



# **UNIVERSIDAD TÉCNICA DE AMBATO**

## **FACULTAD DE INGENIERÍA EN SISTEMAS**

### **CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS**

**TEMA:**

---

**“DESARROLLO E IMPLANTACIÓN DE UN SISTEMA  
AUTOMATIZADO DE CONTROL DE MATRICULAS, PENSIONES Y  
NOTAS PARA LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ”**

---

Proyecto de pasantía de grado, previo a la obtención del título de Ingeniero en  
Sistemas Computacionales e Informáticos.

**AUTOR:**

**Gladys Inés Martínez Barreno**

**TUTOR:**

**Ing. Luís Sailema**

**Ambato – Ecuador**

**Noviembre/2006**

## **APROBACIÓN DEL TUTOR**

En calidad de Tutor del trabajo de investigación sobre el tema:

**DESARROLLO E IMPLANTACIÓN DE UN SISTEMA AUTOMATIZADO DE CONTROL DE MATRICULAS, PENSIONES Y NOTAS PARA LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ**, de Gladys Inés Martínez Barreno, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Universidad Técnica de Ambato, considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometidos a la evaluación de conformidad con el Art. 68 del Capítulo IV Pasantías, del Reglamento de Graduación de Pregrado de la Universidad Técnica de Ambato.

Ambato, Noviembre del 2006

Tutor

-----  
Ing. Luís Sailema

## DEDICATORIA

*A mis amados padres Luís Alonso Martínez Martínez y Silvia de Lourdes Barreno Barreno reciban esta modesta dedicación como un homenaje a su grandeza, que de niña me dieron cuidados y de mujer fortaleza. Por brindarme siempre su apoyo incondicional, por quererme y mimarme ya sea en los momentos buenos o malos que hemos pasado. Tengan presente que la gloria más grande que tengo es el ser hija de ustedes*

*Mis padres son lo mejor del mundo, por eso ellos merecen hoy, mañana y siempre todos mis honores, mi cariño y mi respeto.*

## AGRADECIMIENTO

*A Dios por guiarme por el camino correcto y por ser la luz que siempre a brillado en mis noches mas oscuras.*

*A mis queridos padres Luís A. Martínez M. y Silvia L. Barreno B. que ha sido siempre personas admirables, que me han brindado cuidados, amor y comprensión, quienes con sus sabios consejos me orientaron mis pasos por el camino recto de la vida, quienes siempre me han apoyado moralmente y económicamente, quienes siempre me han apoyado para que estudiara esta carrera.*

*A mí adorado esposo Roberto S. Pérez P. por su gran amor, comprensión y apoyo incondicional que siempre me ha brindado.*

*A mis hermanas Roció, Silvia, Anita y Mariela quienes con sus apoyos morales me ayudaron a no abandonar la batalla contra la adversidad en los días en que se me acumularon las responsabilidades de estudio y atención a la familia.*

*A mi tutor Ing. Luís Sailema por compartir conmigo sus conocimientos y por su guía en este proyecto*

*A mi tío Juan N. Martínez M. quien con sus consejos me ha motivado a seguir siempre adelante y no desmayar en el camino antes alcanzar mis metas.*

*A mis amigas y amigos que siempre han estado junto a mí en todo momento.*

*A todos les agradezco de corazón, muchas gracias.*

## ÍNDICE

Portada .....	i
Página de aprobación del tutor o director.....	ii
Dedicatoria.....	iii
Agradecimiento.....	iv
Índice.....	v
Resumen ejecutivo.....	ix
Introducción.....	x
CAPITULO I.....	1
EL PROBLEMA DE INVESTIGACIÓN	
1.1 Tema de Investigación.....	1
1.2 Planteamiento del problema.....	1
1.3 Justificación.....	2
1.4 Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos.....	3
CAPITULO II.....	4
MARCO TEORICO	
2.1 Antecedentes Investigativos.....	4
2.3 Fundamentación legal.....	4
2.4 Categorías fundamentales.....	5
2.4.1. Informática.....	5
2.4.2. Sistemas de Información.....	6
2.4.2.1 Tipos y Usos de los Sistemas de Información.....	7
2.4.3. Sistemas Cliente Servidor.....	8
2.4.3.1. Cliente.....	8
2.4.3.2. Servidor.....	8
2.4.3.3. Características del modelo cliente/servidor.....	8
2.4.3.4. Tipos de clientes.....	10

2.4.3.5. Tipos de servidor.....	10
2.4.4. Base de Datos.....	11
2.4.4.1. Beneficios de usar Base de Datos.....	11
2.4.4.2. Normalización .....	12
2.4.4.2.1. Primera forma normal (1FN) .....	14
2.4.4.2.2. Segunda forma normal (2FN) .....	14
2.4.4.2.3. Tercera forma normal (3FN) .....	15
2.4.4.2.4. Forma normal de Boyce-Codd (BCFN) .....	15
2.4.4.2.5. Cuarta Forma Normal (4FN).....	16
2.4.4.2.6. Quinta forma normal (5FN).....	16
2.4.5. SQL Server.....	17
2.4.5.1. Servicios de SQL Server.....	18
2.4.5.1. 1. Servicio MSSQLServer.....	18
2.4.5.1. 2. Servicio SQLServerAgent.....	18
2.4.5.1.3. Servicio Microsoft Distributed Ransaction Coordinator.....	19
2.4.5.1.4. Servicio Microsoft Search.....	19
2.4.5.2. Software de SQL Server.....	19
2.4.5.2.1. Administración DE SQL Server.....	19
2.4.5.2.2. Seguridad en SQL Server.....	20
2.4.5.3. Herramientas y asistentes para administración de SQL Server.....	20
2.4.6. Visual Basic.....	21
2.4.6.1. Características generales.....	22
2.4.6.2. Ventajas e inconvenientes.....	23
2.4.6.3. Tipos de variables.....	25
2.4.7. Educación.....	27
2.4.7.1. Características generales de la educación.....	27
2.4.7.2. Objetivos y principios en los que se basa la educación.....	28
2.4.7.3. Educación particular.....	28
2.4.7.4. Control Estudiantil.....	29
2.4.7.5. Notas o Calificaciones Escolares.....	29
2.5 Hipótesis.....	29
2.6 Señalamiento de variables de la hipótesis.....	29

CAPITULO III.....	30
METODOLOGIA	
3.1 Enfoque.....	30
3.2 Modalidad básica de la investigación.....	30
3.3 Nivel o tipo de investigación.....	30
3.4 Población y muestra.....	31
3.5 Recolección de información.....	31
3.6 Procesamiento y análisis.....	31
CAPITULO IV.....	32
ANÁLISIS DEL SISTEMA	
4.1 Análisis del Sistema Actual.....	32
4.2 Análisis del Sistema Propuesto.....	32
4.3 Estudio de Factibilidades.....	33
4.3.1 Factibilidad Económica.....	33
4.3.2 Factibilidad Técnica.....	36
4.3.3 Factibilidad Operacional.....	37
4.4 Análisis Estructurado.....	38
4.4.1 Diagrama de contexto.....	38
4.4.2 Diagrama de flujo de datos.....	39
4.4.2.1 Diagrama de nivel 1.....	39
4.4.2.2 Diagrama de nivel 2.....	40
4.4.3 Diccionario de datos.....	42
4.4.3.1 Lista de procesos en forma de árbol.....	42
4.4.3.2 Procesos.....	42
CAPITULO V.....	51
DISEÑO	
5.1 Diseño lógico.....	51
5.2 Diseño físico.....	52.
5.3 Diccionario de datos del diseño físico.....	53

5.3.1 Estructura y descripción de las tablas.....	53
5.4 Diseño de entradas.....	58
5.5 Diseño de Salidas.....	60
5.6 Diagrama de Procesos.....	62
CAPITULO VI.....	65
IMPLEMENTACIÓN Y PRUEBAS	
6.1 Implementación.....	65
6.2 Pruebas.....	86
CAPITULO VII.....	87
IMPLANTACIÓN	
7.1 Puesta en Marcha.....	87
7.2 Capacitación.....	88
CAPITULO VIII.....	89
CONCLUSIONES Y RECOMENDACIONES	
1.1 Conclusiones.....	89
1.2 Recomendaciones.....	90
Bibliografía.....	91
Glosario.....	92
Anexos.....	93

## **RESUMEN EJECUTIVO**

La Unidad Educativa “González Suárez”, que actualmente cuenta con 96 años al servicio de la Educación, dispone de un edificio propio ubicado en las calles Pichincha y Quiz Quiz, que alberga alrededor de 1200 estudiantes.

En la actualidad la institución no posee reglamentos, estatutos que ayuden a reforzar los diferentes servicios que presta a sus usuarios.

Las actividades que se realizan para la matriculación, pago de pensiones y asignación de notas, se laboran manualmente causando el difícil acceso a la información lo cual ocasiona molestias a los estudiantes, profesores y alumnos.

Es necesario que la Unidad Educativa “González Suárez” implante un sistema automatizado del control estudiantil para dar mayor atención al padre de familia, estudiantes, empleados y profesores, para brindar un mejor servicio y optimizarlos procesos valiéndose de las últimas tecnologías relacionados a los procesos administrativos.

## INTRODUCCION

El presente proyecto de pasantía de graduación contiene los siguientes capítulos:

### CAPITULO I

**“El Problema de Investigación”** contiene la descripción de la situación actual que determina al problema, sus posibles causas y consecuencias, la formulación del problema delimitando contenido, espacio y tiempo, el planteamiento de preguntas, justificación por los cuales merece ser estudiado y los objetivos que se pretenden adquirir.

### CAPITULO II

**“Marco Teórico”** contiene las teorías conceptuales que fundamentan la investigación, como investigaciones previas que servirán de soporte a la nueva investigación, estudio de la institución, definición de términos implicados en la investigación y determinación de variables.

### CAPITULO III

**“Metodología”** contiene la manera en la que se va a realizar la investigación es el ¿Como? Y ¿Con que? Se va a investigar, estableciendo el enfoque, modalidad y nivel de la investigación, instrumentos y procesamientos de la información.

### CAPITULO IV

**“Análisis del Sistema”** contiene el estudio del sistema existente, problemas y carencias detectadas, también contiene la alternativa de solución desarrollada, es decir, el estudio del nuevo sistema.

## **CAPITULO V**

“**Diseño**” contiene rediseño del esquema, y desarrollo del modelo de datos, los cuales permiten que el proyecto desarrollado esté apropiado con las necesidades de la institución.

## **CAPITULO VI**

“**Implementación y Pruebas**” contiene el código más elemental del sistema desarrollado, las pruebas de integración y las pruebas ejecutadas al sistema.

## **CAPITULO VII**

“**Implantación**” contiene el funcionamiento del sistema desarrollado, la instalación de sus herramientas y la capacitación para los usuarios que van utilizar el nuevo sistema.

## **CAPITULO VIII**

“**Conclusiones y Recomendaciones**” contiene sugerencias para garantizar el correcto control de la información de los estudiantes y mejorar el manejo del sistema estudiantil.

## CAPITULO I

### EL PROBLEMA DE INVESTIGACIÓN

#### 1.5 Tema de Investigación

Desarrollo e implantación de un sistema automatizado de control de matriculas, pensiones y notas para la unidad educativa “González Suárez”.

#### 1.6 Planteamiento del problema

##### Contextualización

El Sistema de Matrículas es el que permite, en primera instancia, a una alumna y a sus padres, vincularse formalmente como integrantes de la comunidad educativa. Esta vinculación se establece mediante un contrato de servicios educativos, en el cual se enuncian los derechos, deberes y obligaciones económicas de los contratantes.

En la actualidad la institución no cuenta con ninguna clase de sistema por lo que para el control de matriculas lo hacen manualmente mientras que con los pagos de pensiones y notas lo registran en un computador con programas diferentes para esas transacciones. Con la implementación del sistema automatizado se ahorra tiempo y recursos, calidad en el servicio y ordenamiento de la información.

##### Análisis Crítico

El desconocimiento de nuevas tecnologías, la falta de capacitación y la falta de recursos económicos, han causado el difícil acceso a los datos, falta de organización en la información y en otras actividades; trayendo como consecuencias que la atención sea deficiente, lo cual conlleva la pérdida de tiempo, pérdida económica, pérdida de recursos y perdida de información.

##### Prognosis

Si se continúa con los diferentes problemas mencionados, a futuro se puede generar desconfianza en el servicio y pérdida de alumnos.

Para solucionar estos problemas planteamos la “Implantación de un sistema de control de matriculas, pensiones y notas”.

### **Formularon del problema**

¿Un sistema de control automático de matriculas, pensiones y notas puede facilitar la administración de la información académica que sea más accesible y legible en la Unidad Educativa “González Suárez”?

### **Preguntas directrices**

¿Cuál sería el sistema de almacenamiento que ofrezca facilidad al usuario?

¿Qué servicios son los más requeridos por los usuarios de la Institución?

¿Qué elementos intervienen en el sistema para la búsqueda de la información?

¿Es necesaria la utilización de una interfaz Grafica para la presentación de los datos?

### **Delimitación**

El presente proyecto se realizara en la ciudad de Ambato en la Unidad educativa “González Suárez”, abarcando el periodo de Mayo a Septiembre del 2006.

## **1.7 Justificación**

Como hemos venido observando a través de los años el mundo ha dado grandes cambios en lo que respecta al campo tecnológico y cultural.

Diariamente apreciamos como la labor manual es remplazada por las computadoras, haciendo que todo el trabajo que realizaban varias personas hoy solo es aplicado por medio de una sola y la ayuda de la informática. Muchas instituciones se mantienen siempre con la tecnología actualizada, siendo imprescindible el tratamiento automatizado de la información.

La Unidad Educativa “González Suárez” no posee en la actualidad de reglamentos, estatutos que ayuden a reforzar los diferentes servicios que

presta a sus usuarios. Todas sus actividades laboran manualmente causando el difícil acceso a la información lo cual ocasiona molestias a los estudiantes, profesores y alumnos.

Las instituciones educativas son uno de los ejes principales de la enseñanza, por ello es necesario que la Unidad Educativa “González Suárez” implante un sistema automatizado del control estudiantil para proporcionar mayor atención al usuario, mejoramiento y optimización de los procesos valiéndose de las últimas tecnologías relacionados a los procesos administrativos.

## **1.8 Objetivos**

### **Objetivo General**

Desarrollar e implantar un sistema que permita automatizar los procesos de la información relacionados al control estudiantil de la Unidad Educativa “González Suárez”.

### **Objetivos Específicos**

- ❖ Mejorar la calidad de atención al usuario, mediante búsquedas rápidas de los datos.
- ❖ Facilitar el trabajo al personal administrativo por medio de la clasificación de la información.
- ❖ Determinar los elementos fundamentales que intervienen en un sistema de control matriculación.
- ❖ Crear una base de datos que almacene gran cantidad de información de manera sencilla y que el acceso a los datos sea eficiente.
- ❖ Diseñar e implantar un sistema cliente/servidor que permita registrar todas las transacciones que se realicen con los estudiantes y docentes.

## **CAPITULO II**

### **MARCO TEORICO**

#### **2.1 Antecedentes Investigativos**

Revisado la información existente en referencia a los sistemas de matriculación y control de pensiones y notas se ha encontrado un trabajo investigativo realizado por el Ing. Naranjo Calderón Jorge Danilo de la Universidad Técnica de Ambato FIS 2002, quien en sus conclusiones establece que es necesario un software para automatizar las tareas fundamentales que se realizan en las instituciones que permita el fácil acceso a los datos y un control adecuado de los mismos.

