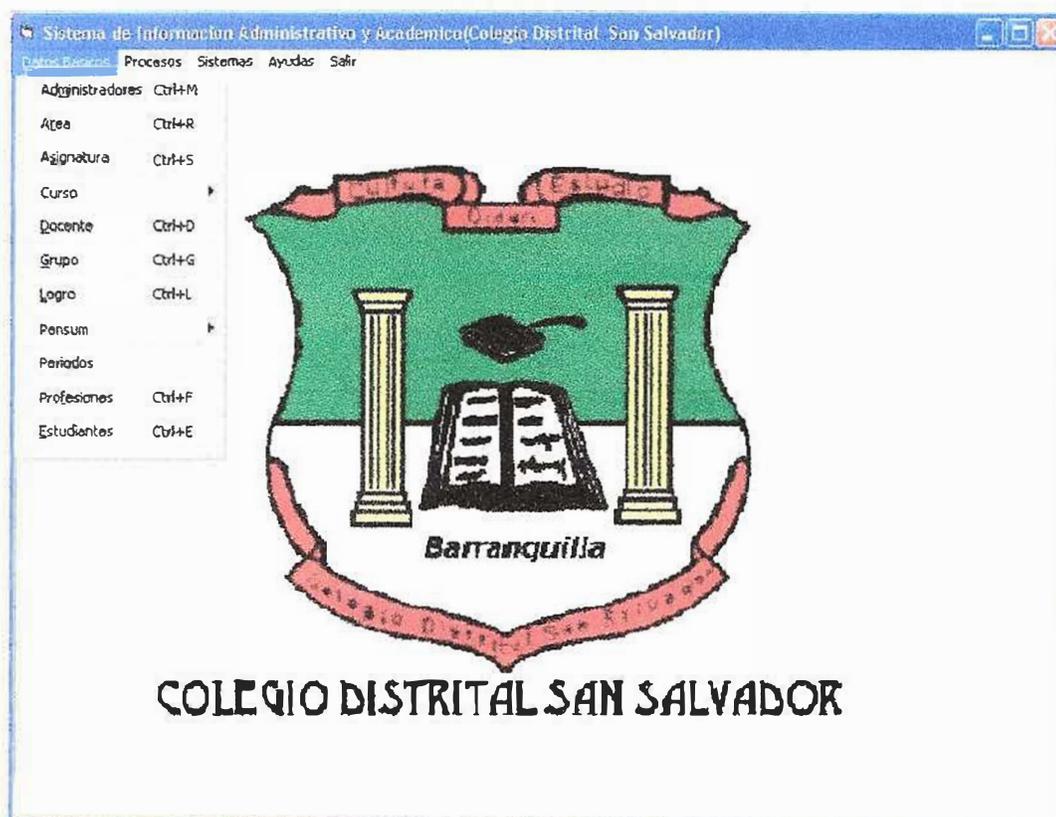
A través de esta pantalla podemos ingresar al sistema utilizando un usuario y una contraseña existente. La pantalla posee tres botones que son:

Aceptar: Permite el ingreso al sistema.

Limpiar: Limpia las cajas de texto de la pantalla

Salir: Cierra la ventana de acceso

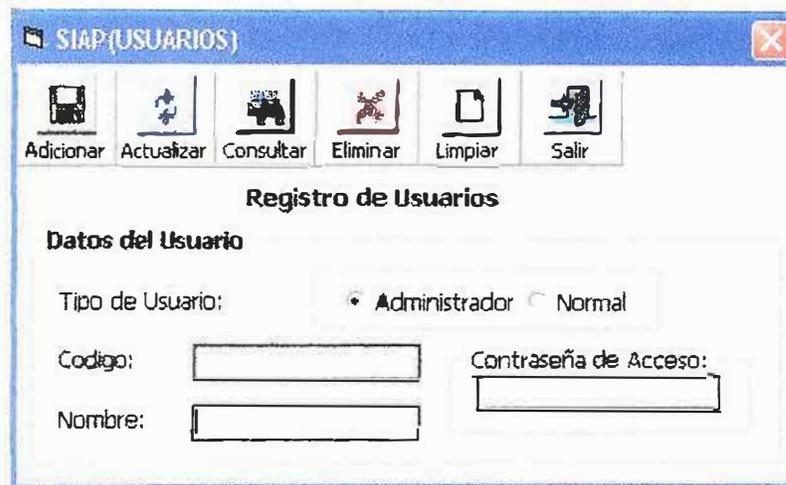
Al momento de ingresar un usuario con una contraseña invalida, se muestra un mensaje de caja de “Acceso Denegado”, dado el caso de que el usuario no exista o que la contraseña sea incorrecta.



Pantalla del sistema con el submenú Datos Básicos.

Esta pantalla muestra el sistema de información general con el acceso a cada uno de los submenús. En este caso muestra el submenú Datos Básicos, que su función principal es realizar la carga de todos los datos que se manejen en la institución. Este muestra las opciones de Administradores, Áreas, Asignaturas, Curso, Docente, Grupo, Logro, Pensum, Profesiones y Estudiantes.





Pantalla Usuarios

Esta pantalla permite crear los usuarios que van a trabajar con esta aplicación. Existen dos tipos de usuarios que pueden trabajar en el sistema, como son los administradores y los usuarios normales. Los usuarios administradores son los que pueden realizar operaciones sin que se presenten restricciones en el sistema, mientras que los usuarios normales son aquellos que van a realizar transacciones básicas del sistema, cumpliendo con las restricciones de seguridad del sistema.



Pantalla de áreas.

Esta pantalla permite llevar un registro de todas las áreas de enseñanza que trabaja la institución. Se registran por código, El nombre del área respectivo y el estado en que se encuentra. El botón examinar  permite consultar las asignaturas existentes. En la parte superior están las operaciones que se explican a continuación:

Adicionar: Permite guardar registros en la base de datos.

Actualizar: permite modificar un registro existente.

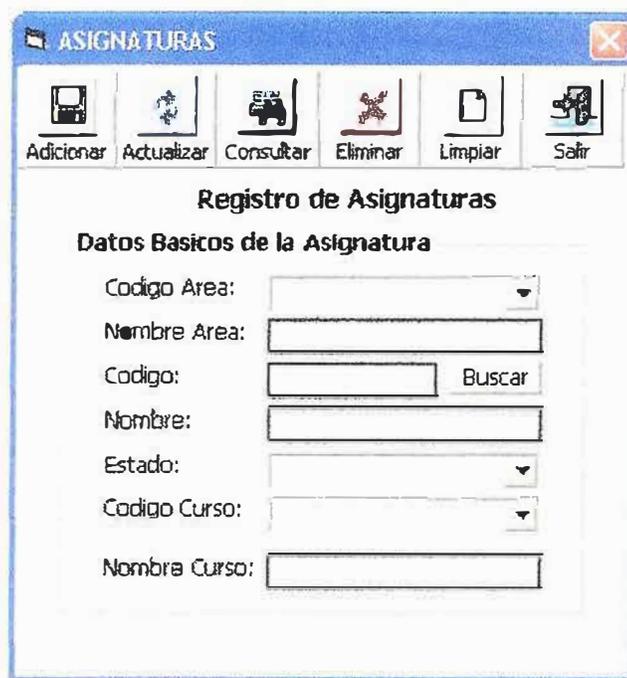
Consultar: Permite verificar si un registro existe o no existe.

Eliminar: Borra el registro actual de la base de datos.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime el registro que se encuentra en la pantalla.

Salir: Cierra la ventana y vuelve al menú principal.



The screenshot shows a window titled "ASIGNATURAS" with a toolbar containing six icons: Adicionar, Actualizar, Consultar, Eliminar, Limpiar, and Salir. Below the toolbar is a form titled "Registro de Asignaturas" with a sub-section "Datos Basicos de la Asignatura". The form contains the following fields:

- Codigo Area: [Dropdown menu]
- Nombre Area: [Text input field]
- Codigo: [Text input field] with a "Buscar" button to its right.
- Nombre: [Text input field]
- Estado: [Dropdown menu]
- Codigo Curso: [Dropdown menu]
- Nombre Curso: [Text input field]

Pantalla de Asignaturas

Esta pantalla permite llevar registros de todas las asignaturas que se dictan en la institución. En este caso las asignaturas registradas corresponden a un área de trabajo específica. Ej.: Las asignaturas de Ciencias Naturales y Química corresponden al área de Biología,

Geografía e Historia al área de Ciencias Sociales y así sucesivamente. En la parte superior están las operaciones que se explican a continuación:

Adicionar: Permite guardar una nueva asignatura.

Actualizar: permite modificar una asignatura existente.