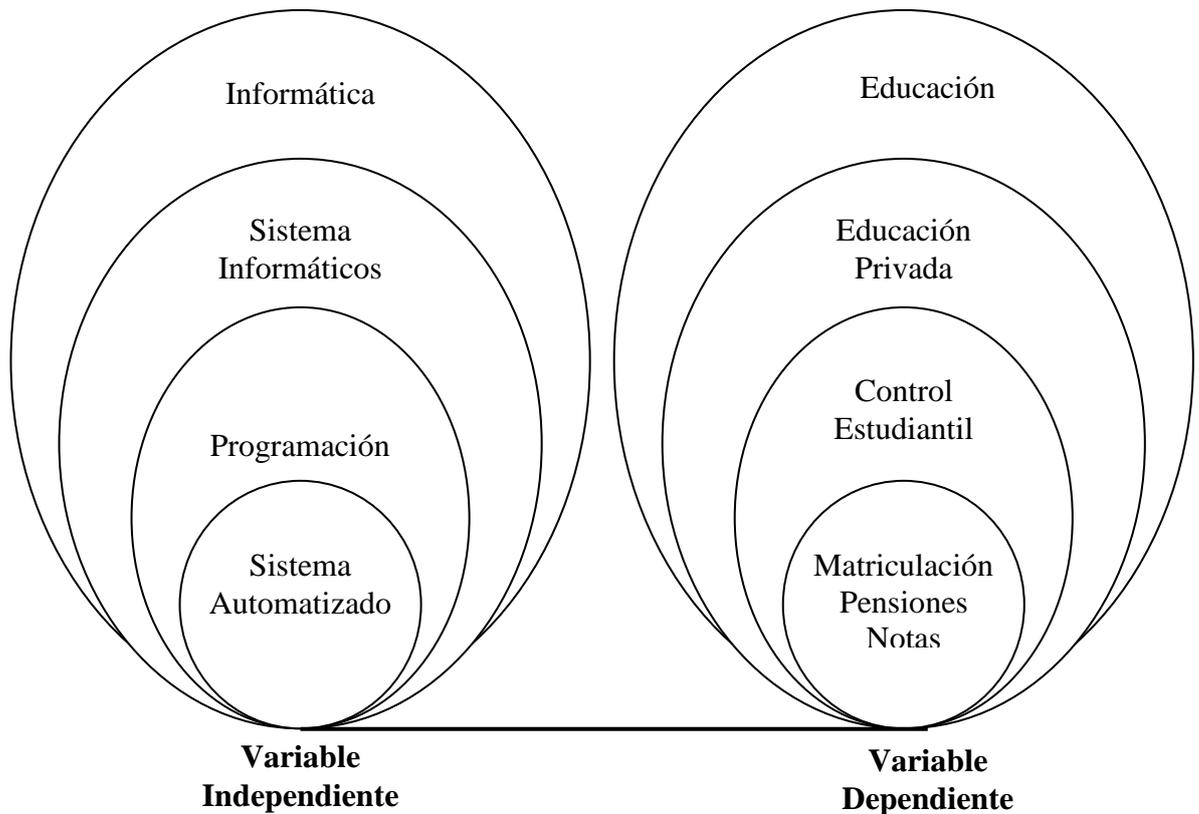
Esta conclusión será tomada en cuenta en el trabajo investigativo a realizarse.

#### **2.3 Fundamentación legal**

La Escuela González Suárez se creó el año 1910, inicialmente funcionó en la Av. Simón Bolívar. Lleva el nombre del ilustre historiador ecuatoriano Monseñor FEDERICO GONZALEZ SUAREZ, como un homenaje de reconocimiento a su noble y sacrificada labor educativa a favor de la juventud ecuatoriana.

En el año 1923 pasa a la protección y dirección de los Padres Josefinos de Bellavista, quienes se radicaron en Ambato para contribuir en la formación académica y cristiana de la niñez y juventud tungurahuese. En el año 1996 se unifica la escuela González Suárez con el colegio Leonardo Murialdo, y, con acuerdo ministerial N° 3501 del 12 de julio del mismo año, se crea la UNIDAD EDUCATIVA GONZALEZ SUAREZ.

## 2.4 Categorías fundamentales



### 2.4.1. Informática

La **Informática** es la ciencia del tratamiento automático de la información a través de un computador (llamado también ordenador o computadora).

Entre las tareas más populares que ha facilitado esta tecnología se encuentran: elaborar documentos, enviar y recibir correo electrónico, dibujar, crear efectos visuales y sonoros, maquetar folletos y libros, manejar la información contable en una empresa, reproducir música, controlar procesos industriales y jugar.

**Informática** es un vocablo inspirado en el francés *informatique*, formado a su vez por la conjunción de las palabras **information** y **automatique**, para dar idea de la automatización de la información que se logra con los sistemas computacionales. La informática es un amplio campo que incluye los

fundamentos teóricos, el diseño, la programación y el uso de las computadoras (ordenadores).

#### 2.4.2. Sistemas de Información

Un **Sistema de Información** es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

El equipo computacional: el hardware necesario para que el sistema de información pueda operar. El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

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

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

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

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

**Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

**Salida de Información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes.

#### **2.4.2.1 Tipos y Usos de los Sistemas de Información**

Durante los próximos años, los Sistemas de Información cumplirán tres objetivos básicos dentro de las organizaciones:

- Automatización de procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.
- Lograr ventajas competitivas a través de su implantación y uso.

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas

Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información.

### **2.4.3. Sistemas Cliente Servidor**

**2.4.3.1. CLIENTE.-** Es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN o WAN. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente.

**2.4.3.2. SERVIDOR.-** Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc.

#### **2.4.3.3. CARACTERISTICAS DEL MODELO CLIENTE/SERVIDOR**

En el modelo CLIENTE/SERVIDOR podemos encontrar las siguientes características:

2. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
3. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.
4. Un servidor da servicio a múltiples clientes en forma concurrente.

5. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
6. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
7. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.

8. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
9. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo.

#### **2.4.3.4. TIPOS DE CLIENTES**

##### **"cliente flaco":**

Servidor rápidamente saturado.

Gran circulación de datos de interfase en la red.

##### **"cliente gordo":**

Casi todo el trabajo en el cliente.

No hay centralización de la gestión de la BD.

Gran circulación de datos inútiles en la red.

#### **2.4.3.5. TIPOS DE SERVIDOR**

##### **Servidores de archivos**

Servidor donde se almacena archivos y aplicaciones de productividad como por ejemplo procesadores de texto, hojas de cálculo, etc.

##### **Servidores de bases de datos**

Servidor donde se almacenan las bases de datos, tablas, índices. Es uno de los servidores que más carga tiene.

##### **Servidores de transacciones**

Servidor que cumple o procesa todas las transacciones. Valida primero y recién genera un pedido al servidor de bases de datos.

##### **Servidores de Groupware**

Servidor utilizado para el seguimiento de operaciones dentro de la red.

##### **Servidores de objetos**

Contienen objetos que deben estar fuera del servidor de base de datos. Estos objetos pueden ser videos, imágenes, objetos multimedia en general.

## **Servidores Web**

Se usan como una forma inteligente para comunicación entre empresas a través de Internet. Este servidor permite transacciones con el acondicionamiento de un browser específico.

El cliente envía mensajes que representados en solicitudes SQL hacia el servidor de bases de datos. Los resultados de cada orden de SQL son devueltos al cliente. El DBMS se encarga de recolectar los datos desde su base de datos, no envía los registros completos, teniéndose un uso mucho más eficiente de la capacidad de procesamiento distribuida. Es usual que se generen aplicaciones en el cliente y en el servidor. Los servidores de bases de datos constituyen el fundamento de los sistemas de apoyo de decisiones que precisan de consultas específicas y reportes flexibles.

### **2.4.4. Base de Datos**

**Base de Datos** es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Una Base de Datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de sus utilización y su implementación en maquinas accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

#### **2.4.4.1. Beneficios de usar Base de Datos.**

Minimiza la redundancia de datos. Existe redundancia, pero controlada a fin de poder relacionar los datos.

- Aumento de la consistencia, debido a que la redundancia es mínima.
- Integración de los datos. Se tiene una visión más centralizada de los datos.

- Facilidad para compartir datos. Es de fácil acceso porque se comparten.
- Mayor estandarización. Las aplicaciones accesan la Base de Datos de la misma forma. (Administración centralizada).
- Facilidad para el desarrollo de las aplicaciones. Esta es consecuencia del punto anterior y con esto, aumenta la productividad, porque me preocupó solo de los procesos.
- Flexibilidad en el acceso de los datos. Existe más de una forma de acceder a los datos.
- Reducción de la necesidad de mantención de las aplicaciones.
- Independencia de los datos. Es la capacidad de modificar una definición, respecto de un conjunto de datos en el nivel de abstracción, sin afectar la definición equivalente en el nivel superior siguiente, se llama independencia de datos. (Niveles visión, conceptualización y físico.)
- Independencia física de los datos. Es la capacidad de modificar una definición de datos en el nivel físico sin provocar que se deban escribir los programas de aplicación nuevamente. Cuando hay problemas de eficiencia, se produce el cambio en el nivel físico.
- Independencia lógica de los datos. Es la capacidad de modificar una definición de datos en el nivel conceptual, sin provocar que se vuelvan a escribir los programas de aplicación. La independencia lógica es en general, difícil de lograr porque esta asociada a cambios en los requerimientos de información, los cuales usualmente son complejos.

#### **2.4.4.2. Normalización**

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad. Aquí no se utilizará la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la

correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias entre atributos no deseadas. Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

### **Relación entre las formas normales:**

Uno de los conceptos fundamentales en la normalización es el de *dependencia funcional*. Una **dependencia funcional** es una relación entre atributos de una misma relación (tabla).

La **dependencia funcional** es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son

opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

#### **2.4.4.2.1. Primera forma normal (1FN)**

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales. La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

Un conjunto de relaciones se encuentra en 1FN si ninguna de ellas tiene grupos repetitivos.

#### **2.4.4.2.2. Segunda forma normal (2FN)**

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo que no está en la clave primaria es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las

relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

#### 2.4.4.2.3. Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. La dependencia  $x \longrightarrow z$  es transitiva si existen las dependencias  $x \longrightarrow y$ ,  $y \longrightarrow z$ , siendo  $x$ ,  $y$ , atributos o conjuntos de atributos de una misma relación.

Aunque las relaciones en 2FN tienen menos redundancias que las relaciones en 1FN, todavía pueden sufrir anomalías frente a las actualizaciones. Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).

#### 2.4.4.2.4. Forma normal de Boyce-Codd (BCFN)

Una relación está en la forma normal de Boyce-Codd si, y sólo si, todo determinante es una clave candidata.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si éstas existen.

La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en BCFN está en 3FN. La violación de la BCFN es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la BCFN si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común.

#### **2.4.4.2.5. Cuarta Forma Normal (4FN )**

Una relación R esta en 4FN:

- Si R esta en la BCFN
- No contiene dependencias multivaluadas.

Existe una **dependencia multivaluada** cuando hay tres atributos (A, B y C) en una relación, tal que por cada valor de A existe un bien definido conjunto de valores de B y un bien definido conjunto de valores de C, sin embargo el conjunto de valores de B es independiente del conjunto C y viceversa.

- La relación esta en la BCFN.
- La clave primaria de la relación se expresa en función de los tres atributos.
- Por cada computadora existe un conjunto de paquetes y un conjunto de tiendas que las venden.

#### **2.4.4.2.6. Quinta forma normal (5FN)**

Existe otro nivel de normalización que se aplica a veces, pero es de hecho algo oculto y en la mayoría de los casos no es necesario para obtener la mejor funcionalidad de la estructura de datos o aplicación. Su principio sugiere:

1. La tabla original debe ser reconstruida desde las tablas resultantes en las cuales a sido troceada.

Los beneficios de aplicar esta regla aseguran que no ha creado ninguna columna extraña en las tablas y que la estructura de las tablas que se ha creado sea del tamaño justo que tiene que ser. Es una buena práctica aplicar esta regla, pero a no ser que se esté tratando con una extensa estructura de datos probablemente no se la necesitará

#### **2.4.5. SQL SERVER**

SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS) que usa Transact-SQL para mandar peticiones entre un cliente y el SQL

SQL Server usa la arquitectura Cliente / Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo Servidor y tareas que corran en computadoras tipo Cliente:

- ◆ El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server.
- ◆ SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc) entre las múltiples peticiones.

La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

### **2.4.5.1. SERVICIOS DE SQL SERVER**

Los servicios de SQL Server incluyen MSSQLServer, SQLServerAgent, Microsoft Distributed Transaction Coordinator (MSDTC), y Microsoft Search. Aunque estos servicios de SQL generalmente corren en Windows NT, también pueden correr como aplicaciones.

#### **2.4.5.1. 1. SERVICIO MSSQLServer:**

Este servicio es el motor de la Base de Datos. Este es el componente que procesa todas las declaraciones de Transact-SQL y administra todos los archivos que definen a la Base de Datos dentro del Servidor. Sus características son:

- ◆ Asignar los recursos de la computadora a múltiples usuarios simultáneos.
- ◆ Previene problemas lógicos, tales como sincronización de peticiones de usuarios que desean actualizar la misma información al mismo tiempo.
- ◆ Garantiza la integridad y consistencia de datos.

#### **2.4.5.1. 2. SERVICIO SQLServerAgent:**

Este es un servicio que trabaja conjuntamente con SQL Server para crear y administrar tareas locales o externas; letras y operadores.

#### **2.4.5.1.3. SERVICIO MICROSOFT DISTRIBUTED TRANSACTION COORDINATOR:**

MSDTC permite a los clientes incluir muchos tipos de datos en una transacción. Coordina la correcta realización de las transacciones distribuidas para asegurar que todas las actualizaciones en todos los servidores son permanentes; o en caso de errores, que las modificaciones son canceladas.

#### **2.4.5.1.4. SERVICIO MICROSOFT SEARCH:**

Este servicio es un motor de full-text que corre como un servicio de Windows NT. El soporte Full Text involucra la habilidad de emitir queries hacia los datos y la creación y mantenimiento de índices que facilitan dichos queries.

### **2.4.5.2. SOFTWARE DE SQL SERVER**

SQL Server incluye una variedad de software para administrar y mantener al servidor, encontrando ayuda acerca de temas específicos, diseñando y creando Bases de Datos y buscando información.

#### **2.4.5.2.1. ADMINISTRACIÓN DE SQL SERVER:**

SQL Server puede ser administrado usando:

- ◆ Utilidades Batch incluidas en SQL Server, tales como OSQL o BCP.
- ◆ Herramientas de administración gráfica incluidas en SQL Server.
- ◆ Aplicaciones COM-compatibles: tal como Visual Basic.

#### **2.4.5.2.2. SEGURIDAD EN SQL SERVER**

SQL Server valida a los usuarios con 2 niveles de seguridad; autenticación del login y validación de permisos en la Base de Datos de cuentas de usuarios y de roles. La autenticación identifica al usuario que está usando una cuenta y verifica sólo la habilidad de conectarse con SQL Server. El usuario debe tener permiso para acceder a las Bases de Datos en el Servidor. Esto se cumple para asignar permisos específicos para la Base de Datos, para las cuentas de usuario y los roles. Los permisos controlan las actividades que el usuario tiene permitido realizar en la Base de Datos del SQL Server.

#### **2.4.5.3. HERRAMIENTAS Y ASISTENTES PARA ADMINISTRACIÓN DE SQL SERVER:**

SQL Server provee un número de herramientas administrativas y asistentes que atienden aspectos particulares de SQL Server. La siguiente tabla describe las herramientas y asistentes de SQL Server:

<b>HERRAMIENTA GRÁFICA</b>	<b>APLICACIÓN</b>
Configuración Cliente de SQL Server	Utilidad para administrar la configuración cliente para componentes de comunicación
Monitor de Funcionamiento de SQL Server	Archivo usado para integrar SQL Server con El Monitor de Funcionamiento de Windows NT, para informar las estadísticas más recientes de actividad
SQL Server Profiler	Utilidad para capturar el record continuo de la actividad del servidor

<p style="text-align: center;">Analizador de Queries de SQL Server</p>	<p>Herramienta gráfica de Queries usada para analizar el plan de un query, visualizar información estadística, y administrar varios queries en diferentes ventanas al mismo tiempo.</p>
--	---

**Tabla 1.**

#### **2.4.6. Visual Basic**

**Visual Basic** es un lenguaje de programación desarrollado por Alan Cooper. Visual Basic es un lenguaje y compilador visual que desciende del lenguaje de programación BASIC.

**Visual Basic 6.0** es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de PCs, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que desarrollan aplicaciones complejas en poquísimos minutos (comparado con lo que cuesta programar en **Visual C++**, por ejemplo). En el caso de los programadores novatos por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje. El precio que hay que pagar por utilizar **Visual Basic 6.0** es una menor velocidad o eficiencia en las aplicaciones.

**Visual Basic 6.0** es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

**Visual Basic 6.0** es también un programa basado en objetos, aunque no orientado a objetos como **C++** o **Java**. La diferencia está en que **Visual Basic 6.0** utiliza **objetos** con **propiedades** y **métodos**, pero carece de los

mecanismos de **herencia** y **polimorfismo** propios de los verdaderos lenguajes orientados a objetos como **Java** y **C++**.

En este primer capítulo se presentarán las características generales de **Visual Basic 6.0**, junto con algunos ejemplos sencillos que den idea de la potencia del lenguaje y del modo en que se utiliza.