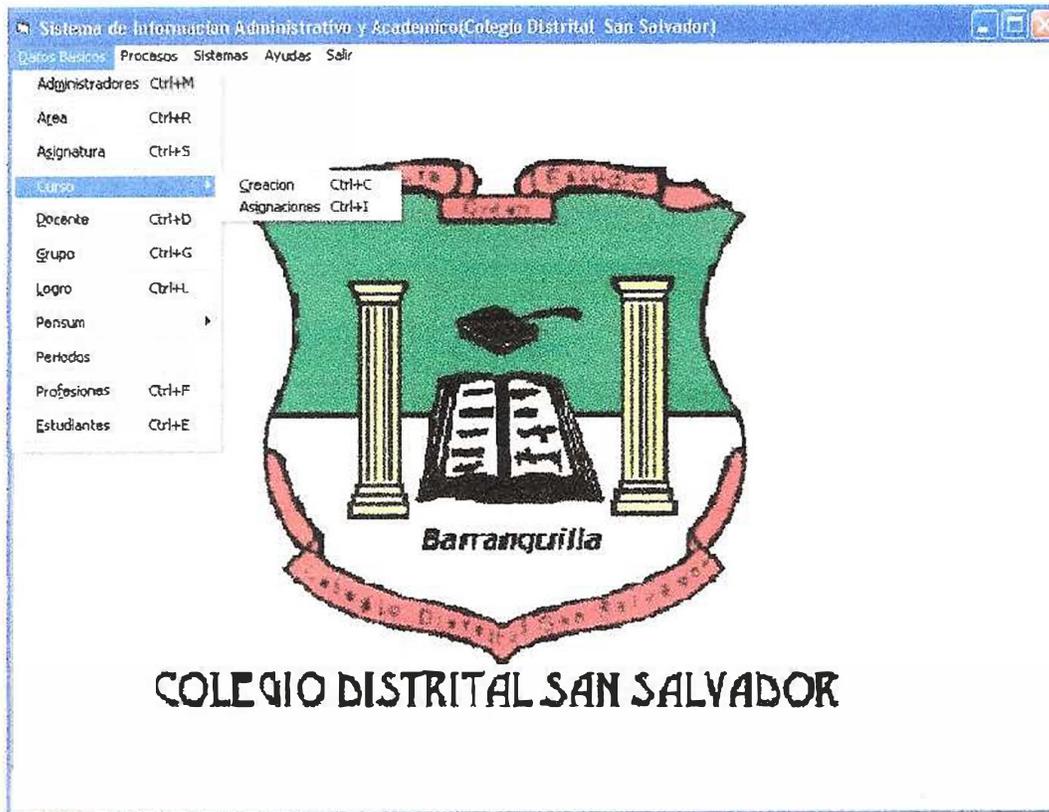
Consultar: Permite verificar si una asignatura existe o no existe.

Eliminar: Borra la asignatura actual de la base de datos.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime la asignatura que se encuentra en la pantalla.

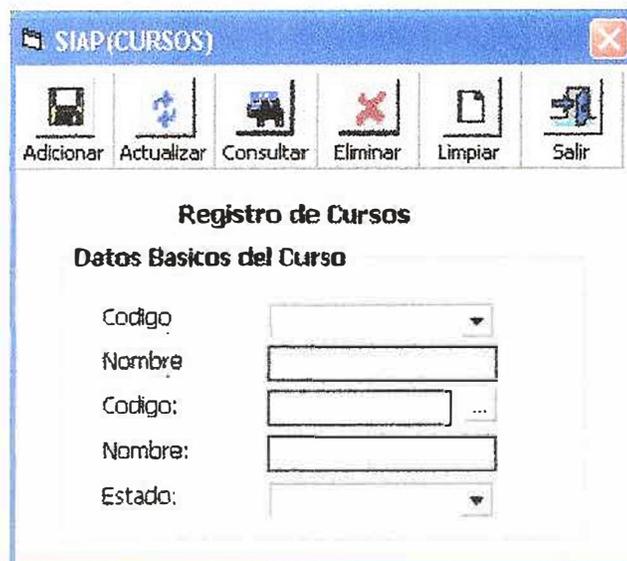
Salir: Cierra la ventana y vuelve al menú principal.



Submenú del ítem Cursos

Este es el submenú de cursos que presenta las opciones de creación y asignación.





Pantalla de creación de cursos

Esta pantalla permite llevar la creación de cursos en la institución.

El botón examinar  permite consultar los cursos que se encuentran creados en la institución por código. Se registra el nombre y el estado en que se encuentra. Igualmente se encuentra la barra de herramientas en la parte superior que permite realizar las siguientes operaciones:

Adicionar: Permite guardar un nuevo registro del curso.

Actualizar: permite modificar un curso existente.

Consultar: Permite verificar si un curso existe o no existe.

Eliminar: Borra el curso actual de la base de datos.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime el curso que se encuentra en la pantalla.

Salir: Cierra la ventana y vuelve al menú principal.

SIAP (ASIGNACION DE CURSOS)

Adicionar Actualizar Consultar Eliminar Limpiar Salir

Asignacion de Cursos

Asignaciones

Jornada:

Curso:

Estado:

Grupo:

Estado

Pantalla de Asignación de cursos

Esta pantalla permite realizar la asignación de los cursos a un grupo específico. Como en los casos anteriores, el botón examinar  permite consultar los cursos creados en la institución. Luego se asignan a los grupos existentes en la institución. Esta pantalla se explicara un poco mas adelante. Las cajas transparentes del lado de la derecha muestran el campo correspondiente a cada caja de texto.

Aquí se registra el curso, el grupo que va a ser asignado, (grupo A, grupo B, grupo C,...), la jornada de estudio, el año lectivo, y el código del pensum, que se explicara en la pantalla de pensum mas adelante.

Ahora vamos a la pantalla de docentes con la asignación de asignaturas.

SIAP (DOCENTES)

Adicionar Actualizar Consultar Eliminar Limpiar Salir

Registro de Docentes

Datos Basicos Del Docente

Id del docente: Tipo de

Apellidos: Telefono:

Nombres: Num. del Escalafon:

Fecha de Nac: 11/05/2005 Estado:

Direccion: Sexo:

Asignacion de Asignaturas:

Codigo	Nombres

Seleccionar

Asignar

Eliminar

Pantalla de Registro de Docentes y Asignación de Asignaturas.

Esta pantalla permite llevar registros de todos los docentes que laboran en la institución, teniendo en cuenta a su vez las asignaturas que les han sido asignadas para dictar en un año lectivo. La barra de herramientas trabaja las operaciones de la siguiente forma:

Adicionar: permite crear registros de los docentes. Dado el caso de que el docente exista, pero no tenga asignaturas asignadas, se realiza la consulta y después en la ventana de la parte inferior podemos asignarle las asignaturas correspondientes.

Antes de asignar las asignaturas primero se seleccionan oprimiendo el botón . Se selecciona el número de asignaturas correspondientes al docente, las asignaturas seleccionadas se van acumulando en la tabla y luego se oprime el botón .

Si se selecciono una asignatura que no corresponde a la lista o la asignatura no la dicta el docente actual, se oprime el botón , y la asignatura se elimina automáticamente de la lista.

Actualizar: Permite modificar el registro del docente, ya sea por información personal, o realizando las operaciones de asignación de asignaturas, ya sea Seleccionar, Asignar o Eliminar.

Consultar: Permite ver si un docente existe o no en la institución por ID. Si el registro existe, el registro se muestra con las asignaturas

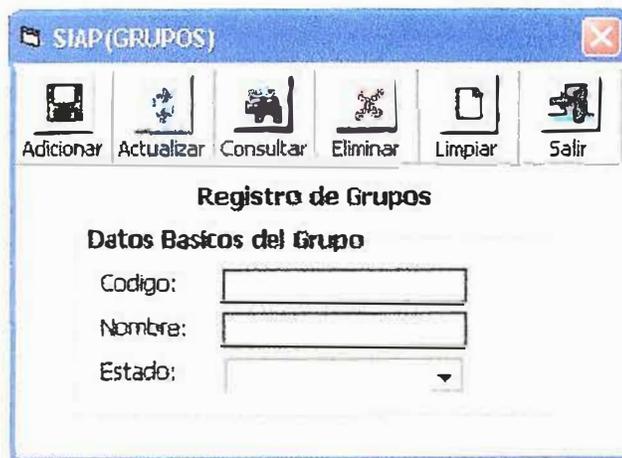
asignadas. Si el registro no existe, se muestra un mensaje de aviso diciendo que no existe registro.

Eliminar: Permite borrar el registro del docente junto con sus asignaturas asignadas, haciendo caso a la confirmación si realmente desea borrar el presente registro.

Limpiar: Permite limpiar las cajas de texto y la tabla donde se asignan las asignaturas.

Imprimir: Imprime el registro del docente y las asignaturas asignadas.

Salir: Cierra la ventana y vuelve al menú principal.



The screenshot shows a window titled "SIAP (GRUPOS)" with a menu bar containing six icons: Adicionar, Actualizar, Consultar, Eliminar, Limpiar, and Salir. Below the menu bar is a form titled "Registro de Grupos" with a sub-section "Datos Basicos del Grupo". The form contains three input fields: "Codigo:" with a text box, "Nombre:" with a text box, and "Estado:" with a dropdown menu.

Pantalla de creación de grupos.



Esta pantalla permite crear los grupos que hacen parte de la institución. Se registran por código, el nombre del grupo y el estado en que se encuentra el grupo. Igualmente la barra de la parte superior realiza las operaciones de:

Adicionar: Permite guardar un nuevo grupo.

Actualizar: permite modificar un grupo existente.

Consultar: Permite verificar si un grupo existe o no existe.

Eliminar: Borra el grupo actual de la base de datos.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime el grupo que se encuentra en la pantalla.

Salir: Cierra la ventana y vuelve al menú principal.

The screenshot shows a window titled "Logros" with a standard Windows-style title bar. Below the title bar is a toolbar with six icons and their corresponding labels: "Adicionar" (Add), "Actualizar" (Update), "Consultar" (Consult), "Eliminar" (Delete), "Limpiar" (Clear), and "Salir" (Exit). The main area of the window is titled "Datos Generales del Logro" and contains the following fields:

- "Codigo de La Asignatura": A text input field.
- "Codigo del Logro": A text input field.
- "Calificador": A dropdown menu with a blue icon and a downward arrow.
- "Descripcion": A large text input field.
- "Estado": A dropdown menu with "Activo" selected and a downward arrow.

To the right of the "Codigo del Logro" field, there is a legend for the "Calificador" values:

- [E]=Exelente
- [B]=Bueno
- [A]=Aceptable
- [D]=Deficiente
- [I]=Insuficiente

Pantalla de Logros.

En esta pantalla se registran los logros a evaluar correspondientes en las asignaturas. Los logros se registran por código, la descripción que en este caso es lo que se va a evaluar, el estado en que se encuentra el logro actual, ya sea activo o inactivo, y la asignatura a la cual pertenece ese logro. Si el logro a crear esta en una asignatura que no existe, entonces se procede a crear la asignatura en la pantalla de asignaturas. En la parte inferior se encuentra el código de la

asignatura que se extrae con el botón examinar  y también extrae el nombre de esa asignatura en la caja transparente de la derecha. Esta pantalla también posee la barra de funciones con los mismos procedimientos:

Adicionar: Permite guardar un nuevo logro.

Actualizar: permite modificar un logro existente.

Consultar: Permite verificar si un logro existe o no existe.

Eliminar: Borra el logro actual de la base de datos.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime el registro del logro actual que se encuentra en la pantalla.

Salir: Cierra la ventana y vuelve al menú principal.

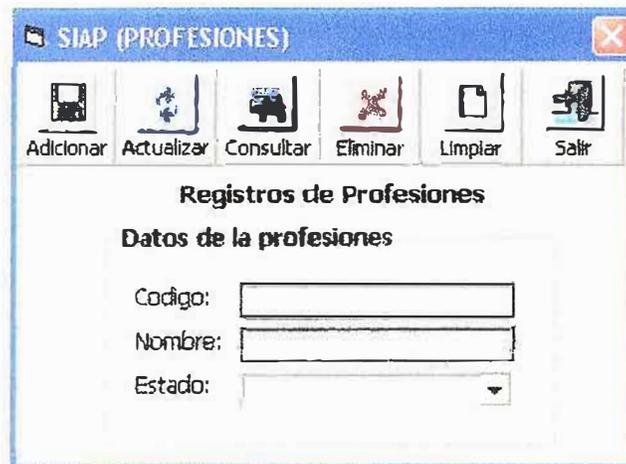
Ahora se muestra la pantalla de Pensum y Asignación de Asignaturas.



Pantalla de Creación de Pensum y Asignación de Asignaturas.

Esta pantalla permite crear los pensum correspondientes a cada año. El pensum se registra con un código, el nombre del pensum y el estado, ya sea activo o inactivo. Al momento de crear o modificar las asignaturas del pensum, si el pensum no existe, se crea y luego en la misma ventana se le asignan las asignaturas correspondientes. Con el

botón Seleccionar vemos el número de asignaturas que le vamos a asignar, se presiona el botón Asignar luego vamos a la barra de operaciones y le damos adicionar. De esta forma el pensum queda registrado con las asignaturas correspondientes. Si se va a suprimir una asignatura en cualquier caso solo seleccione la asignatura de la lista y oprime el botón Eliminar y automáticamente la asignatura queda eliminada de la lista.



Pantalla de Profesiones.

La función de esta pantalla es llevar el registro de todas las profesiones de cada uno de los docentes que laboran en la institución. La barra de operaciones hace las siguientes funciones:

Adicionar: Permite guardar una nueva profesión.

Actualizar: permite modificar una profesión existente.

Consultar: Permite verificar si una profesión existe o no existe.

Eliminar: Borra una profesión.

Limpiar: Limpia las cajas de texto de la ventana.

Imprimir: Imprime la profesión que se encuentra en la pantalla.

Salir: Cierra la ventana y vuelve al menú principal.

A continuación vamos a ver la pantalla de estudiantes y acudientes, que se encuentra dividida en dos pestañas totalmente relacionadas.



SIAP(ESTUDIANTES)

Adicionar Actualizar Consultar Eliminar Limpiar Imprimir Salir

ESTUDIANTES ACUDIENTES

Datos Basicos

Id estudiante:

Apellidos:

Nombres:

Fecha Nac:

Direccion:

Telefono:

Sexo:

Tipo De Docum:

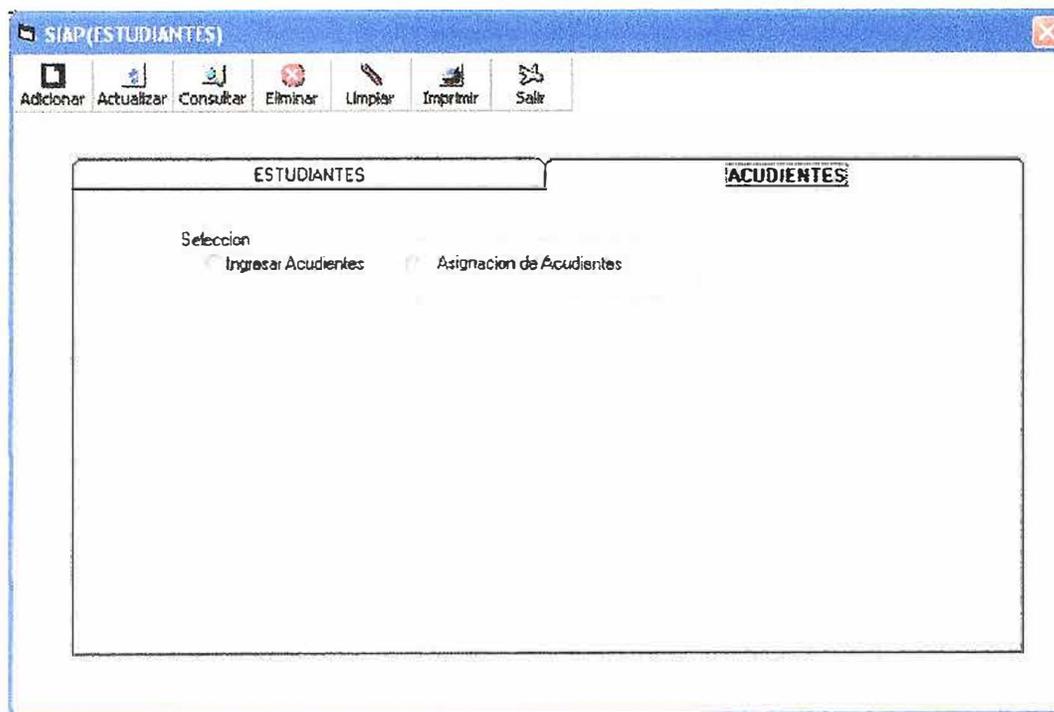
Estado:

Tipo de cliente:

Observaciones:

Pantalla Estudiantes con la pestaña de registro de Estudiantes.

Esta pantalla permite llevar un registro completo de todos los estudiantes y aspirantes a ingresar a la institución. Ahora en la siguiente página vamos a ver la pestaña de acudientes y miremos la función que realiza.



Pantalla de Estudiantes en la pestaña de acudientes.