#### **2.4.6.1. Características generales**

Es un lenguaje de fácil aprendizaje pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios poderoso que facilita el rápido desarrollo de aplicaciones gráficas. Su principal innovación, que luego fue adoptada por otros lenguajes, fue el uso de un tipo de dll, llamado inicialmente vbx y posteriormente ocx, que permiten contener toda la funcionalidad de un control y facilitar su rápida incorporación a los formularios.

Su sintaxis, derivada del antiguo BASIC, ha sido ampliada con el tiempo al agregarse las características típicas de los lenguajes estructurados modernos. Se ha agregado una implementación limitada de la Programación Orientada a Objetos (los propios formularios y controles son objetos), que, sin embargo, no admite ni polimorfismos ni herencia. No requiere de manejo de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL ,etc) a través de ADO.

Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje.

El compilador de Microsoft genera ejecutables que requieren una DLL para que sus ejecutables funcionen, en algunos casos llamada MSVBVMxy.DLL (acrónimo de "MicroSoft Visual Basic Virtual Machine x.y", siendo x.y la versión) y en otros VBRUNXXX.DLL ("Visual Basic Runtime X.XX"), que provee todas las funciones implementadas en el lenguaje. Además existen un gran número de bibliotecas (DLL) que facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones.

#### **2.4.6.2. Ventajas e inconvenientes**

##### **Ventajas**

- Visual Basic es un lenguaje simple y por tanto fácil de aprender.
- Su mayor facilidad radica en el dibujado de formularios, mediante el arrastre de controles.
- La sintaxis está cercana al lenguaje humano.
- Es un lenguaje RAD, centrado en conseguir en el menor tiempo posible los resultados deseados, por eso mismo su mayor uso está en las pequeñas aplicaciones, como gestión de bares, empresas, restaurantes...
- Tiene una ligera implementación de POO
- Permite el tratamiento de mensajes de Windows.
- Gran parte del trabajo en el diseño de formularios está realizado, gracias a la gran gama de controles incorporados junto al lenguaje que ahorran costes de tiempo de desarrollo.
- Soporta el uso de componentes COM y ActiveX.
- Permite crear controles personalizados fácilmente del mismo modo que el diseño de formularios.
- Permite generar librerías dinámicas (DLL) ActiveX de forma nativa y Win32 (no ActiveX, sin interfaz COM) mediante una reconfiguración de su enlazador en el proceso de compilación.

## Inconvenientes

- Es propietario de Microsoft, por tanto nadie que no sea del equipo de desarrollo de esta compañía decide la evolución del lenguaje.
- Sólo existe un compilador & IDE, llamado igual que el lenguaje.
- Sólo genera ejecutables para Windows.
- No existe forma alguna de portar el código a otras plataformas fuera de Windows (al contrario que con los lenguajes .NET gracias al Proyecto Mono)
- La sintaxis es bastante inflexible.
- Los ejecutables generados son relativamente lentos.
- NO es adecuado para aplicaciones grandes, multimedia, videojuegos, editores gráficos...
- NO permite características de programación avanzada.
- NO permite programación a bajo nivel ni incrustar secciones de código en ASM.
- Sólo permite el uso de funciones de librerías dinámicas (DLL) stdcall.
- Para que los ejecutables que genera funcionen necesita una DLL llamada MSVBVMxy.DLL: MicroSoft Visual Basic Virtual Machine x.y (versión). Provee todas las funciones y características implementadas en el lenguaje.
- Unas pocas funcionalidades están indocumentadas.
- La ligera implementación de POO no permite sacar el máximo provecho de este modelo de programación.
- NO soporta tratamiento de procesos como parte del lenguaje.
- NO tiene manejo de excepciones.
- NO incluye operadores a nivel de bits.
- NO permite el manejo de memoria dinámica, punteros, etc. como parte del lenguaje.
- NO avisa de ciertos errores o advertencias (se puede configurar el compilador para generar ejecutables sin los controladores de desbordamiento de enteros o las comprobaciones de límites en arrays)

entre otros dejando así mas de la mano del programador la tarea de controlar dichos errores)

- El tratamiento de mensajes de Windows es básico e indirecto.
- La gran gama de controles incorporados son, sin embargo en algunos casos, muy generales, lo que lleva a tener que reprogramar nuevos controles para una necesidad concreta de la aplicación.
- Los controles personalizados no mejoran la potencia de la API de Windows, y en determinados casos acudir a ésta será el único modo de conseguir el control personalizado deseado.
- La forma de programación que plantea en un principio Visual Basic ha ocasionado que muchos programadores de Visual Basic practiquen malas costumbres, entre las más comunes:
  - ◆ Variables globales
  - ◆ Variables sin declarar (aunque se puede prevenir tal acción mediante una cláusula en el código)
  - ◆ Variables de tipo indefinido (Variant) que realizan conversión automática de los tipos de datos genéricos del lenguaje de forma transparente.
  - ◆ Código innecesario
  - ◆ Código ilegible
  - ◆ Código repetido
  - ◆ Uso de goto y etiquetas (aunque no son instrucciones imprescindibles a la programar pudiéndose evitar su uso)
  - ◆ Uso de controles como simples contenedores de datos
  - ◆ Dependencia de los controles a la hora de programar (desarrollo de interfaces gráficas, programación de

### **2.4.6.3. TIPOS DE VARIABLES**

Las variables pueden ser de los siguientes tipos: (El número indicado en segundo lugar indica el número de Bytes que ocupa en memoria.)

Booleana	(2)	Admite los valores 0 y 1, o True (verdadero) y False (falso)
Byte	(1)	Números enteros, en el rango de 0 a 255
Integer	(2)	Números enteros en el rango de -32768 a 32767
Long	(4)	Números enteros en el rango de -2147483648 a 2147483647
Single	(4)	Punto flotante, simple precisión
Doble	(8)	Punto flotante, doble precisión.
Currency	(8)	Entero, con punto decimal fijo (Típico de monedas)
String	(*)	Cadenas alfanuméricas de longitud variable o fija
Date	(8)	Fechas
Objet	(4)	Referencia a objetos
Variant	(**)	Otros tipos de datos

(\*) Una variable tipo String ocupa el mismo número de bytes que caracteres tenga la cadena.

(\*\*) Una variable tipo Variant ocupa 16 bytes si se trata de un número y 22 bytes + longitud de la cadena si se trata de un dato tipo cadena de caracteres.

Existen también variables definidas por el usuario (Ya verá la sentencia Type). En este tipo de variables se pueden introducir muchos datos de distinto tipo. Los bytes necesarios para almacenar esa variable dependerá de los datos que se hayan definido.

Dentro de las variables **Objet** (variables que se refieren a objetos) existe una gran variedad que a estas alturas del curso no debemos complicarnos con ellas. Pero para adelantar algo, veremos que una variable puede ser del tipo Form - Formulario - , tipo Picture, etc. etc.

Cada tipo de variable ocupa unos determinados bytes. Si no se define una variable, VB toma como tipo por defecto para la variable el tipo Variant. Este tipo ocupa mas bytes que, por ejemplo, un integer. Si el tipo de dato que

vamos a introducir en una variable es un integer, y no la hemos declarado como tal, VB asumirá para esa variable que es del tipo Variant, lo que le llevará a gastar mas bytes de memoria (16) que los que necesitaría (2) si la hubiésemos declarado previamente. Si esa variable va a estar en el rango de 0 a 255, y no declaramos previamente que la variable va a se del tipo Byte, o la declaramos como integer, p. e., estamos desperdiciando memoria RAM y posiblemente, retardando la ejecución del programa. Lo mismo podemos decir del resto de las variables, y lo importante que es definir las y definir las bien.

#### **2.4.7. Educación**

La educación es parte de aquellos derechos considerados de segunda generación o llamados económicos, sociales y culturales que se efectivizan a través de acciones concretas o de carácter positivo por parte del Estado, algunos autores lo caracterizan por ser de aquellos que generan una obligación de "deber-hacer"; en esta perspectiva estos derechos están ligados a la disponibilidad de recursos, por eso algunos autores consideraban que no eran exigibles de manera directa, sino tenían un valor *programático*, es decir de estándares a ser alcanzados.

##### **2.4.7.1. Características generales de la educación**

La educación es considerada derecho irrenunciable de las personas, deber inexcusable del Estado, la sociedad y la familia; área prioritaria de la inversión pública, requisito del desarrollo nacional y garantía de la equidad social.

La educación, pública y privada, debe inspirarse en principios éticos, pluralistas, democráticos, humanistas y científicos.

El Estado es responsable de la definición y ejecución de las políticas que permitan alcanzar los objetivos indicados

La educación tiene que ver con derechos elementales de la persona, con lo que se llaman derechos y garantías constitucionales.

#### **2.4.7.2. Objetivos y principios en los que se basa la educación**

De acuerdo a los disposiciones constitucionales la educación en todos los niveles, modalidades y sistemas debe promover el respeto a los derechos humanos, desarrollar un pensamiento crítico, fomentar el civismo; proporcionar destrezas para la eficiencia en el trabajo y la producción; estimular la creatividad y el pleno desarrollo de la personalidad y las especiales habilidades de cada persona y debe impulsar la interculturalidad, la solidaridad y la paz.

Como un objetivo funcional se establece que la educación preparará a los ciudadanos para el trabajo y para producir conocimiento. Para lograr estos objetivos se establece que en todos los niveles del sistema educativo se debe procurar que los estudiantes realicen prácticas extracurriculares dirigidas a estimular el ejercicio y la producción de artesanías, oficios e industrias.

Se garantiza la libertad de enseñanza y cátedra, se prohíbe todo tipo de discriminación y se reitera el reconocimiento del derecho que tienen los padres a escoger para sus hijos una educación acorde con sus principios y creencias. Se prohíbe la propaganda y proselitismo político en los planteles educativos y se promueve la equidad de género y la coeducación

#### **2.4.7.3. Educación particular**

Afortunadamente la educación particular es aquella que en definitiva ayuda de alguna manera al Estado Ecuatoriano, para aquellas personas que disponen de los recursos necesarios, además muchos de estos centros educativos, superan a los de la Educación fiscal, y de esta manera dan nuevas oportunidades a los jóvenes actuales.

#### **2.4.7.4. Control Estudiantil**

El Control estudiantil es todo lo relacionado con los alumnos y los maestros, en cuanto al desempeño de las labores que se realizan en la institución, es de vital importancia para el funcionamiento ordenado y productivo, así como en beneficio del prestigio frente a los educandos.

#### **2.4.7.5. Notas o Calificaciones Escolares**

Es la puntuación obtenida en un examen o en cualquier tipo de prueba que se realiza cada vez que sea necesario evaluar a un alumno.

### **2.5 Hipótesis**

La implantación de un sistema para el control de matriculas, pensiones y notas mejorará la accesibilidad y legibilidad de la información didáctica para todos los usuarios de la Unidad Educativa “González Suárez”.

### **2.6 Señalamiento de variables de la hipótesis**

#### **Variable Independiente**

Sistema automatizado

#### **Variable Dependiente**

Control de matriculas, pensiones y notas de la unidad educativa “González Suárez”

## **CAPITULO III**

### **METODOLOGIA**

#### **3.1 Enfoque**

El presente proyecto fue de tipo cuantitativa puesto que es realista, explicativa, busca las causas y la explicación de los hechos que estudia, los datos fueron tratados aplicando la estadística para la determinación de los resultados.

#### **3.2 Modalidad básica de la investigación**

La realización de la presente investigación esta basada en la investigación de campo la cual analiza el problema partiendo de hechos reales, para la obtención de información y requerimientos que evidencien los objetivos.

La investigación bibliográfica sirvió de mucha ayuda, en todo lo que concierne a la teoría en la que se baso la investigación.

La investigación experimental obtuvo mucha influencia para la etapa de pruebas del tema planteado.

#### **3.3 Nivel o tipo de investigación.**

Para el desarrollo de este proyecto, se tomo contacto directo con la realidad para obtener información de acuerdo a los objetivos planteados inicialmente, se trata de un proyecto especial por desarrollarse un software a un problema específico basado en la investigación de campo. De igual manera se ha utilizado la investigación bibliográfica para obtener la información teórica se

ha recurrido al Internet y a diversos libros que tratan sobre el tema. Para realizar las comprobaciones del sistema que se aplicó, se ha utilizado la investigación experimental.

Para la presente investigación se empezó con el nivel exploratorio el cual permitió averiguar los por menores del proyecto; posteriormente se utilizó el nivel descriptivo para clasificar los elementos y estructuras según el comportamiento de los servicios de la institución; finalmente llegamos al nivel explicativo para comprobar experimentalmente la hipótesis.

### **3.4 Población y muestra.**

Para la población se ha tomado en cuenta los siguientes datos:

	Cantidad
Estudiantes.	1200
Docentes.	50
Personal Administrativo	7

### **3.5 Recolección de información**

Las técnicas que permitan obtener información fueron básicamente encuestas cerradas de selección múltiple y entrevistas al personal involucrado en la matriculación, pago de pensiones y notas.

### **3.6 Procesamiento y análisis**

Una vez obtenido los datos necesarios para el desarrollo del proyecto se realizó tabulaciones o cuadros los cuales permitieron obtener una visión más clara sobre sus necesidades.

## **CAPITULO IV**

### **ANÁLISIS DEL SISTEMA**

#### **4.5 Análisis del Sistema Actual**

En la actualidad la institución no cuenta con ninguna clase de sistema por lo que para el control de matriculas lo hacen manualmente mientras que con los pagos de pensiones y notas lo registran en un computador con programas diferentes para esas tareas.

La falta de sistemas apropiados para matriculas, registro de pensiones y registro de notas, han causado el difícil acceso a los datos del alumno y falta de organización de la información; trayendo como consecuencias que la atención a los estudiantes, padres de familia y profesores sea deficiente, lo cual conlleva la perdida de tiempo, perdidas económicas, perdida de recursos y perdida de datos.

#### **4.6 Análisis del Sistema Propuesto**

La Unidad Educativa González Suárez requiere optimizar los procesos que se lleva a cabo para la matriculación, cobro de pensiones y registro de notas, para mejorar la atención que prestan a los estudiantes, padres de familia y profesores con respuestas rápidas y resultados correctos.

Con la implementación del sistema estudiantil se garantiza la consistencia e integridad de los datos en todo momento, y que siempre se hallan disponibles para el usuario que lo necesite.

## 4.7 Estudio de Factibilidades

### 4.7.1 Factibilidad Económica

Los costos que comprenden en el desarrollo del sistema estudiantil se adquirieron mediante el proceso de estimación de costos los mismos que son detallados a continuación.

Los costos están calculados por años.

#### **SISTEMA ACTUAL**

##### **Gasto por las Personas (Personas)**

	Mensual	Anual
Secretaria	\$200,00	\$2.400,00
Contadora	\$200,00	\$2.400,00
Personas=\$2.400,00+\$2.400,00		
Personas=\$4.800,00		

##### **Gasto de Materiales Directos (MD)**

	Mensual	Anual
Suministros de Oficina	\$45,00	\$540,00
MD = \$540,00		

##### **Gasto de Equipos (Equipos)**

Depreciación (1.67%)

	Mensual	Anual
2 Computadoras (\$700)	\$ 23,38	\$280,56
2 Impresoras (\$130)	\$4,34	\$52,08

Equipos=\$280,56+\$52,08

Equipos=\$332,64

##### **Gastos Varios (Varios)**

Varios =\$20,00

**Gastos Directos (GD)**

GD=Personas + MD + Equipos +Varios

GD=\$4.800,00+\$540,00+\$332,64 +\$20,00

GD=\$5.692,64

**Gastos Indirectos (GI)**

GI=GD\*10%

GI=\$5.692,64\*10%

GI=\$569,264

**Costo del Sistema Actual (Gasto Total)**

GT=GD+GI

GT=\$5.692,64+ \$569,26

GT=\$6.261,904

**Gasto Total** =6.261,904 dólares al año

**SISTEMA NUEVO**

**Gasto por las Personas (Personas)**

	Mensual	Anual
Secretaria	\$200,00	\$2400,00
Contadora	\$200,00	\$2400,00
Personas=\$2400,00+\$2400,00		
Personas=\$4.800,00		

**Gasto de Materiales Directos (MD)**

	Mensual	Anual
Suministros de Oficina	\$15,00	\$180,00
MD = \$180,00		

### **Gasto de Equipos (Equipos)**

Depreciación (1.67%)

	Mensual	Anual
2 Computadoras (\$700)	\$ 23,38	\$280,56
2 Impresoras (\$130)	\$4,34	\$52,08

Equipos=\$280,56+\$52,08

Equipos=\$332,64

### **Gastos Varios (Varios)**

Varios =\$20,00

### **Gastos Directos (GD)**

GD=Personas + MD + Equipos +Varios

GD=\$4.800,00+\$180,00+\$332,64 +\$20,00

GD=\$5.332,64

### **Gastos Indirectos (GI)**

GI=GD\*10%

GI=\$5.332,64\*10%

GI=\$533,264

### **Costo del Sistema Nuevo (Gasto Total)**

GT=GD+GI

GT=\$5.332,64+ \$533,264

GT=\$5.865,904

**Gasto Total** =5.865,904 dólares al año

Por lo tanto es factible de realizarse económicamente; ya que el nuevo sistema estudiantil reduce los costos con una cantidad de 396,00 dólares lo cual permite alcanzar un grande beneficio a la Institución.

#### **4.7.2 Factibilidad Técnica**

##### **Software**

Para el desarrollo del sistema estudiantil se podría utilizar los siguientes lenguajes de programación:

- Delphi 9.0
- Visual Basic 6.0
- Power Builder 9.0
- Visual C++ 6.0
- Visual Java 6.0
- Visual Basic. NET

Mientras que para guardar la base de datos se podría utilizar

- Access
- Oracle
- MySQL
- SQL Server

De las herramientas que hemos mencionado, seleccionamos las siguientes:

- Lenguaje de programación Visual Basic 6.0
- Para guardar la base de datos SQL Server

Fueron seleccionadas estas herramientas ya que esta institución cuenta con un modulo realizado con ellas.

##### **Hardware**

En la institución se cuenta con dos maquinas que cumple con las características necesarias para instalar el sistema estudiantil.

##### **Servidor**

- Capacidad: 80Gb
- Microprocesador Pentium IV de 2.8 Ghz
- Memoria RAM de 512Mb

## **Cliente**

- Capacidad: 40Gb
- Microprocesador Pentium IV de 1.8 Ghz
- Memoria RAM de 256Mb

Por lo tanto es factible de realizarse técnicamente, ya que el sistema podrá ser ejecutado en las maquinas del departamento de Administración sin ningún inconveniente.

### **4.7.3 Factibilidad Operacional**

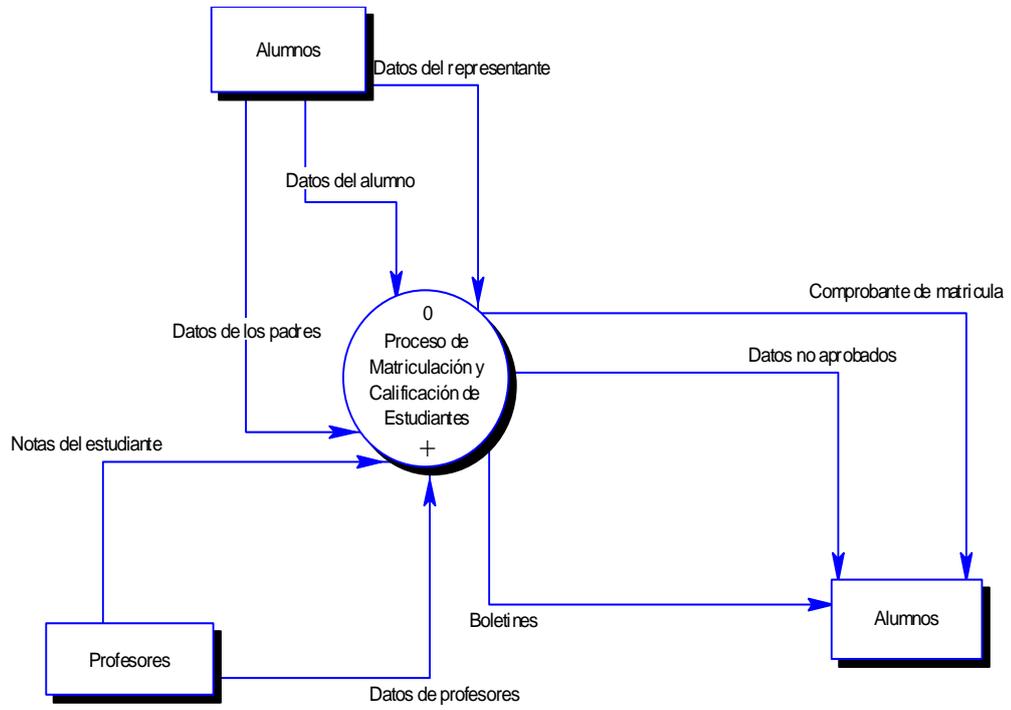
El sistema será desarrollado de manera que controle la mayor parte de errores que el usuario pueda cometer, su interfaz es amigable y fácil de utilizar.

Además se capacitara a todos los usuarios que a utilizaran el sistema, la capacitación consiste principalmente en el manejo del sistema estudiantil antes de su implantación.

Por lo tanto es factible de realizarse operacionalmente, ya que se cuenta con el apoyo de todo el personal de la institución, las mismas que brinda la información veraz cada vez que se le solicita.

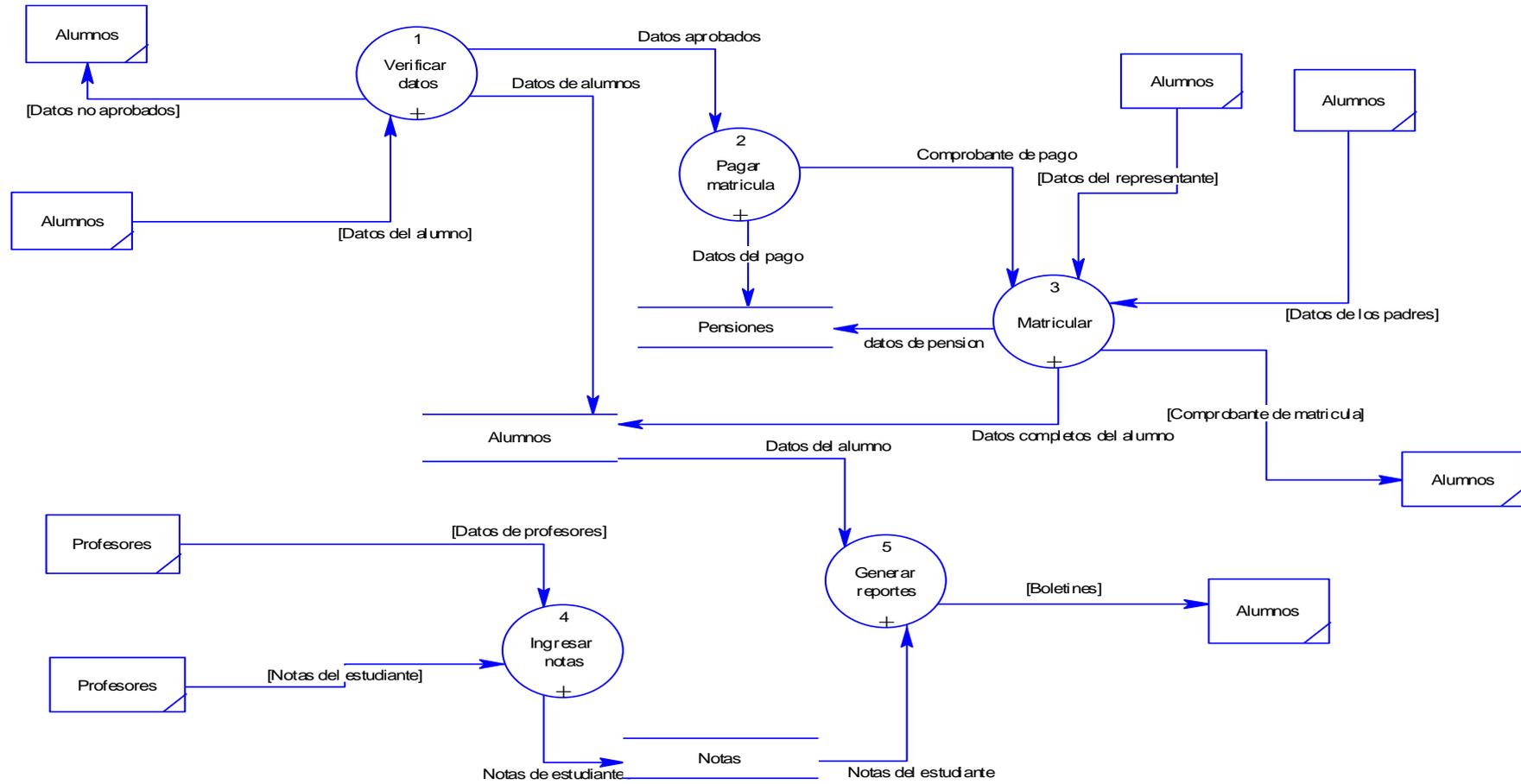
## 4.8 Análisis Estructurado

### 4.8.1 Diagrama de contexto



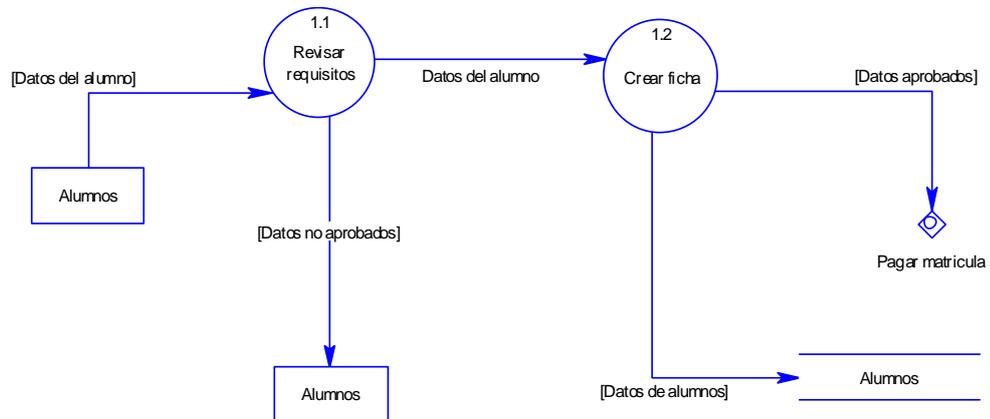
## 4.8.2 Diagrama de flujo de datos

### 4.8.2.1 Diagrama de nivel 1

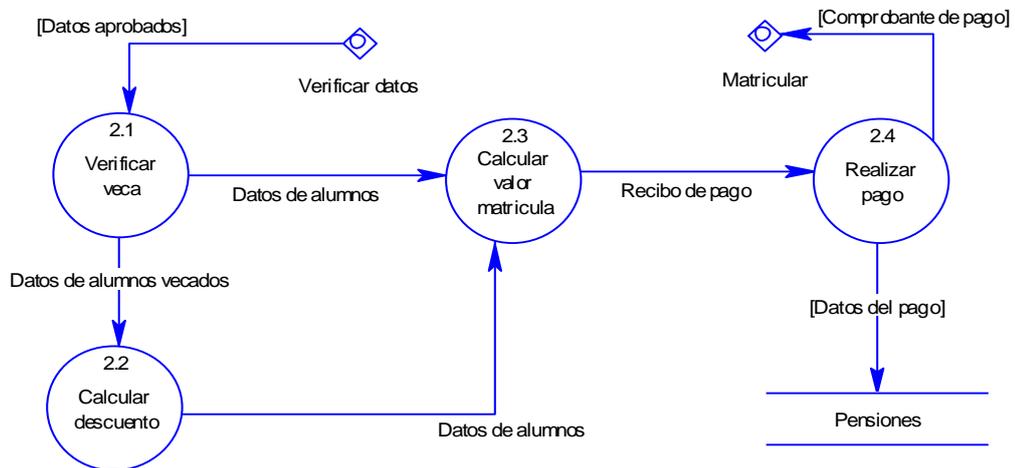


## 4.8.2.2 Diagrama de nivel 2

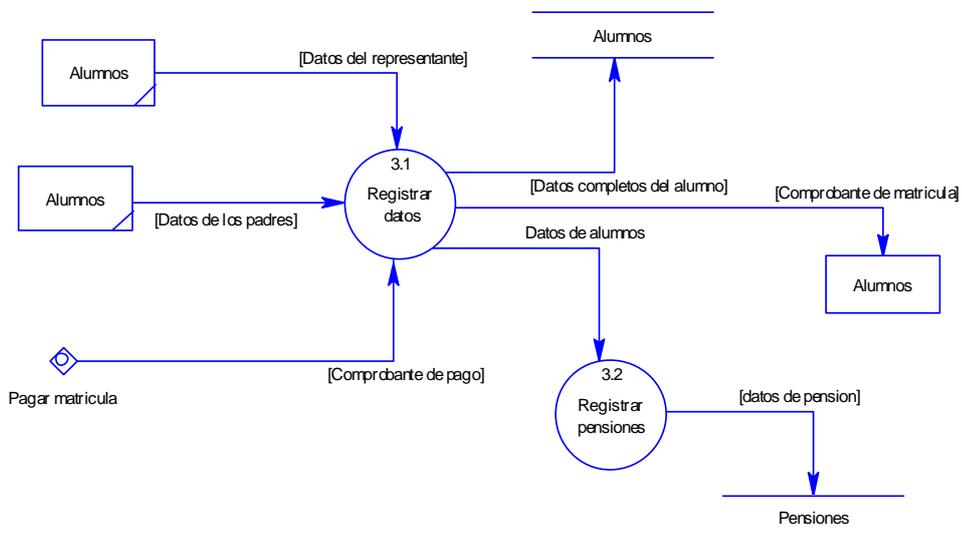
### Verificar datos



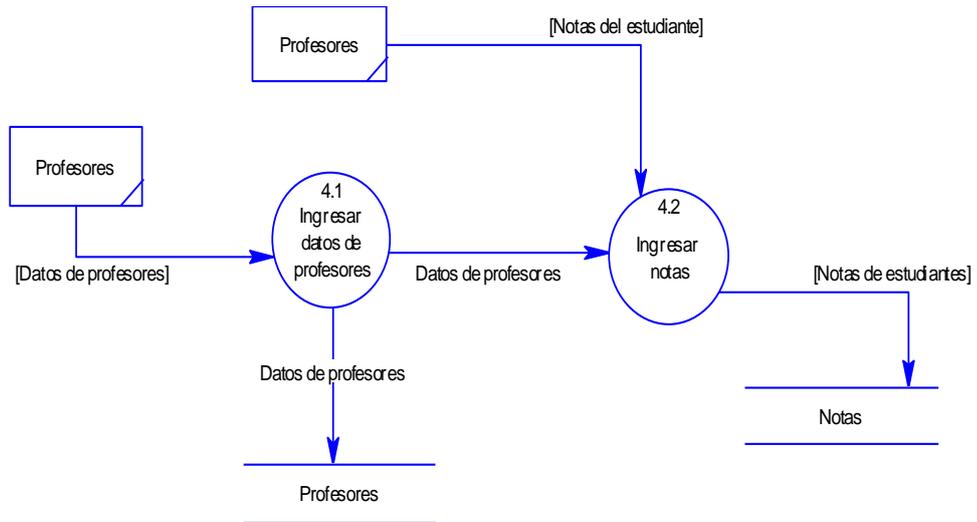
### Pagar Matricula



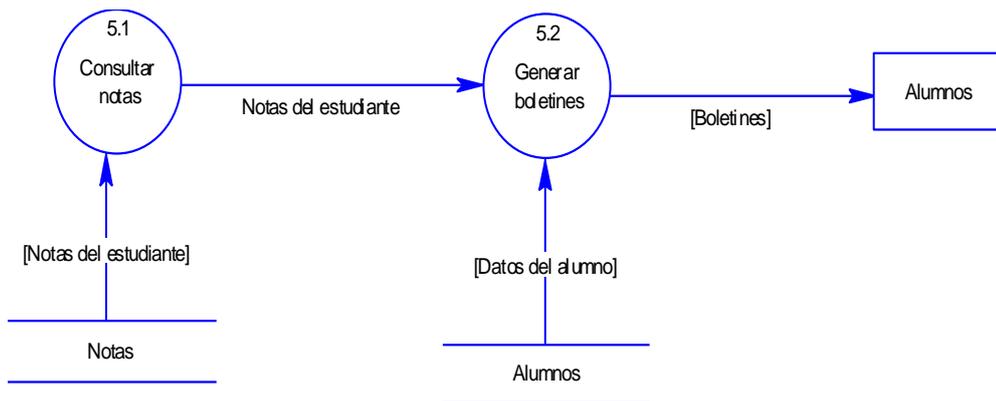
### Matricular



## Ingresar Notas



## Generar Reportes



### 4.8.3 Diccionario de datos

#### 4.8.3.1 Lista de procesos en forma de árbol

Proceso de Matriculación y Calificación de Estudiantes [0]