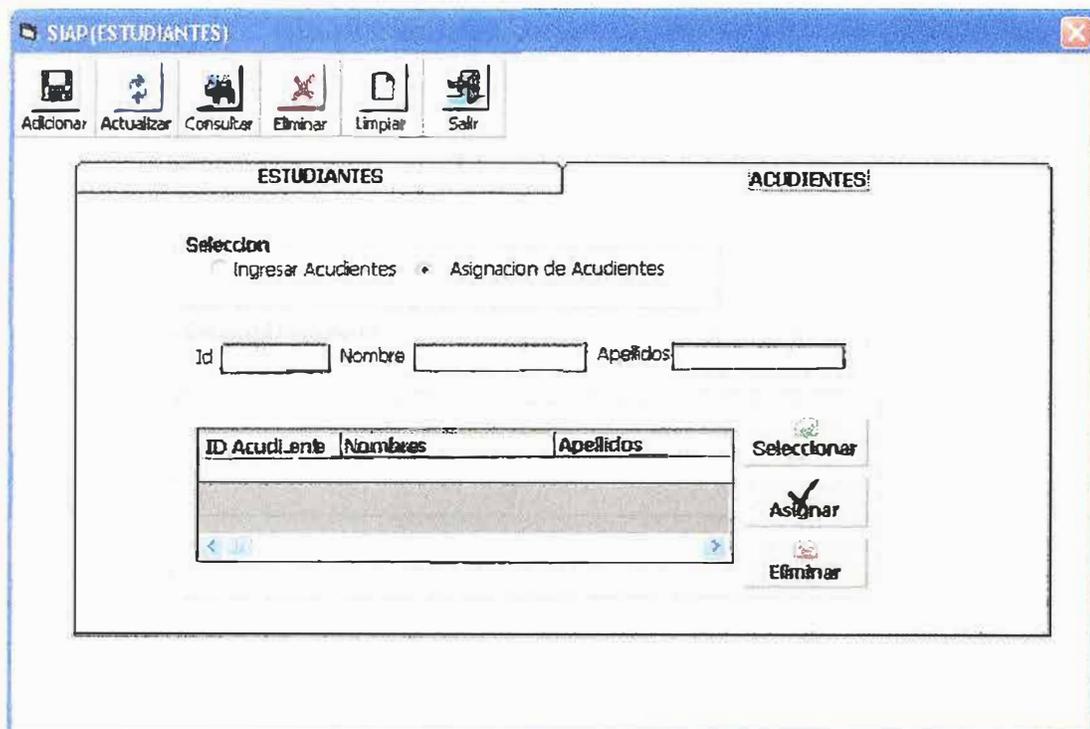
Esta pantalla permite realizar dos funciones principales, que son:

- El registro de acudientes.
- Asignación de acudientes a estudiantes.

Veamos la primera opción:

The screenshot shows a software window titled "SIAP(ESTUDIANTES)". At the top, there is a menu bar with icons and labels: "Adicionar", "Actualizar", "Consultar", "Eliminar", "Limpiar", "Imprimir", and "Salir". Below the menu bar, the main content area is divided into two sections: "ESTUDIANTES" and "ACUDIENES". The "ACUDIENES" section contains a "Selección" area with two radio buttons: "Ingresar Acudientes" (selected) and "Asignación de Acudientes". Below this is a "Datos Basicos Del Acudiente" form with the following fields: "Id:" (text input), "Telefono:" (text input), "Nombres:" (text input), "Fecha Nacimiento:" (text input), "Apellidos:" (text input), "Direccion:" (text input), "Sexo:" (dropdown menu with "Seleccione" selected), "Parentesco:" (dropdown menu with "Seleccione" selected), and "Estado:" (dropdown menu with "Activo" selected).

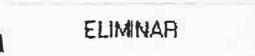
En la primera opción se hace el registro del acudiente con sus datos básicos. Con el botón examinar  hacemos la consulta de los acudientes que existen en la base de datos. Se registra el ID, nombres, apellidos, sexo, estado, teléfono, fecha de nacimiento, dirección y parentesco (relación familiar). Ahora veamos la segunda opción:



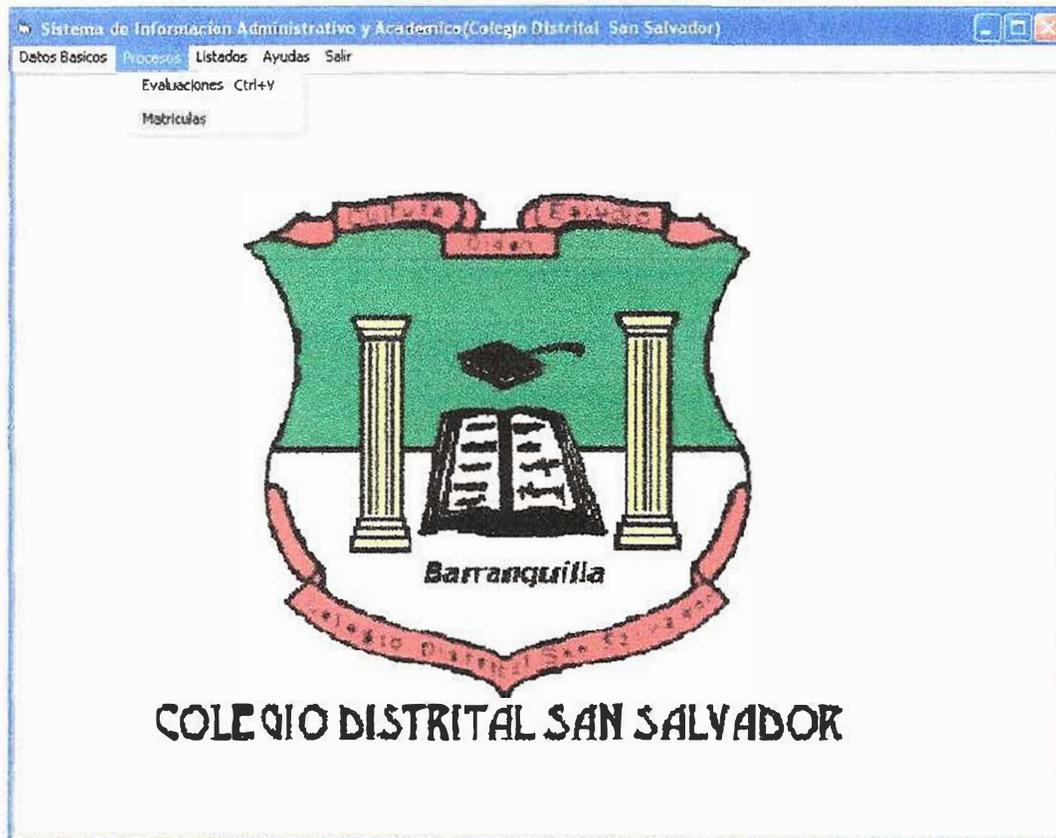
En esta opción de asignación de acudientes, se consulta el estudiante que le va a ser asignado su acudiente o sus acudientes. Una vez consultado el estudiante en la pestaña de estudiante, aparece automáticamente en la pestaña de acudientes, su ID, sus nombres y sus apellidos. Luego en la caja de abajo se encuentran las opciones de selección, asignación y eliminación de acudientes. Antes de asignar los acudientes primero se seleccionan oprimiendo el botón **SELECCIONAR**. Se selecciona el número de acudientes correspondientes al estudiante, los acudientes seleccionados se van

acumulando en la tabla y luego se oprime el botón 

Si se asigno un acudiente que no corresponde a la lista o el acudiente

ya no hace parte de la lista, se oprime el botón  y el

acudiente se elimina automáticamente de la lista.



Submenú Procesos

En este submenú se trabajan los procesos de Evaluaciones y de Matriculas, que se muestran a continuación.

Valoracion

Adicionar Actualizar Consultar Eliminar Limpiar Salir

Busqueda de Estudiantes por Jornada-Curso-Grupo

Jornada: Curso: Grupo:

Identificacion

Matricula: Año Lectivo:

Datos Generales del Periodo

Periodo Estado

Datos de la Valoracion (Logros por Asignaturas)