Verificar datos [1]

Revisar requisitos [1.1]

Crear ficha [1.2]

Pagar matricula [2]

Verificar beca [2.1]

Calcular descuento [2.2]

Calcular valor matricula [2.3]

Realizar pago [2.4]

Matricular [3]

Registrar datos [3.1]

Registrar pensiones [3.2]

Ingresar notas [4]

Ingresar datos de profesores [4.1]

Ingresar notas [4.2]

Generar reportes [5]

Consultar notas [5.1]

Generar boletines [5.2]

#### 4.8.3.2 Procesos

##### Proceso de Matriculación y Calificación de Estudiantes

<b>Nombre:</b>	Proceso de Matriculación y Calificación de Estudiantes
<b>Código:</b>	PRCS_1
<b>Número:</b>	0
<b>Descripción:</b>	Controla matriculas y calificaciones de los estudiantes

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Boletines	Alumnos (Entidad Externa)	X	
Comprobante de matricula	Alumnos (Entidad Externa)	X	
Datos de los padres	Alumnos (Entidad Externa)		X
Datos de profesores	Profesores (Entidad Externa)		X
Datos del alumno	Alumnos (Entidad Externa)		X
Datos del representante	Alumnos (Entidad Externa)		X
Datos no aprobados	Alumnos (Entidad Externa)	X	
Notas del estudiante	Profesores (Entidad Externa)		X

### **Proceso Generar reportes**

<b>Nombre:</b>	Generar reportes
<b>Código:</b>	PRCS_31
<b>Número:</b>	5
<b>Descripción:</b>	Genera los reportes con los datos de los alumnos

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Boletines	Alumnos (Entidad Externa)	X	
Datos del alumno	Alumnos (Repositorio datos)		X
Notas del estudiante	Notas (Repositorio datos)		X

### Proceso Ingresar notas

<b>Nombre:</b>	Ingresar notas
<b>Código:</b>	PRCS_26
<b>Número:</b>	4
<b>Descripción:</b>	Permite ingresar las notas de los alumnos a los profesores

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Datos de profesores	Profesores (Entidad Externa)		X
Notas de estudiantes	Notas (Repositorio datos)	X	
Notas del estudiante	Profesores (Entidad Externa)		X

### Proceso Matricular

<b>Nombre:</b>	Matricular
<b>Código:</b>	PRCS_23
<b>Número:</b>	3
<b>Descripción:</b>	Permite ingresar los datos completos de los alumnos con todos los requisitos que se solicitan en la institución.

Vía de conexión:	Conectado a:	Salidas	Entradas
Comprobante de matricula	Alumnos (Entidad Externa)	X	
Comprobante de pago	Pagar matricula (Proceso)		X
Datos completos del alumno	Alumnos (Repositorio datos)	X	
Datos de los padres	Alumnos (Entidad Externa)		X
datos de pensión	Pensiones (Repositorio datos)	X	
Datos del representante	Alumnos (Entidad Externa)		X

### Proceso Pagar matricula

<b>Nombre:</b>	Pagar matricula
<b>Código:</b>	PRCS_22
<b>Número:</b>	2
<b>Descripción:</b>	Controla el pago de la matricula y genera un papeleta del pago

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Comprobante de pago	Matricular (Proceso)	X	
Datos aprobados	Verificar datos (Proceso)		X
Datos del pago	Pensiones (Repositorio datos)	X	

### Proceso Verificar datos

<b>Nombre:</b>	Verificar datos
<b>Código:</b>	PRCS_21
<b>Número:</b>	1
<b>Descripción:</b>	Verifica si los datos del alumno son aceptados o no en la institución.

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Datos aprobados	Pagar matricula (Proceso)	X	
Datos de alumnos	Alumnos (Repositorio datos)	X	
Datos del alumno	Alumnos (Entidad Externa)		X
Datos no aprobados	Alumnos (Entidad Externa)	X	

### Proceso Crear ficha

<b>Nombre:</b>	Crear ficha
<b>Código:</b>	PRCS_38
<b>Número:</b>	1.2
<b>Descripción:</b>	Permite crear una ficha de cada alumno en la base de datos.

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos aprobados	Pagar matricula (Proceso)	X	
Datos de alumnos	Alumnos (Repositorio datos)	X	
Datos del alumno	Revisar requisitos (Proceso)		X

### Proceso Revisar requisitos

<b>Nombre:</b>	Revisar requisitos
<b>Código:</b>	PRCS_37
<b>Número:</b>	1.1
<b>Descripción:</b>	Verifica si los requisitos están completos y si son correctos

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos del alumno	Alumnos (Entidad Externa)		X
Datos del alumno	Crear ficha (Proceso)	X	
Datos no aprobados	Alumnos (Entidad Externa)	X	

### Proceso Calcular descuento

<b>Nombre:</b>	Calcular descuento
<b>Código:</b>	PRCS_47
<b>Número:</b>	2.2
<b>Descripción:</b>	Calcula el descuento de la matricula y pensión de los alumnos becados de la institución.

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos de alumnos	Calcular valor matricula (Proceso)	X	
Datos de alumnos becados	Verificar beca (Proceso)		X

### Proceso Calcular valor matricula

<b>Nombre:</b>	Calcular valor matricula
<b>Código:</b>	PRCS_48
<b>Número:</b>	2.3
<b>Descripción:</b>	Calcula el valor final a pagar de la matricula del alumno

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Datos de alumnos	Calcular descuento (Proceso)		X
Datos de alumnos	Verificar beca (Proceso)		X
Recibo de pago	Realizar pago (Proceso)	X	

### Proceso Realizar pago

<b>Nombre:</b>	Realizar pago
<b>Código:</b>	PRCS_49
<b>Número:</b>	2.4
<b>Descripción:</b>	Permite realizar el pago ya sea en el banco o en colectaría.

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Comprobante de pago	Matricular (Proceso)	X	
Datos del pago	Pensiones (Repositorio datos)	X	
Recibo de pago	Calcular valor matricula (Proceso)		X

### Proceso Verificar beca

<b>Nombre:</b>	Verificar beca
<b>Código:</b>	PRCS_46
<b>Número:</b>	2.1
<b>Descripción:</b>	Verifica si el alumno tiene beca

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos aprobados	Verificar datos (Proceso)		X
Datos de alumnos	Calcular valor matricula (Proceso)	X	
Datos de alumnos becados	Calcular descuento (Proceso)	X	

### Proceso Registrar datos

<b>Nombre:</b>	Registrar datos
<b>Código:</b>	PRCS_61
<b>Número:</b>	3.1
<b>Descripción:</b>	Permite registrar todos los datos completos del alumno, padres y representante.

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Comprobante de matricula	Alumnos (Entidad Externa)	X	
Comprobante de pago	Pagar matricula (Proceso)		X
Datos completos del alumno	Alumnos (Repositorio datos)	X	
Datos de alumnos	Registrar pensiones (Proceso)	X	
Datos de los padres	Alumnos (Entidad Externa)		X
Datos del representante	Alumnos (Entidad Externa)		X

### Proceso Registrar pensiones

<b>Nombre:</b>	Registrar pensiones
<b>Código:</b>	PRCS_62
<b>Número:</b>	3.2
<b>Descripción:</b>	Ayuda a registrar los datos de las pensiones del alumno

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos de alumnos	Registrar datos (Proceso)		X
datos de pensión	Pensiones (Repositorio datos)	X	

### Proceso Ingresar datos de profesores

<b>Nombre:</b>	Ingresar datos de profesores
<b>Código:</b>	PRCS_127
<b>Número:</b>	4.1
<b>Descripción:</b>	Permite registrar la información de todos los profesores que pertenecen a la institución

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos de profesores	Ingresar notas (Proceso)	X	
Datos de profesores	Profesores (Repositorio datos)	X	
Datos de profesores	Profesores (Entidad Externa)		X

### Proceso Ingresar notas

<b>Nombre:</b>	Ingresar notas
<b>Código:</b>	PRCS_128
<b>Número:</b>	4.2
<b>Descripción:</b>	Permite ingresar las notas a los profesores para luego guardarlos en el repositorio notas

<b>Vía de conexión:</b>	<b>Conectado a:</b>	<b>Salidas:</b>	<b>Entradas:</b>
Datos de profesores	Ingresar datos de profesores (Proceso)		X
Notas de estudiantes	Notas (Repositorio datos)	X	
Notas del estudiante	Profesores (Entidad Externa)		X

### Proceso Consultar notas

<b>Nombre:</b>	Consultar notas
<b>Código:</b>	PRCS_136
<b>Número:</b>	5.1
<b>Descripción:</b>	Permite consultar las calificaciones que tienen los alumnos

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Notas del estudiante	Notas (Repositorio datos)		X
Notas del estudiante	Generar boletines (Proceso)	X	

### Proceso Generar boletines

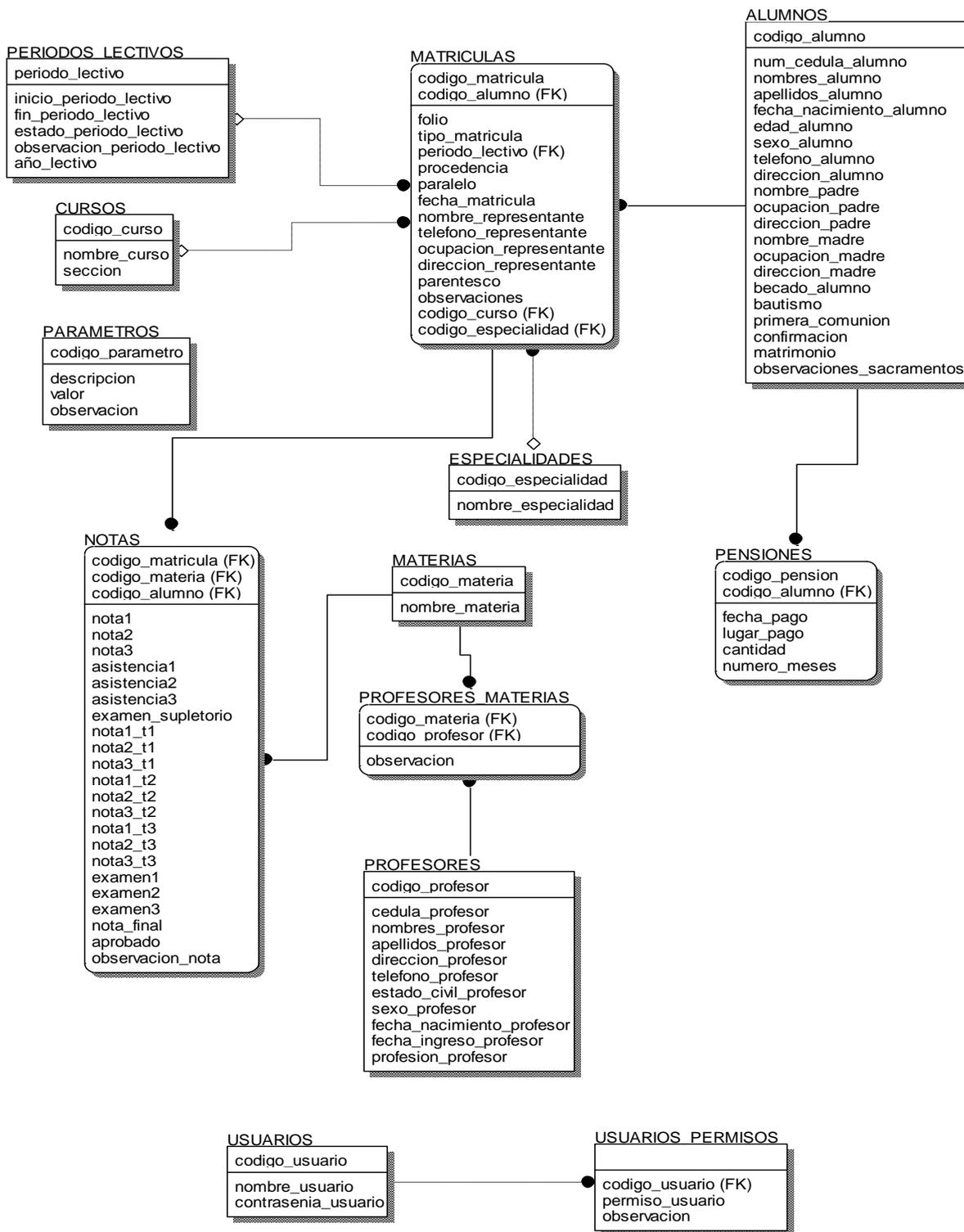
<b>Nombre:</b>	Generar boletines
<b>Código:</b>	PRCS_137
<b>Número:</b>	5.2
<b>Descripción:</b>	Ayuda a la generación de los boletines del estudiante

Vía de conexión:	Conectado a:	Salidas:	Entradas:
Boletines	Alumnos (Entidad Externa)	X	
Datos del alumno	Alumnos (Repositorio datos)		X
Notas del estudiante	Consultar notas (Proceso)		X

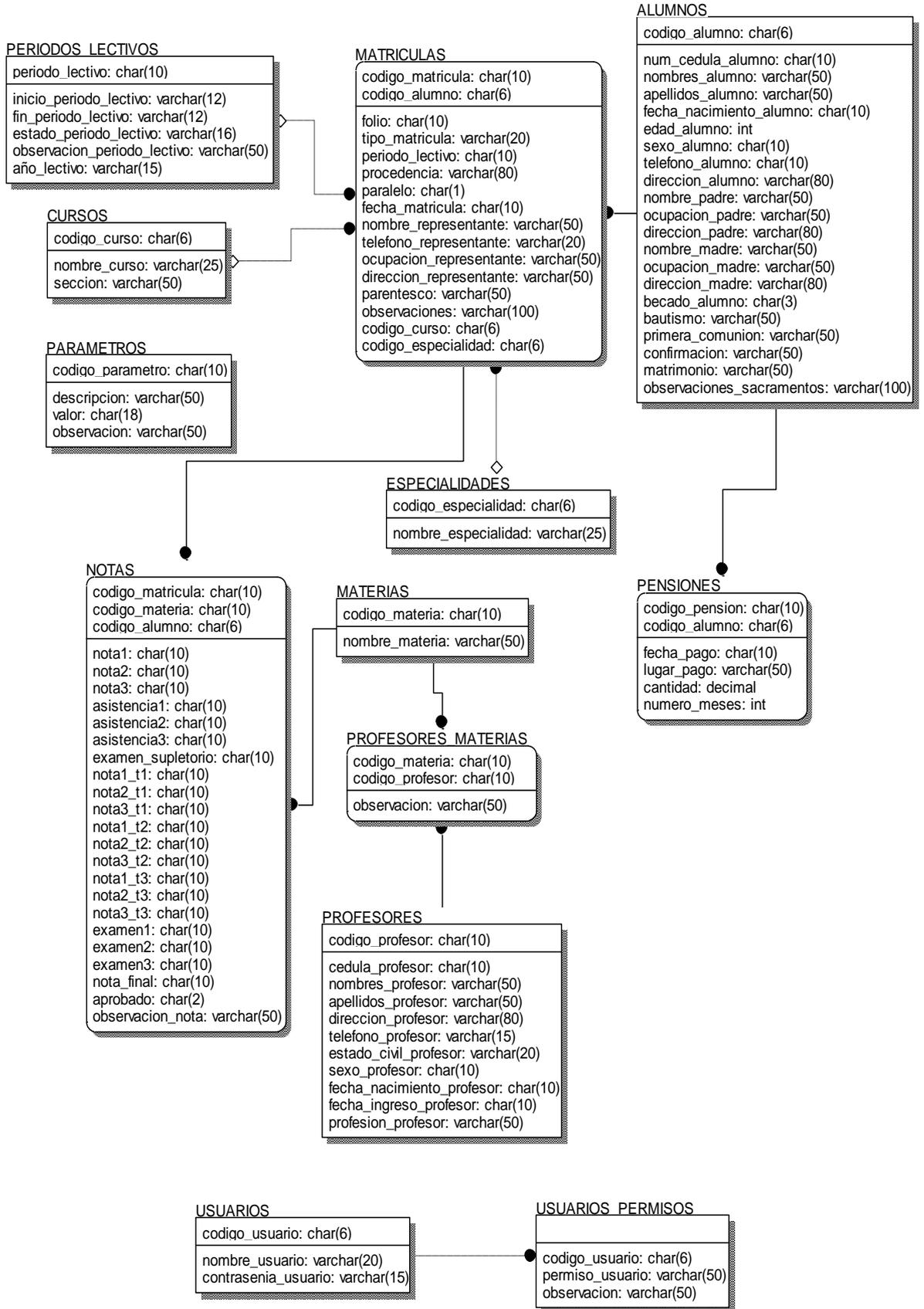
## CAPITULO V

### DISEÑO

#### 5.7 Diseño lógico



## 5.8 Diseño físico



## 5.9 Diccionario de datos del diseño físico

### 5.9.1 Estructura y descripción de las tablas

Tablas de datos	Detalle
ALUMNOS	Permite almacenar los datos personales de los alumnos
CURSOS	Almacena los cursos de la institución
ESPECIALIDADES	Aquí se almacena las especialidades que posee el establecimiento
MATERIAS	Permite guardar todas las materias
PARAMETROS	Permite almacenar todos los valores iniciales de un periodo lectivo
PENSIONES	Se guardan las pensiones pagadas por los alumnos
PERIODOS_LECTIVOS	Almacena todos los periodos registrados
PROFESORES	Todos los profesores de la institución
PROFESORES_MATERIAS	Tabla que relaciona los profesores con la s materias
USUARIOS	Los diferentes usuarios del sistema
USUARIOS_PERMISOS	Los permisos que se le asigna a cada usuario
MATRICULAS	Todos los alumnos matriculados actualmente
NOTAS	Permite almacenar las notas del año lectivo