Asignaturas Agregar

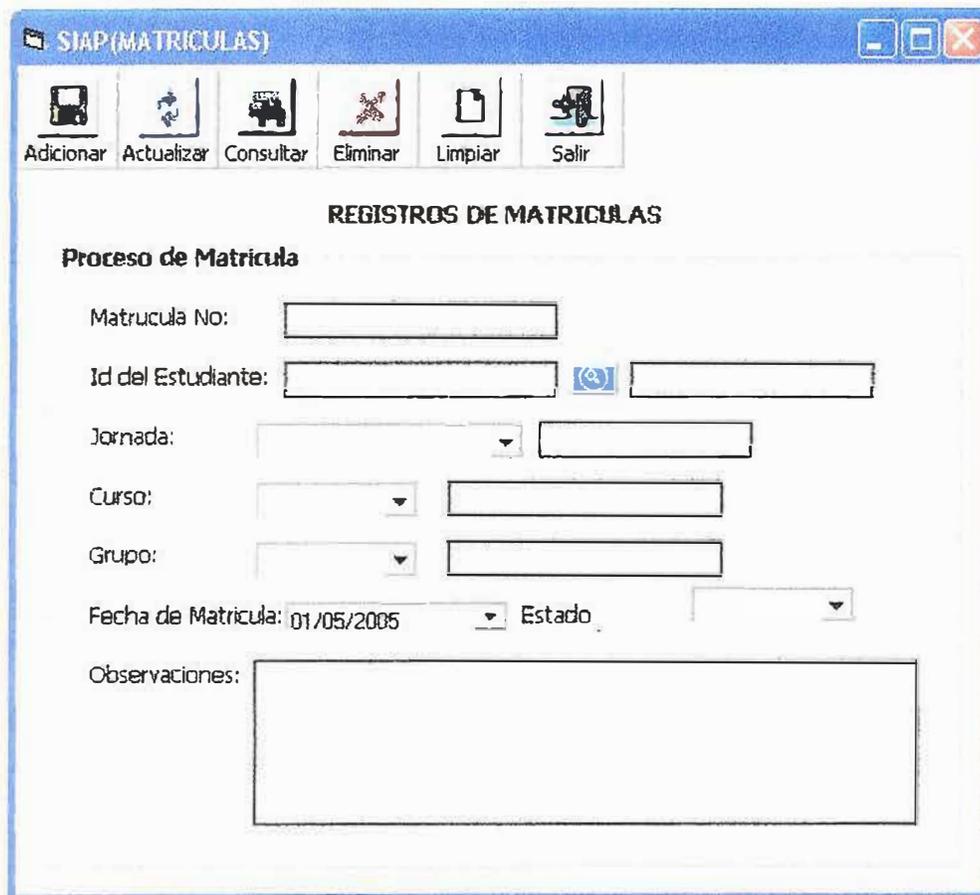
Cod Asig	Nombre	Cal	Descripcion

Eliminar Eliminar Todos

Observaciones

Pantalla de Evaluaciones





SIAP (MATRICULAS)

Adicionar Actualizar Consultar Eliminar Limpiar Salir

REGISTROS DE MATRICULAS

Proceso de Matricula

Matricula No:

Id del Estudiante:

Jornada:

Curso:

Grupo:

Fecha de Matricula: 01/05/2005 Estado

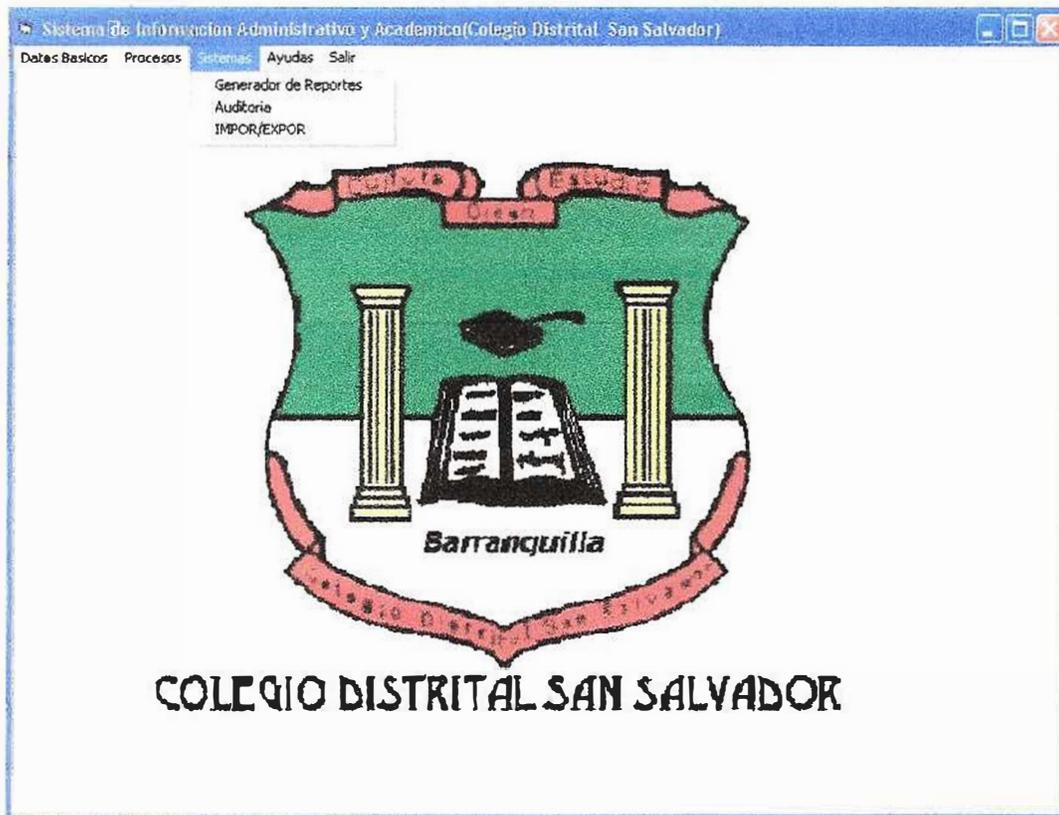
Observaciones:

Pantalla de Matriculas

En esta pantalla se realiza la matricula oficial del estudiante o aspirante que vaya a ingresar en la institución. La matricula tiene un numero que va en forma secuencial. El estudiante o aspirante debe de estar registrado en la base de datos, teniendo en cuenta que se registró en la pantalla de estudiantes, vista anteriormente. El botón

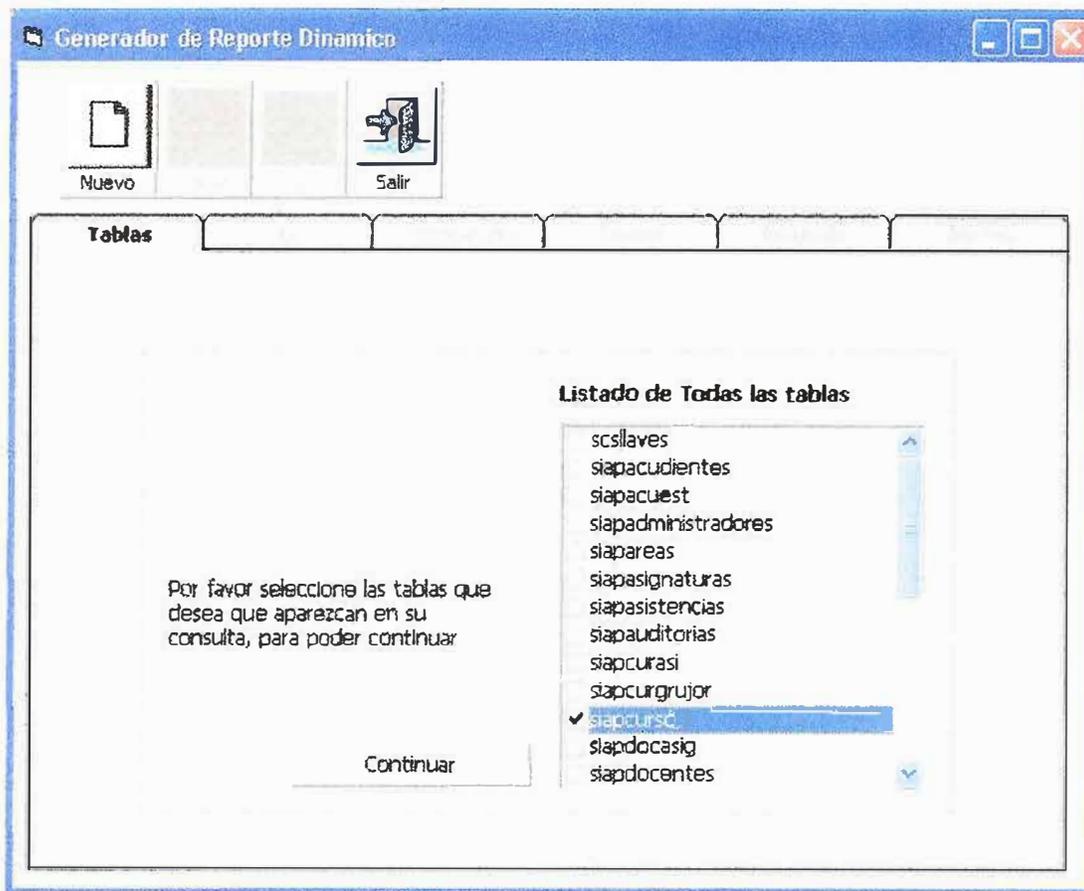


realiza la consulta de estudiantes registrados, una vez consultado aparece el nombre del estudiante del lado derecho del botón, en la caja transparente, se registra la jornada, el curso al que va ser asignado, el grupo a que pertenece ese curso, la fecha de matricula, el estado y las observaciones, que son casos pendientes al momento de realizar la matricula. ej.: Observaciones: falta documento de identidad, pendiente certificado de salud.