### Campos de datos por tabla

<b>Tabla ALUMNOS</b>			
Nombre	Tipo	Tamaño	Descripción
codigo_alumno(PK)	Char	6	Código para cada alumno
num_cedula_alumno	char	10	Cedula del alumno
nombres_alumno	varchar	50	Nombres del alumno
apellidos_alumno	varchar	50	Apellidos del alumno
fecha_nacimiento_alumno	char	10	Fecha de nacimiento
edad_alumno	int		Edad del alumno
sexo_alumno	char	10	Sexo
telefono_alumno	char	10	Teléfono
direccion_alumno	varchar	80	Dirección
nombre_padre	varchar	50	Nombre del padre
ocupacion_padre	varchar	50	Ocupación del padre
direccion_padre	varchar	80	Dirección del padre
nombre_madre	varchar	50	Nombre de la madre
ocupacion_madre	varchar	50	Ocupación de la madre
direccion_madre	varchar	80	Dirección de la madre
becado_alumno	char	3	Si el alumno es becado
bautismo	varchar	50	Tiene bautismo
primera_comunion	varchar	50	Hizo primera comunión
confirmacion	varchar	50	Tiene confirmación
matrimonio	varchar	50	Si los padres son casados
observaciones_sacramentos	varchar	100	Observación de los sacramentos

<b>Tabla CURSOS</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_curso(PK)	char	6	Código asignado a cada curso
nombre_curso	varchar	25	Nombre del curso
seccion	varchar	50	Sección del curso

<b>Tabla ESPECIALIDADES</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_especialidad(PK)	char	6	Código de la especialidad
nombre_especialidad	varchar	25	Nombre especialidad

<b>Tabla MATERIAS</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_materia(PK)	char	10	Código de la materia
nombre_materia	varchar	50	Nombre de la materia

<b>Tabla PARAMETROS</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_parametro(PK)	char	10	Código del parámetro
descripción	varchar	50	Descripción del parámetro
valor	char	18	Costo de matricula y pensiones.
observación	varchar	50	Observación tenga uno de los valores ingresados.

<b>Tabla PENSIONES</b>				
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Tabla relacionada</b>	<b>Descripción</b>
codigo_pension(PK)	Char	10		Código de pensión
fecha_pago	char	10		Fecha de pago
lugar_pago	varchar	50		Lugar de pago
cantidad	decimal			Cantidad a pagar
numero_meses	int			Numero de meses
codigo_alumno	char	6	ALUMNOS	Código alumno

<b>Tabla PERIODOS LECTIVOS</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
periodo_lectivo (PK)	char	10	Código del periodo lectivo
inicio_periodo_lectivo	varchar	12	Inicio del periodo lectivo
fin_periodo_lectivo	varchar	12	Fin del periodo lectivo
estado_periodo_lectivo	varchar	16	Estado del periodo, es decir abierto o cerrado
observacion_periodo_lectivo	varchar	50	Observación periodo
año_lectivo	varchar	15	Año lectivo en el que esta trabajando.

<b>Tabla PROFESORES</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_profesor (PK)	char	10	Código del profesor
cedula_profesor	char	10	Cedula del profesor
nombres_profesor	varchar	50	Nombre Profesor
apellidos_profesor	varchar	50	Apellido profesor
direccion_profesor	varchar	80	Dirección profesor
telefono_profesor	varchar	15	Teléfono profesor
estado_civil_profesor	varchar	20	Estado civil
sexo_profesor	char	10	Sexo
fecha_nacimiento_profesor	char	10	Fecha de nacimiento
fecha_ingreso_profesor	char	10	Fecha que ingreso a trabajar en el establecimiento
profesion_profesor	varchar	50	Profesión que tiene el profesor

<b>Tabla PROFESORES MATERIAS</b>				
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Tabla relacionada</b>	<b>Descripción</b>
codigo_materia(PK)(FK)	char	10	MATERIAS	Código materia
codigo_profesor(PK)(FK)	char	10	PROFESORES	Código profesor
observacion	varchar	50		Observación

<b>Tabla USUARIOS</b>			
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Descripción</b>
codigo_usuario(PK)	char	6	Código del usuario
nombre_usuario	varchar	20	Nombre del usuario
contrasenia_usuario	varchar	15	Contraseña del usuario

<b>Tabla USUARIOS_PERMISOS</b>				
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Tabla relacionada</b>	<b>Descripción</b>
codigo_usuario(FK)	char	6	USUARIOS	Código del usuario
permiso_usuario	varchar	50		Permisos que tiene cada usuario
observacion	varchar	50		Observación que tenga el usuario

<b>Tabla MATRICULAS</b>				
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Tabla relacionada</b>	<b>Descripción</b>
codigo_matricula(PK)	char	10	PERIODOS_LECTIVOS	Código matricula
folio	char	10		Numero folio
tipo_matricula	varchar	20		Tipo matricula
periodo_lectivo(FK)	char	10		Código del periodo lectivo
procedencia	varchar	1		A que institución pertenecía un alumno
paralelo	char	10		Paralelo
fecha_matricula	char	50		Fecha matricula
nombre_representante	varchar	20		Nombre representante
telefono_representante	varchar	50		Fono representante
ocupacion_representante	varchar	50		Ocup_representante
direccion_representante	varchar	50	Dir representante	
parentesco	varchar	100	Parentesco que tiene el estudiante con el representante	
observaciones	varchar	6	CURSOS ALUMNOS ESPECIALIDADES	Observaciones que tenga un alumno
codigo_curso(FK)	char	6		Código de curso
codigo_alumno(PK)(FK)	char	6		Código del alumno
codigo_especialidad(FK)	char			Código de especialidad

<b>Tabla NOTAS</b>					
<b>Nombre</b>	<b>Tipo</b>	<b>Tamaño</b>	<b>Tabla relacionada</b>	<b>Descripción</b>	
codigo_matricula(PK)(FK)	char	10	MATRICULAS	Código matricula	
codigo_materia(PK)(FK)	char	10		MATERIAS	Código materia
nota1	char	10			Nota del 1 trimestre
nota2	char	10			Nota del 2 trimestre
nota3	char	10			Nota del 3 trimestre
asistencia1	char	10			Asistencia del 1 trimestre
asistencia2	char	10			Asistencia del 2 trimestre
asistencia3	char	10			Asistencia del 3 trimestre
examen_supletorio	char	10			Nota del examen de supletorio.
codigo_alumno(PK)	char	6		ALUMNOS	Código del alumno
nota1_t1	char	10			Nota del 1 mes
nota2_t1	char	10			Nota del 2 mes
nota3_t1	char	10			Nota del 3 mes
nota1_t2	char	10			Nota del 4 mes
nota2_t2	char	10			Nota del 5 mes
nota3_t2	char	10			Nota del 6 mes
nota1_t3	char	10			Nota del 7 mes
nota2_t3	char	10			Nota del 8 mes
nota3_t3	char	10			Nota del 9 mes
examen1	char	10			Nota del 1 examen
examen2	char	10			Nota del 2 examen
examen3	char	10			Nota del 3 examen
nota_final	char	10			Suma total de notas
aprobado	char	2			Registra si aprueba la materia o no
observacion_nota	varchar	50		Observacion de la notas del alumno	

## 5.10 Diseño de entradas

### Formulario de Notas

**INGRESO DE NOTAS DE UN ESTUDIANTE**

Alumno:

Año:

Paralelo:

Especialidad:

Materia:

Campos editables

**NUEVA INGRESO**

DETALLE	NOTAS
Primer Mes	<input type="text"/>
Segundo Mes	<input type="text"/>
Tercer Mes	<input type="text"/>
Examen 1.T	<input type="text"/>
Cuarto Mes	<input type="text"/>
Quinto Mes	<input type="text"/>
Sexto Mes	<input type="text"/>
Examen 2.T	<input type="text"/>
Séptimo Mes	<input type="text"/>
Octavo Mes	<input type="text"/>
Noveno Mes	<input type="text"/>
Examen 3.T	<input type="text"/>

Campos editables

**GUARDAR NOTAS**

**CANCELAR**

**SALIR**

Botones de acción

E t i q u e t a s

Botones de navegación

|◀   ◀   ▶   ▶|

## Formulario de matriculas

MATRICULACIÓN DE ALUMNOS	
E t i q u e t a s	N. - De Matricula: <input type="text"/>
	Nombres: <input type="text"/>
	Apellidos: <input type="text"/>
	Fecha de Nacimiento: <input type="text"/>
	Sexo: <input type="text"/>
	Teléfono: <input type="text"/>
	Dirección: <input type="text"/>
	Nombre del padre: <input type="text"/>
	Ocupación del padre: <input type="text"/>
	Nombre de la madre: <input type="text"/>
Ocupación de la madre: <input type="text"/>	
DATOS DEL REPRESENTANTE:	
Nombre: <input type="text"/>	
Ocupación: <input type="text"/>	
Dirección: <input type="text"/>	
Teléfono: <input type="text"/>	
Año: <input type="text"/>	
Paralelo: <input type="text"/>	
Especialidad: <input type="text"/>	
Campos editables	
Botones de acción	
NUEVA MATRICULA	
GUARDAR DATOS	
CANCELAR	
SALIR	
Botones de navegación	
◀ ▶	

## 5.11 Diseño de Salidas

### Reporte de Alumnos Matriculados

	<b>UNIDAD EDUCATIVA "GONZÁLEZ SUÁREZ"</b>
	<b>JOSEFINOS DE MURIALDO</b>
Pichincha y Quiz Quiz	2843993 Ambato

**LISTA DE ALUMNOS MATRICULADOS**

Año: ..... }  
Paralelo: ..... } Detalle  
Especialidad: ..... }

N.-	NOMINA
1	.....
2	.....
3	.....
.	.....
.	.....
Etc.	.....

**1**  
# de página

## Reporte de Certificados de Alumnos Matriculados



UNIDAD EDUCATIVA "GONZÁLEZ SUÁREZ"

JOSEFINOS DE MURIALDO

Pichincha y Quiz Quiz

2843993 Ambato

### CERTIFICADO DE MATRICULA

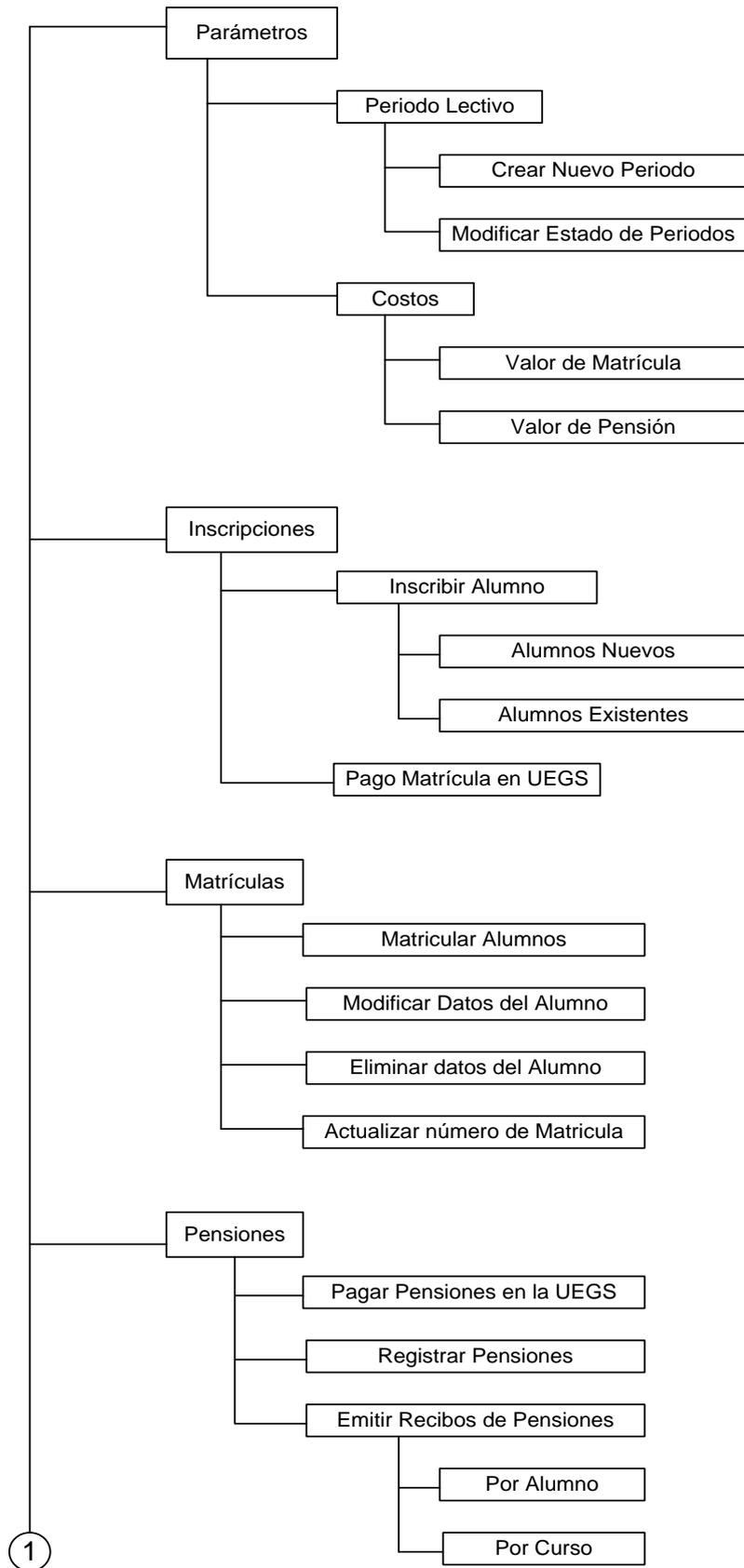
El Rector y la secretaria certifican que en los registros del plantel, previo el cumplimiento de los requisitos legales, ha sido inscrita la siguiente matricula:

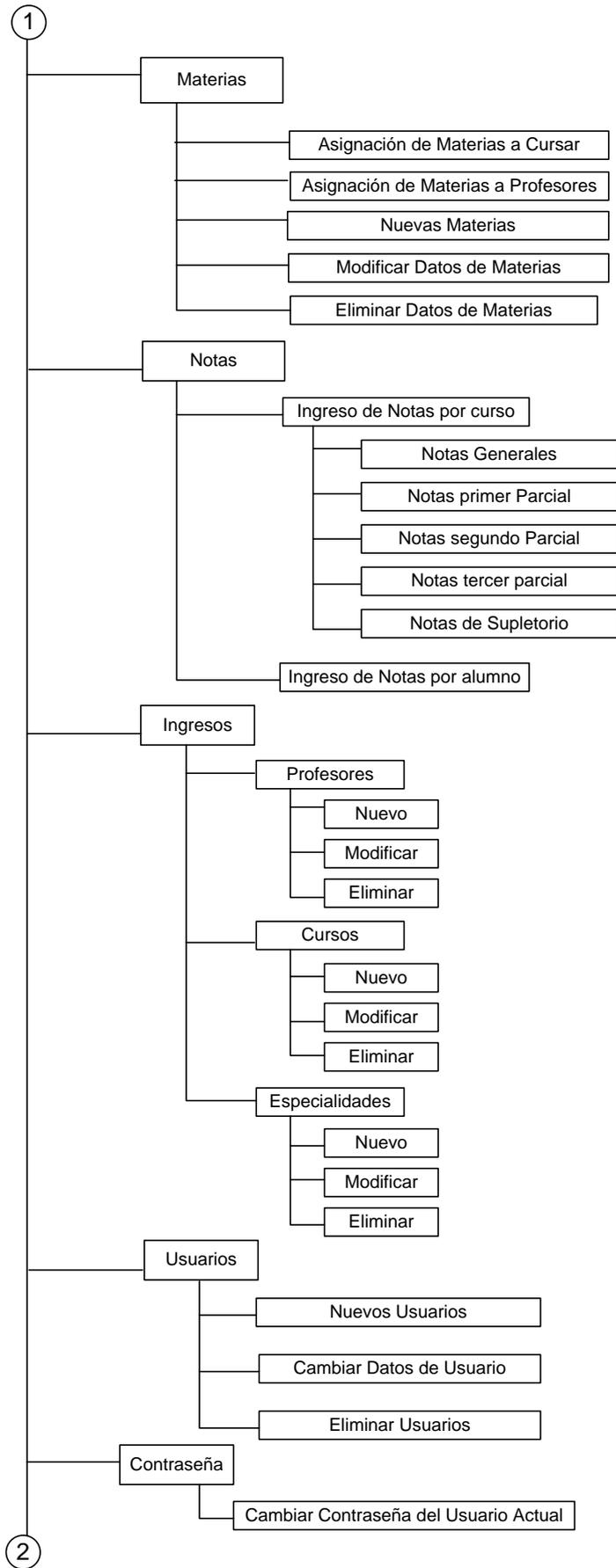
Nombre:	.....	} Detalle
Matricula N.-	.....	
Folio:	.....	
Año – Lectivo:	.....	
Curso:	.....	
Paralelo:	.....	
Especialidad:	.....	
Fecha:	.....	