Submenú Sistema

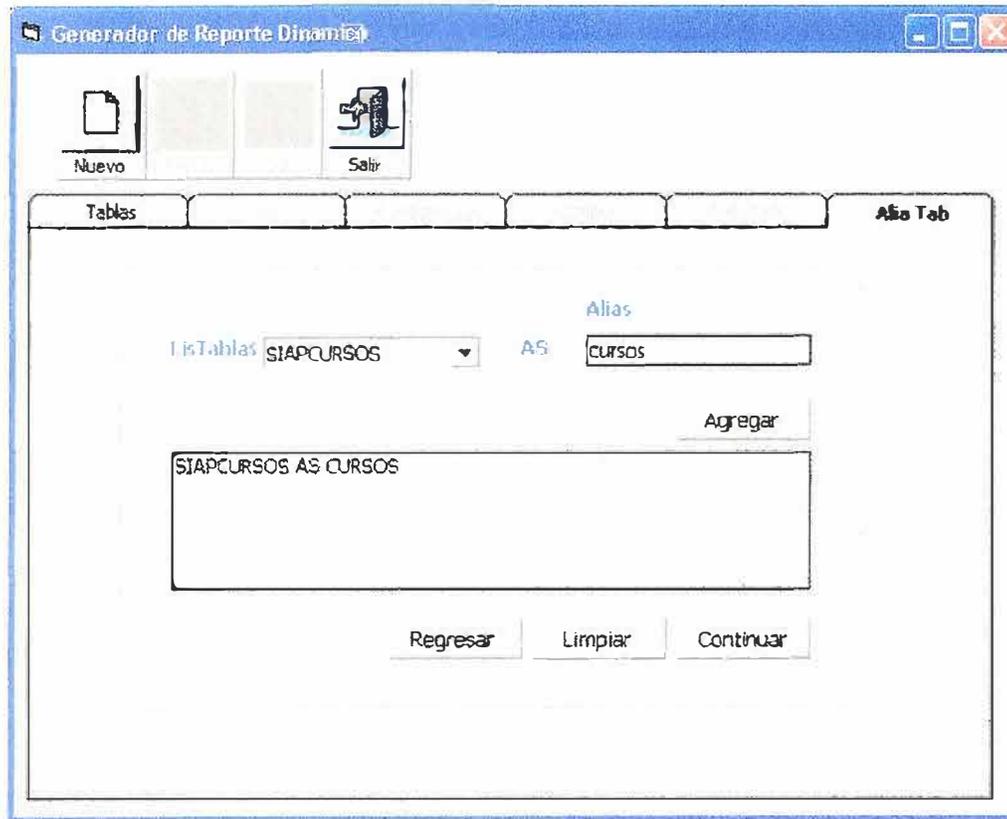
Este submenú permite realizar operaciones con el sistema y con la base de datos. Este submenú posee los ítems de: Generador de reportes, Auditoria e IMPOR/EXPOR.



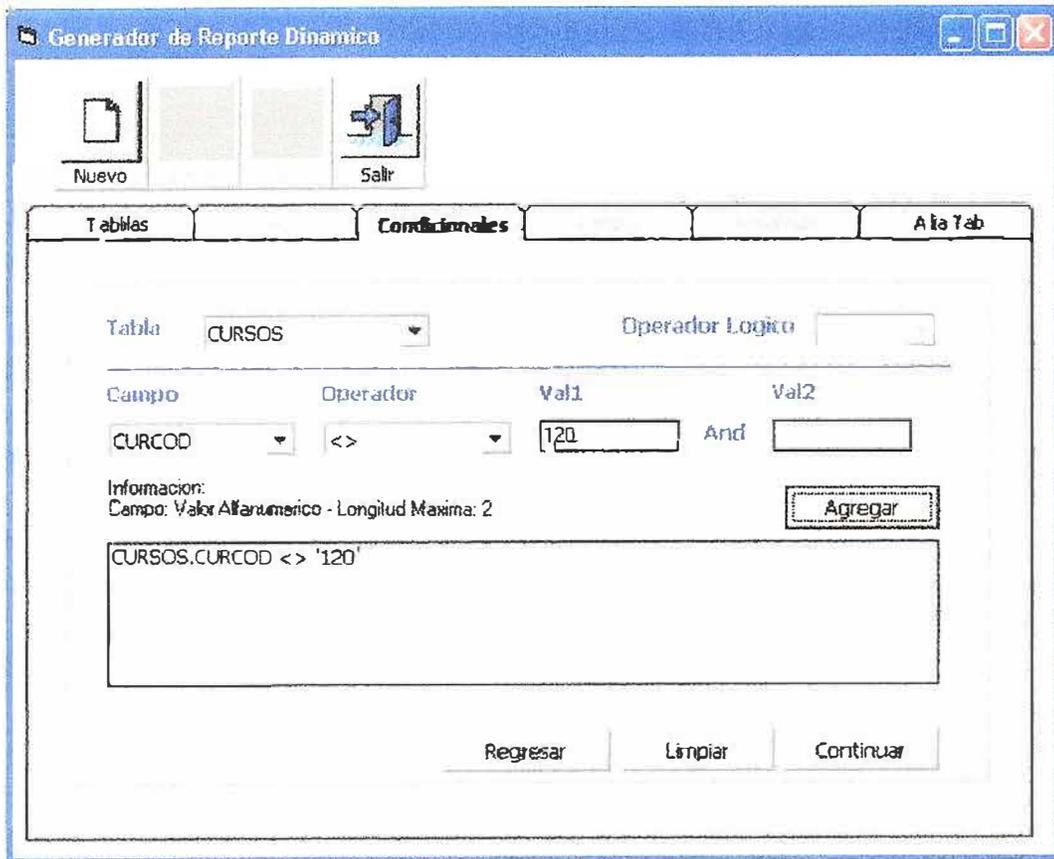
Pantalla generador de reporte dinámico

Esta pantalla que hace parte del menú sistema permite generar reportes a partir de la información referente a cada una de las tablas de la base de datos del sistema. Para poder realizar un reporte de la base de datos, se deben ejecutar los siguientes pasos:

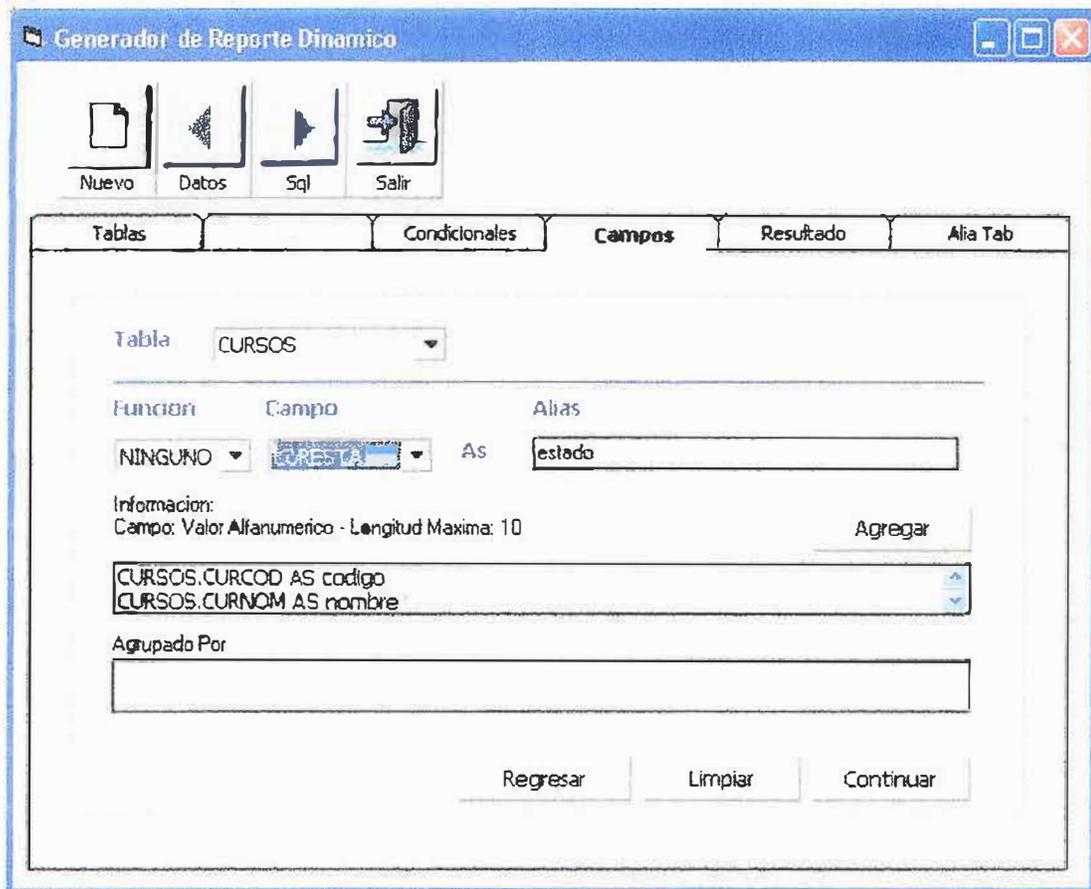
1. En la pestaña **tablas** se debe seleccionar por lo menos una tabla, y luego le damos click en continuar.