-----  
RECTOR

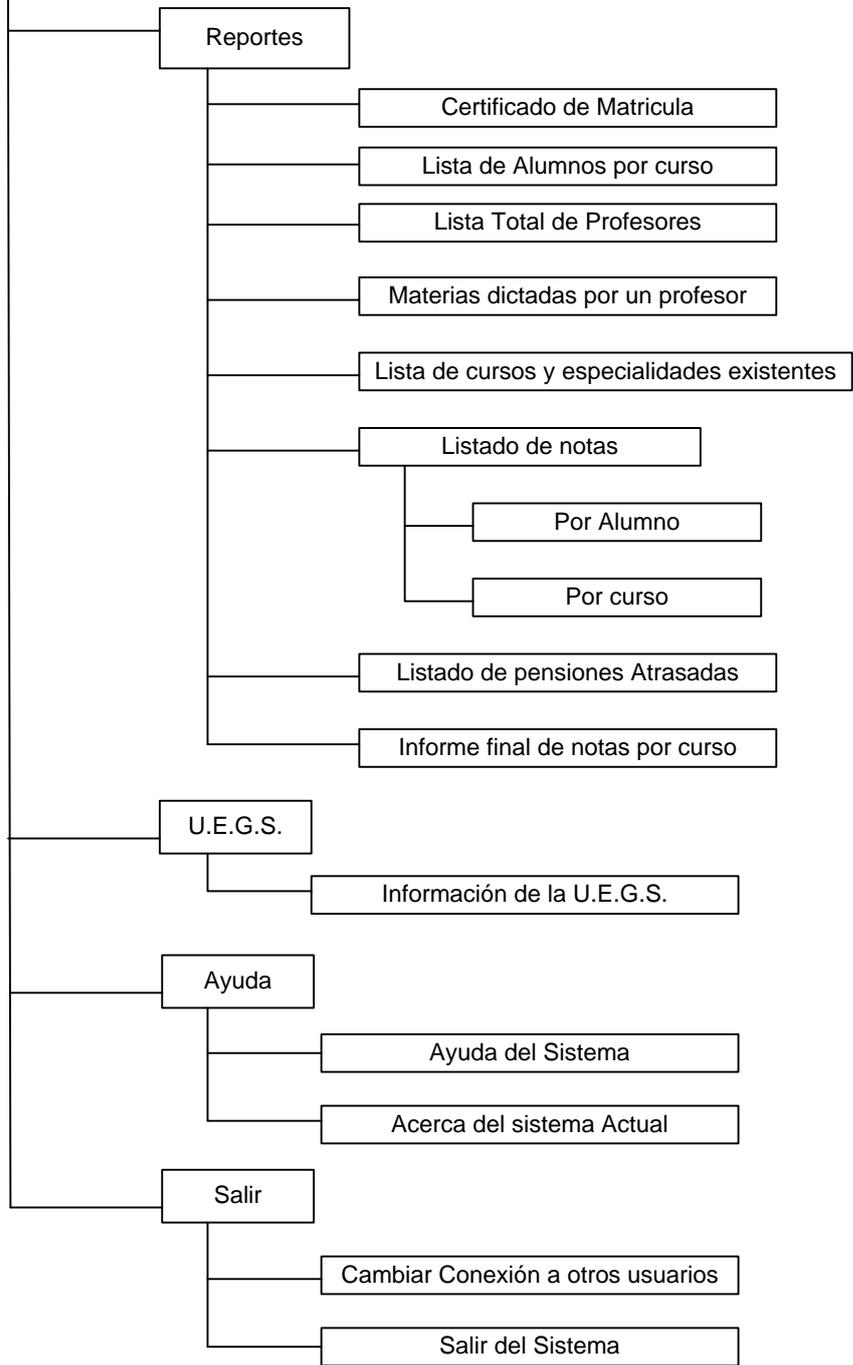
-----  
SECRETARIO

## 5.12 Diagrama de Procesos





2



## CAPITULO VI

### IMPLEMENTACIÓN Y PRUEBAS

#### 6.3 Implementación

Para la implementación del Sistema se ha tomado en cuenta, todos los requerimientos y necesidades de la institución, ya que por ser un sistema de gran importancia para ella, debe de contener la solución a todas sus necesidades.

#### Codificación del sistema

#### Función que permite generar claves para algunas tablas de la base de datos

```
Function codigo3(cadena As String) As String
Dim letra2 As String, letra As String
Dim largo As Integer, codigo As Long
largo = Len(cadena)
letra = Mid(cadena, 2, largo)
codigo = Trim(CLng(letra) + 1)
If codigo > 9 Then
    If codigo >= 100 Then
        If codigo >= 1000 Then
            If codigo >= 10000 Then
                letra2 = Mid(cadena, 1, 1) & CStr(codigo)
            Else
                letra2 = Mid(cadena, 1, 2) & CStr(codigo)
            End If
        Else
            letra2 = Mid(cadena, 1, 3) & CStr(codigo)
        End If
    Else
        letra2 = Mid(cadena, 1, 3) & 0 & CStr(codigo)
    End If
Else
    letra2 = Mid(cadena, 1, 3) & 0 & 0 & CStr(codigo)
End If
codigo3 = letra2
End Function
```

## Código del modulo matriculas

### VERIFICA SI YA ESTA HABILITADO UN PERIODO LECTIVO

```
Private Sub Form_Load()
Set RSp1 = New Recordset
RSp1.Open "SELECT periodo_lectivo,
inicio_periodo_lectivo,fin_periodo_lectivo,estado_periodo_lectivo,observacion_p
eriodo_lectivo,año_lectivo from PERIODOS_LLECTIVOS where
estado_periodo_lectivo='ABIERTO'", comm, adOpenStatic, adLockOptimistic
If RSp1.RecordCount > 0 Then
matricula_alumno.Item(4).Text = RSp1!año_lectivo
AÑO_LLECTIVO_BASE = RSp1!año_lectivo
End If
matricula_alumno.Item(1).Enabled = False
matricula_alumno.Item(7).Enabled = False
matricula_alumno.Item(4).Enabled = False
alumno.Item(6).Enabled = False
cancelar.Enabled = False
guardar.Enabled = False
    Frame1.Enabled = False
    Frame2.Enabled = False
    Frame3.Enabled = False
    Frame4.Enabled = False
Frame6.Enabled = False
Call vaciar_textbox
End Sub
```

### GENERA UN NUEVO CODIGO DE MATRICULA PARA EL ALUMNO

```
Private Sub generar_codigo_matricula()
'On Error GoTo codigo_matricula
    Dim cod_final As String
    Set RS1 = New Recordset
    RS1.Open "SELECT codigo_matricula FROM MATRICULAS order by
codigo_matricula asc ", comm, adOpenStatic, adLockOptimistic
    If RS1.RecordCount > 0 Then
        RS1.MoveLast
        cod_final = RS1!codigo_matricula
        comp = codigo3(cod_final)
    Else
        comp = "M00001"
    End If 'inserta nuevo codigo
matricula_alumno.Item(1).Enabled = True
matricula_alumno.Item(1).Text = comp
matricula_alumno.Item(1).Enabled = False
RS1.Close
'Exit Sub
'codigo_matricula:
End Sub
```

## GENERA UN CODIGO PARA CADA ALUMNO

```
Private Sub generar_codigo_alumno()  
'On Error GoTo codigo_alumno  
    Dim cod_final As String, nuevo_cod As String  
    Set RS1 = New Recordset  
    RS1.Open "SELECT codigo_alumno FROM ALUMNOS order by  
codigo_alumno asc ", comm, adOpenStatic, adLockOptimistic  
    If RS1.RecordCount > 0 Then  
        RS1.MoveLast  
        cod_final = RS1!codigo_alumno  
        nuevo_cod = codigo3(cod_final)  
    Else  
        nuevo_cod = "A00001"  
    End If  
    'inserta nuevo codigo  
    alumno.Item(1).Text = nuevo_cod  
    matricula_alumno(15).Text = nuevo_cod  
    RS1.Close  
'Exit Sub  
'codigo_alumno:  
End Sub
```

## PERMITE RECUPERAR DATOS DEL SISTEMA DE UN ALUMNO

```
Private Sub CARGAR_DATOS()  
On Error GoTo Control  
alumno.Item(1) = RS!codigo_alumno  
alumno.Item(2) = RS!num_cedula_alumno  
alumno.Item(3) = RS!nombres_alumno  
alumno.Item(4) = RS!apellidos_alumno  
alumno.Item(5) = RS!fecha_nacimiento_alumno  
calendario.Value = RS!fecha_nacimiento_alumno  
alumno.Item(6) = RS!edad_alumno  
alumno.Item(7) = RS!sexo_alumno  
alumno.Item(8) = RS!telefono_alumno  
alumno.Item(9) = RS!direccion_alumno  
alumno.Item(10) = RS!nombre_padre  
alumno.Item(11) = RS!ocupacion_padre  
alumno.Item(12) = RS!direccion_padre  
alumno.Item(13) = RS!nombre_madre  
alumno.Item(14) = RS!ocupacion_madre  
alumno.Item(15) = RS!direccion_madre  
alumno.Item(16) = RS!becado_alumno  
cadena_bautismo = RSN1!bautismo  
cadena_comunion = RSN1!primera_comunion  
cadena_confirmacion = RSN1!confirmacion  
cadena_matrimonio = RSN1!matrimonio
```

```

cadena_observaciones = RSN1!observaciones_sacramentos
matricula_alumno.Item(1) = RS2!codigo_matricula
matricula_alumno.Item(2) = RS2!folio
matricula_alumno.Item(3) = RS2!tipo_matricula
matricula_alumno.Item(4) = RS2!periodo_lectivo
matricula_alumno.Item(5) = RS2!procedencia
matricula_alumno.Item(6) = RS2!paralelo
matricula_alumno.Item(7) = RS2!fecha_matricula
matricula_alumno.Item(8) = RS2!nombre_representante
matricula_alumno.Item(9) = RS2!telefono_representante
matricula_alumno.Item(10) = RS2!ocupacion_representante
matricula_alumno.Item(11) = RS2!direccion_representante
matricula_alumno.Item(12) = RS2!PARENTESCO
matricula_alumno.Item(13) = RS2!observaciones
matricula_alumno.Item(14) = RS2!codigo_curso
matricula_alumno.Item(15) = RS2!codigo_alumno
matricula_alumno.Item(16) = RS2!codigo_especialdiad
Exit Sub
Control:
End Sub

```

## **GUARDA LA INFORMACION DEL ALUMNO EN LA BASE DE DATOS**

```

Private Sub guardar_Click()
'On Error GoTo error_guardar
Dim sql As String, edad_a As Variant
Dim respuesta As String, bandera As Integer
Dim resp As String
Call pasar_datos_textbox
'VERIFICAR CEDULA
If alumno(2).Text = "" Then
    bandera = 0
    resp = vbYes
    respuesta = ""
Else
    bandera = 1
    respuesta = VERIFICAR_CEDULA(alumno(2).Text)
End If
If (respuesta = "incorrecta") Then
resp = MsgBox("El número de cédula ingresado es incorrecto" + " " + "Desea guardar los datos con la cedula incorrecta?", vbYesNo + vbInformation, "Control Alumnos")
Else
resp = vbYes
End If
If resp = vbYes Then 'guarda
'VERIFICAR EDAD
Call calcular_edad

```

```

edad_a = CVar(alumno.Item(6).Text)
If (edad_a >= 4 And edad_a < 25) Then
    Call verificar_datos_completos
    If espacios_blanco = 0 Then
        Call guardar_datos_base
        RSN1.Update
        RSN1.Requery
        RSN1.MoveLast
        RSN2.Update
        RSN2.Requery
        RSN2.MoveLast
        MsgBox "DATOS GUARDADOS CORRECTAMENTE", vbInformation,
"BASE DE DATOS"
        cancelar.Enabled = False
        guardar.Enabled = False
        nuevo.Enabled = True
        RSN1.Close
        RSN2.Close
        cancelar.Enabled = False
        guardar.Enabled = False
        nuevo.Enabled = True
        Frame1.Enabled = False
        Frame2.Enabled = False
        Frame3.Enabled = False
        Frame4.Enabled = False
        Frame6.Enabled = False
    Else
        MsgBox "Faltan datos requeridos, complete correctamente los datos",
vbInformation, "Datos Incompletos"
    End If
Else
    MsgBox "La edad ingresada del estudiante no es aceptada, Favor modificar",
vbInformation, "Edad no válida"
End If
Else
alumno.Item(2).SetFocus
End If 'fin de control cedula
'Exit Sub
'error_guardar:
End Sub

```

#### **'datos completos del alumno**

```

Private Sub verificar_datos_completos()
'On Error GoTo error_datos_completos
espacios_blanco = 0
For i = 1 To 16 Step 1
    If alumno.Item(i) = "" Then
        If (i <> 2 And i <> 8) Then

```

```

        espacios_blanco = 1
    End If
End If
Next
For j = 1 To 16 Step 1
If (j <> 2) Then
    If matricula_alumno.Item(j) = "" Then
        If (j <> 2 And j <> 9 And j <> 13) Then
            espacios_blanco = 1
        End If
    End If
End If
End If
Next
'Exit Sub
'error_datos_completos:
End Sub

```

### **'Calcular edad**

```

Private Sub calcular_edad()
'On Error GoTo error_edad
Dim cadena_edad As Date
Dim n_dias As Double
Dim anio As Integer
Dim meses As Integer
    cadena_edad = calendario.Value
    n_dias = Date - cadena_edad
    anio = Int(n_dias / 365.25)
    n_dias = n_dias - (365.25 * anio)
    meses = Int(n_dias / 30.4375)
    n_dias = Int(n_dias - (30.4375 * meses))
edad_años = anio
'MsgBox Str(edad_dia) & " " & "DIAS" & " " & Str(meses) & " " & "MESES" &
" " & Str(n_dias) & " " & "AÑOS", vbInformation, "EDAD DEL ESTUDIANTE"
alumno.Item(6).Enabled = True
alumno.Item(6).Text = edad_años
alumno.Item(6).Enabled = False
meses_ver.Caption = Str(meses)
años.Caption = Str(edad_años)
'Exit Sub
'error_edad:
End Sub

```

### **'almacenar datos del alumno**

```

Private Sub guardar_datos_base()
'On Error GoTo error_guardar_datos
RSN1!codigo_alumno = alumno.Item(1)
RSN1!num_cedula_alumno = alumno.Item(2)
RSN1!nombres_alumno = UCase(Trim(alumno.Item(3)))