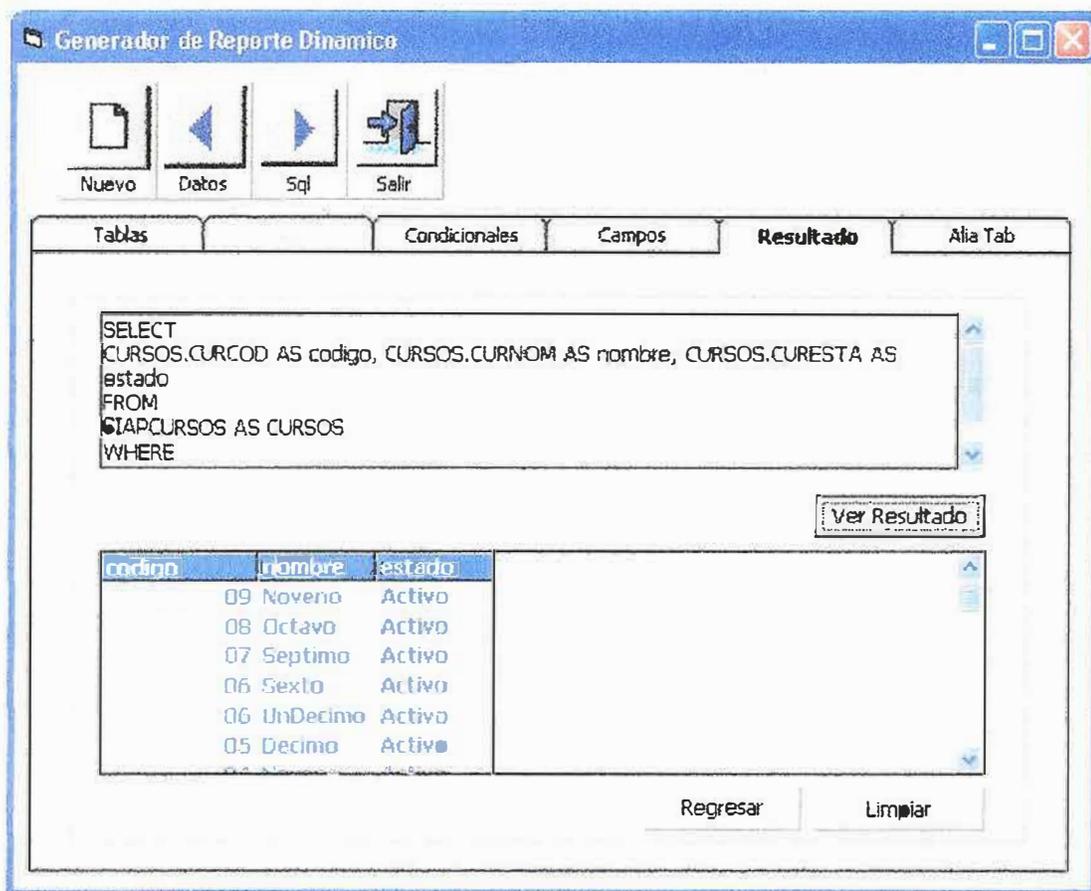
2. En la pestaña **alia tab** se le asigna un alias o nombre a la tabla, y luego se hace click en el boton agregar. Luego se hace click sobre el boton continuar.



3. En la pestaña **condicionales** se realiza el tipo de consulta, seleccionando que campo de la tabla se va a consultar, el operador de condición (like, =, <>) y el valor a buscar. Luego lo agregamos y después se hace click en el botón continuar.



4. en la pestaña **campos** primero se selecciona el tipo de funcion a realizar en dicha consulta, luego se seleccionan uno por uno los campos de la tabla que van a ser generados en la consulta y se le asigna un alias o nombre, después se agregan, y por ultimo se hace click en el botón continuar.



5. En la pestaña **resultado** podemos ver la construcción completa de la consulta, como parte final del generador de reportes. Para poder ver el resultado de la consulta, se presiona el botón ver resultado y luego se muestran las filas consultadas en el cuadro de la parte inferior.

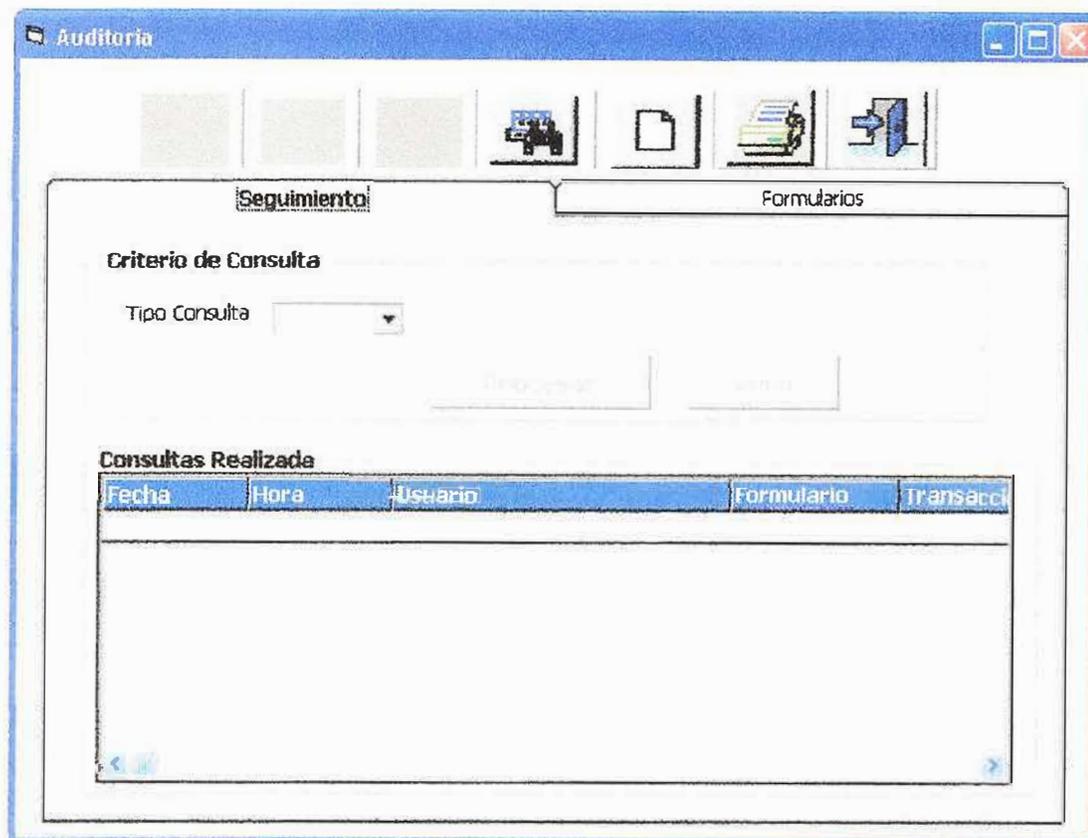
En la parte superior de la pantalla se encuentran cuatro botones con el que podemos trabajar el generador. Estos se describen a continuación:

NUEVO: Permite iniciar una nueva consulta o generar un nuevo reporte.

DATOS: Permite almacenar el resultado de la consulta en un archivo de texto, con extensión **.TXT**.

SQL: Permite guardar la consulta construida en un archivo con extensión **.SQL**

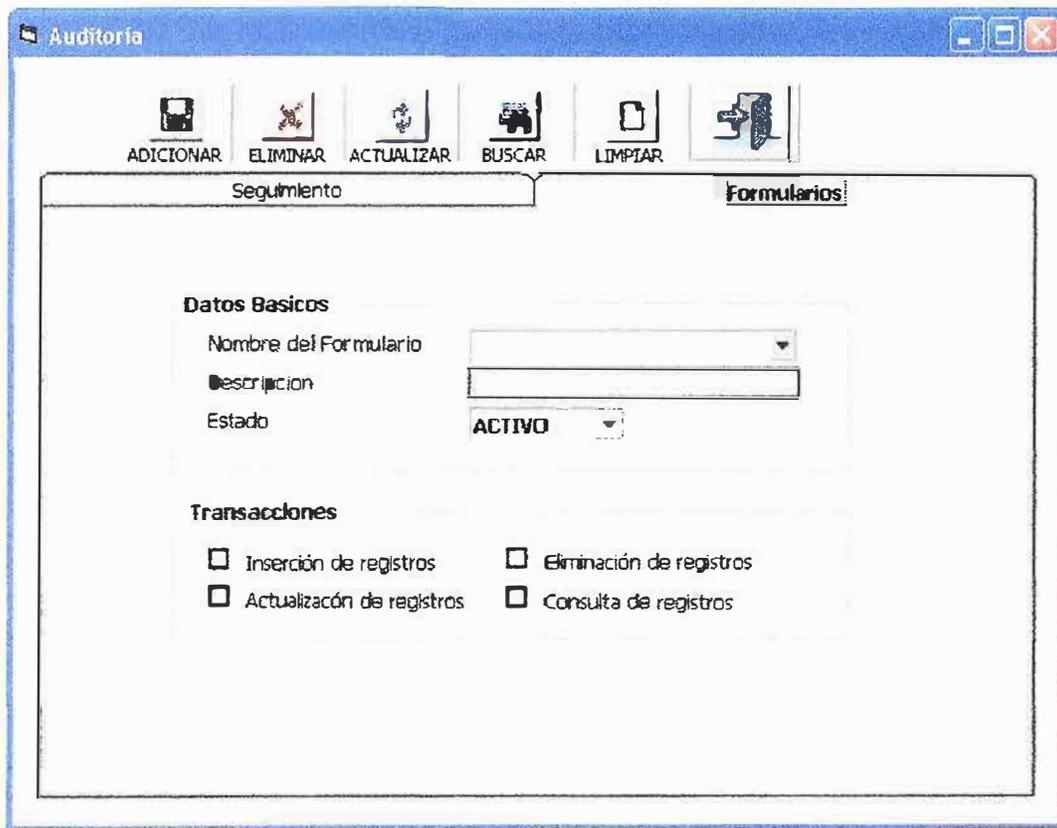
SALIR: Permite cerrar la pantalla de generador de reporte.



Pantalla Auditoria en la pestaña de Seguimiento

Esta pantalla permite registrar las transacciones que se han procesado durante la operación del sistema. En el criterio de consulta se selecciona el tipo de consulta, ya sea por todos los usuarios del sistema, o por un usuario en específico, y por fecha. Si se selecciona por todos los usuarios, se despega un conjunto de filas mostrando todos los usuarios, la fecha y hora, sobre que pantalla hizo la

operación, que tipo de transacción realizó y la descripción de esa transacción. Si la opción escogida es por un usuario, entonces se muestran los usuarios del sistema y se selecciona cual es el usuario que se desea ver. Si la opción escogida es por fecha, entonces aparecerán como parámetros una fecha inicial y una fecha final o actual (DD/MM/AA) y aparecerán las transacciones que se hayan hecho en ese intervalo de tiempo.

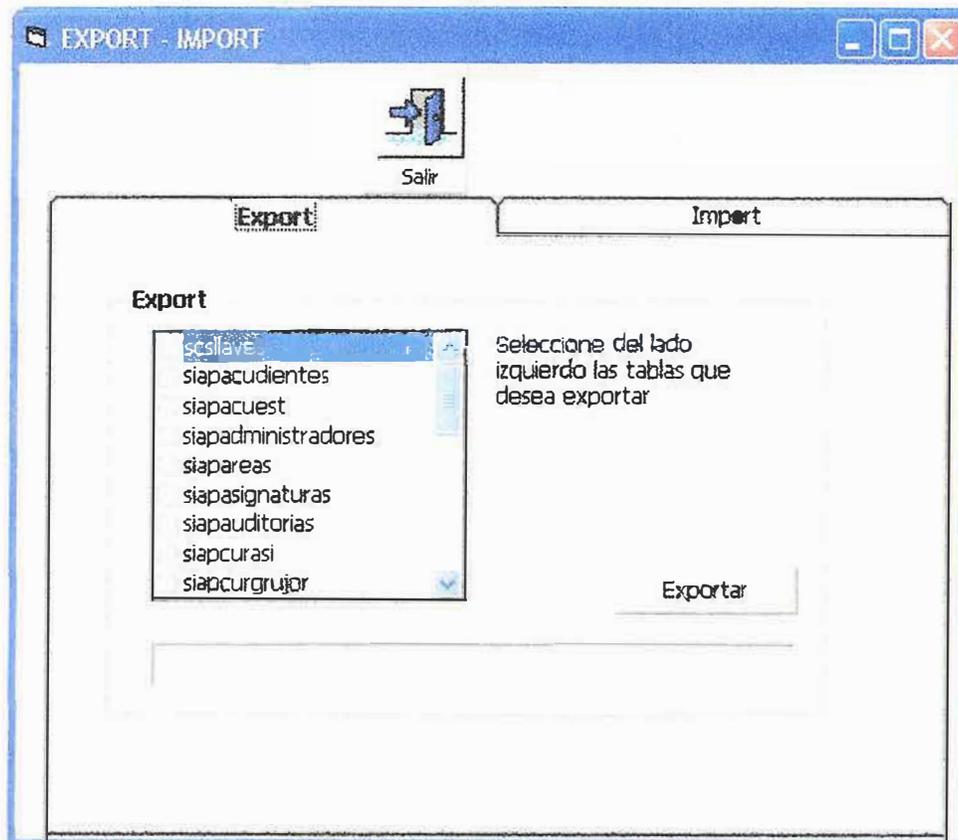


Pantalla Auditoria en la pestaña de formularios

Esta pantalla tiene como función principal llevar un registro de las pantallas sobre las transacciones a evaluar en la auditoria. Como se observa en la imagen, vemos las transacciones que se realizan en cada una de las pantallas, como son la inserción, actualización, consulta y borrado de registros. Es obligatorio que para realizar la auditoria del sistema, el administrador debe de escoger una pantalla a



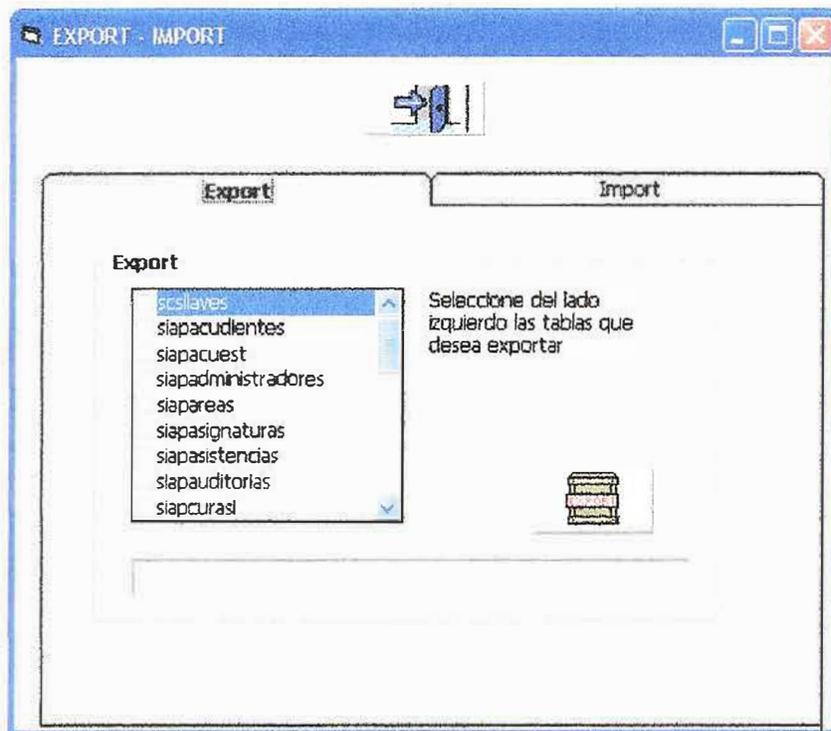
auditar, describir la función del formulario, determinar el estado y luego activar las transacciones, para luego registrarlo.



Pantalla IMPOR/EXPOR en la pestaña EXPOR

Esta pantalla permite realizar la importación de tablas de la base de datos hacia un archivo de texto, con extensión .imex. Para poder

realizar el proceso de importación, se debe de seleccionar al menos una tabla, luego al dar click sobre el boton EXPORT se especifica la ubicación donde el usuario quiere guardar el archivo, y al hacer clic sobre el boton guardar se guarda la información de las tablas seleccionadas en un archivo de texto.



Pantalla IMPOR/EXPOR en la pestaña IMPOR

Esta pantalla permite importar datos desde una ubicación específica hacia la base de datos del sistema.

Primero especificamos la ubicación del archivo donde se encuentran los datos a importar esa ruta se carga en la caja de texto “Archivo De Datos”, luego guardamos un archivo con extensión .imex con un nombre de error, en el cual se van a almacenar los errores que se generan al momento de realizar la importación, en caso de que esta se haya efectuado incorrectamente. Luego se selecciona la tabla en donde se almacenara la información, esto se hace seleccionándola del combo llamado “Tablas” y como paso final de la importación se debe hacer click en el boton “import”.



Pantalla de ayudas en línea

Esta pantalla de ayudas en línea muestra todas las ayudas disponibles del sistema. Estas se encuentran divididas en cada una de las interfaces del sistema, que están en las carpetas ubicadas a la izquierda de la pantalla. Al hacer click sobre cada una de las carpetas se muestra el nombre de la pantalla con una descripción clara, los procesos o transacciones que se ejecutan sobre esta. A su vez, muestra un conjunto de conceptos relacionados con esa pantalla.