```

```

RSN1!apellidos_alumno = UCase(Trim(alumno.Item(4)))
RSN1!fecha_nacimiento_alumno = alumno.Item(5)
RSN1!edad_alumno = alumno.Item(6)
RSN1!sexo_alumno = UCase(Trim(alumno.Item(7)))
RSN1!telefono_alumno = alumno.Item(8)
RSN1!direccion_alumno = alumno.Item(9)
RSN1!nombre_padre = UCase(Trim(alumno.Item(10)))
RSN1!ocupacion_padre = UCase(Trim(alumno.Item(11)))
RSN1!direccion_padre = alumno.Item(12)
RSN1!nombre_madre = UCase(Trim(alumno.Item(13)))
RSN1!ocupacion_madre = UCase(Trim(alumno.Item(14)))
RSN1!direccion_madre = alumno.Item(15)
RSN1!becado_alumno = UCase(Trim(alumno.Item(16)))
RSN1!bautismo = cadena_bautismo
RSN1!primera_comunion = cadena_comunion
RSN1!confirmacion = cadena_confirmacion
RSN1!matrimonio = cadena_matrimonio
RSN1!observaciones_sacramentos = cadena_observaciones
RSN2!codigo_matricula = matricula_alumno.Item(1)
RSN2!tipo_matricula = matricula_alumno.Item(3)
RSN2!periodo_lectivo = matricula_alumno.Item(4)
RSN2!procedencia = UCase(Trim(matricula_alumno.Item(5)))
RSN2!paralelo = UCase(Trim(matricula_alumno.Item(6)))
RSN2!fecha_matricula = matricula_alumno.Item(7)
RSN2!nombre_representante = UCase(Trim(matricula_alumno.Item(8)))
RSN2!telefono_representante = matricula_alumno.Item(9)
RSN2!ocupacion_representante = UCase(Trim(matricula_alumno.Item(10)))
RSN2!direccion_representante = matricula_alumno.Item(11)
RSN2!PARENTESCO = UCase(Trim(matricula_alumno.Item(12)))
RSN2!observaciones = UCase(Trim(matricula_alumno.Item(13)))
RSN2!codigo_curso = matricula_alumno.Item(14)
RSN2!codigo_alumno = matricula_alumno.Item(15)
RSN2!codigo_especialdiad = matricula_alumno.Item(16)
'Exit Sub
'error_guardar_datos:
End Sub

```

### **Verifica si la cedula es correcta**

Function VERIFICAR\_CEDULA(cadena\_cedula As String) As String

Dim veri, decena, suma\_total, resta, suma\_impar, auxdig, auxdig1, numero,  
dig, impar, auximpar, par, suma\_par As Integer

numero = cadena\_cedula

auximpar = suma\_impar = suma\_par = suma\_total = 0

For impar = 1 To 9 Step 2

auxdig = Val(Mid(numero, impar, 1))

auximpar = auxdig \* 2

If auximpar > 9 Then

auximpar = auximpar - 9

```

    End If
    suma_impar = suma_impar + auximpar
Next impar
For par = 2 To 8 Step 2
    auxdig = Val(Mid(numero, par, 1))
    suma_par = suma_par + auxdig
Next par
suma_total = suma_par + suma_impar
decena = (Int(suma_total / 10) * 10) + 10
veri = decena - suma_total
aux_veri = (Val(Mid(numero, 10, 1)))
If Not (veri = aux_veri) Then
    VERIFICAR_CEDULA = "incorrecta"
Else
    VERIFICAR_CEDULA = "correcta"
End If
End Function

```

### **PERMITE REALIZAR EL INGRESO DE UNA NUEVA MATRICULA**

```

Private Sub nuevo_Click()
'On Error GoTo error_añadir
Call vaciar_textbox
Call cargar_combos
Call generar_codigo_matricula
Call generar_codigo_alumno
'Periodo lectivo
Dim fecha_actual As String, año_actual As Integer
fecha_actual = Date
año_actual = Val(Mid(fecha_actual, 7, 4))
matricula_alumno.Item(7).Enabled = True
matricula_alumno.Item(4).Enabled = True
matricula_alumno.Item(4).Text = AÑO_LECTIVO_BASE
matricula_alumno.Item(7).Text = CStr(Date)
matricula_alumno.Item(7).Enabled = False
matricula_alumno.Item(4).Enabled = False
'Nuevo registro
Set RSN1 = New Recordset
Set RSN2 = New Recordset
RSN1.Open "SELECT
codigo_alumno,num_cedula_alumno,nombres_alumno,apellidos_alumno,fecha_n
acimiento_alumno,edad_alumno,sexo_alumno,telefono_alumno,direccion_alumn
o,nombre_padre,ocupacion_padre,direccion_padre,nombre_madre,ocupacion_ma
dre,direccion_madre,becado_alumno,bautismo,primera_comunion,confirmacion,
matrimonio,observaciones_sacramentos from ALUMNOS", comm, adOpenStatic,
adLockOptimistic
If RSN1.RecordCount > 0 Then
RSN1.MoveFirst

```

```

End If
RSN2.Open "SELECT codigo_matricula, tipo_matricula, periodo_lectivo,
procedencia,paralelo,fecha_matricula,nombre_representante,telefono_representant
e,ocupacion_representante,direccion_representante,parentesco,observaciones,codi
go_curso,codigo_alumno,codigo_especialidad from MATRICULAS", comm,
adOpenStatic, adLockOptimistic
If RSN2.RecordCount > 0 Then
RSN2.MoveFirst
End If
RSN1.AddNew
RSN2.AddNew
cancelar.Enabled = True
guardar.Enabled = True
nuevo.Enabled = False
Frame1.Enabled = True
Frame2.Enabled = True
Frame3.Enabled = True
Frame4.Enabled = True
Frame6.Enabled = True
'Exit Sub
'error_añadir:
' MsgBox "Error inesperado al agregar nuevas matriculas", vbInformation, "Base
de Datos"
End Sub

```

### **Código del modulo pensiones**

#### **FILTRA LOS ALUMNOS QUE ESTAN MATRICULADOS**

```

Private Sub buscar_datos()
Set RSbuscardatos = New Recordset
RSbuscardatos.Open "SELECT codigo_alumno asCodigo, num_cedula_alumno
as Cedula,[nombres_alumno]+' '+[apellidos_alumno] as Nomina,edad_alumno as
Edad,sexo_alumno as Sexo FROM ALUMNOS where esta_matriculado= 'SI' ",
comm, adOpenStatic, adLockOptimistic
If txt_opcion = "" Then
RSbuscardatos.Filter = ""
RSbuscardatos.Requery
Else
RSbuscardatos.Filter = "Nomina like "" & txt_opcion & ""%"
End If

Set grd_gridalumnos.DataSource = RSbuscardatos
grd_gridalumnos.Columns(2).Width = 3800
grd_gridalumnos.Columns(3).Width = 600
End Sub

```

## RECUPERA LOS DATOS DEL ALUMNO Y LOS DATOS DE PENSIONES

```
Private Sub buscar_alumnos()  
    'On Error GoTo error_buscar  
    Dim texto_buscar As String  
    Set RSbuscarpension = New Recordset  
    Set RSbuscaralumno = New Recordset  
    Set RSbuscarcurso = New Recordset  
    Set RSbuscarparalelo = New Recordset  
    Set RSbuscarespecialidad = New Recordset  
    Set RSbuscarmatricula = New Recordset  
    Set RSbuscarmeses = New Recordset  
    RSbuscarpension.Open "SELECT  
codigo_pension,fecha_pago,lugar_pago,cantidad,numero_meses,codigo_al  
umno FROM PENSIONES", comm, adOpenStatic, adLockOptimistic  
    RSbuscaralumno.Open "SELECT  
codigo_alumno,num_cedula_alumno,nombres_alumno,apellidos_alumno,f  
echa_nacimiento_alumno,edad_alumno,sexo_alumno,telefono_alumno,dir  
eccion_alumno,nombre_padre,ocupacion_padre,direccion_padre,nombre_  
madre,ocupacion_madre,direccion_madre,becado_alumno,bautismo,prime  
ra_comunion,confirmacion,matrimonio,observaciones_sacramentos  
FROM ALUMNOS", comm, adOpenStatic, adLockOptimistic  
    If RSbuscaralumno.RecordCount > 0 Then  
        texto_buscar = codigo_seleccionado  
        RSbuscaralumno.Requery  
        RSbuscaralumno.Close  
        RSbuscaralumno.Open "SELECT  
codigo_alumno,num_cedula_alumno,nombres_alumno,apellidos_alumno,f  
echa_nacimiento_alumno,edad_alumno,sexo_alumno,telefono_alumno,dir  
eccion_alumno,nombre_padre,ocupacion_padre,direccion_padre,nombre_  
madre,ocupacion_madre,direccion_madre,becado_alumno,bautismo,prime  
ra_comunion,confirmacion,matrimonio,observaciones_sacramentos,saldo  
FROM ALUMNOS WHERE codigo_alumno='" & texto_buscar & "'",  
comm, adOpenStatic, adLockOptimistic  
        tipo_pension = RSbuscaralumno!becado_alumno  
        RSbuscarpension.Close  
        RSbuscarpension.Open "SELECT  
codigo_pension,fecha_pago,lugar_pago,cantidad,numero_meses,codigo_al  
umno FROM PENSIONES WHERE codigo_alumno='" &  
RSbuscaralumno!codigo_alumno & "' ", comm, adOpenStatic,  
adLockOptimistic  
        RSbuscarmeses.Open "SELECT  
SUM(numero_meses)'suma_meses' FROM PENSIONES WHERE  
codigo_alumno='" & RSbuscaralumno!codigo_alumno & "' ", comm,  
adOpenStatic, adLockOptimistic  
        If RSbuscarmeses.RecordCount > 0 Then  
            meses_ya_pagados = RSbuscarmeses!suma_meses  
        End If  
    End If  
End Sub
```

```

        If RSbuscarmeses!suma_meses >= 10 Then
            MsgBox "El alumno " + RSbuscaralumno!apellidos_alumno + "
" + RSbuscaralumno!nombres_alumno + " ya tiene cancelado todas las
pensiones ", vbInformation, "Control Pensiones"
            Call cancelar_Click
        Else
            RSbuscarmatricula.Open "select
codigo_matricula,folio,tipo_matricula,periodo_lectivo,procedencia,paralel
o,fecha_matricula,nombre_representante,telefono_representante,ocupacion
_representante,direccion_representante,parentesco,observaciones,codigo_c
urso,codigo_alumno,codigo_especialidad from MATRICULAS WHERE
codigo_alumno=" & RSbuscaralumno!codigo_alumno & "" ", comm,
adOpenStatic, adLockOptimistic
            If RSbuscarmatricula.RecordCount > 0 Then
                RSbuscarmatricula.Open "SELECT
codigo_curso,nombre_curso,seccion FROM CURSOS WHERE
codigo_curso=" & RSbuscarmatricula!codigo_curso & "" ", comm,
adOpenStatic, adLockOptimistic
                'si esque tiene especialidad
                no_especialidad = "NO"
                For j = 11 To 13 Step 1
                    Dim cod_cur As String
                    cod_cur = "C000" + Str(j)
                    If RSbuscarmatricula!codigo_curso = cod_cur Then
                        RSbuscarmatricula.Open "SELECT
codigo_especialidad,nombre_especialidad FROM ESPECIALIDADES
WHERE codigo_especialidad=" &
RSbuscarmatricula!codigo_especialidad & "" ", comm, adOpenStatic,
adLockOptimistic
                            no_especialidad = "SI"
                        End If
                    Next
                If RSbuscarmatricula.RecordCount > 0 Then
                    num_m = RSbuscarmeses!suma_meses
                    Call LIMPIAR_CHECK
                    For i = 1 To num_m Step 1
                        check_mes.Item(i).Value = 1
                        check_mes.Item(i).Enabled = False
                    Next
                    If num_m < 10 Then
                        For i = num_m + 1 To 10 Step 1
                            check_mes.Item(i).Enabled = True
                        Next
                    End If
                Else
                    MsgBox "No tiene pagado ninguna pensión", vbInformation,
"Aviso"
                    Call LIMPIAR_CHECK
                End If
            End If
        End If
    End Sub

```

```

        End If
        Frame_pension.Enabled = True
        Call CARGAR_DATOS_PENSIONES
        nuevo.Enabled = True

    Else
        MsgBox "El alumno no esta matriculado ", vbInformation,
"Control Alumnos"
        grilla.Visible = True
        txt_opcion.SetFocus
        End If
    End If
Else
MsgBox "No existe ningún registro de alumnos en la Base de Datos",
vbInformation, "Aviso"
End If
'Exit Sub
'error_buscar:
' MsgBox "Error inesperado al buscar datos", vbInformation, "Base de
Datos"
End Sub

Private Sub CARGAR_DATOS_PENSIONES()
'On Error GoTo Control
Set RSsaldo = New Recordset
Set RSsuma_pensiones = New Recordset
lblalumno.Item(1) = RSbuscaralumno!apellidos_alumno + " " +
RSbuscaralumno!nombres_alumno
lblalumno.Item(3) = RSbuscarchurso!nombre_curso
lblalumno.Item(12) = RSbuscarchurso!seccion
lblalumno.Item(4) = RSbuscarmatricula!paralelo

If no_especialidad = "SI" Then
    lblalumno.Item(5).Enabled = True
    lblalumno.Item(5) = RSbuscarespecialidad!nombre_especialidad
Else
    lblalumno.Item(5).Enabled = False
End If
RSsaldo.Open "SELECT saldo FROM ALUMNOS WHERE
codigo_alumno= " & codigo_seleccionado & """, comm, adOpenStatic,
adLockOptimistic
lblalumno(0).Caption = RSsaldo!saldo
RSsaldo.Close
'Exit Sub
'Control:
End Sub

```

## VERIFICA LOS MESES QUE VA A CANCELAR

```
Private Sub pagar_Click()
    Set RSparametros = New Recordset
    Dim cantidad_disponible As Double
    Dim n_meses As Integer
    Dim cantidad_actual As Double
    Dim cantidad_guardada As Double
    If Text1.Text <> "" Then
        'guardar e imprimir
        Call generar_codigo_pension
        RSparametros.Open "SELECT
codigo_parametro,descripcion,valor,observacion FROM PARAMETROS
WHERE descripcion='Pensiones' ", comm, adOpenStatic,
adLockOptimistic
        valor_pension = RSparametros!valor
        If tipo_pension = "25%" Then
            subtotal = valor_pension / 4
        End If
        If tipo_pension = "50%" Then
            subtotal = valor_pension / 2
        End If
        If tipo_pension = "75%" Then
            subtotal = (valor_pension * 75) / 100
        End If
        If tipo_pension = "NO " Then
            subtotal = valor_pension
        End If
        cantidad_guardada = lblalumno.Item(0).Caption
        cantidad_disponible = cantidad_guardada + CDbl(Text1.Text)
        n_meses = Int(cantidad_disponible / subtotal)
        If n_meses + num_m > 10 Then
            n_meses = 10 - num_m
        End If
        If n_meses > 0 Then
            cantidad_actual = cantidad_disponible - (n_meses * subtotal)
            total_pagar = n_meses * subtotal
            lblalumno.Item(8).Caption = subtotal
            lblalumno.Item(9).Caption = total_pagar
            lblalumno.Item(7).Caption = Str(n_meses)
            lblalumno.Item(10).Caption = Date
            lblalumno.Item(11).Caption = total_pagar
            lblalumno.Item(2).Caption = "Pensiones"
            lblalumno.Item(13).Caption = cantidad_actual
            imprimir.Enabled = True
            menu.Enabled = False
            salir.Enabled = False
            cancelar.Enabled = True
        End If
    End If
End Sub
```

```

    Frame_pension.Enabled = False
    For i = num_m + 1 To 10 Step 1
    check_mes.Item(i).Value = 0
    Next
    If num_m + n_meses <= 10 Then
    For i = num_m + 1 To num_m + n_meses Step 1
    check_mes.Item(i).Value = 1
    Next
    End If
    Else
    lblalumno.Item(13).Caption = cantidad_disponible
    End If
    Else
    MsgBox "Ingrese la cantidad a depositar", vbInformation, "Control de
    Pensiones"
    Text1.SetFocus
    End If
End Sub

```

### **ALMACENA EL PAGO E IMPRIME EL RECIBO**

```

Private Sub imprimir_Click()
    Call registrar_pension
    Call imprimir_recibo_pensiones
    pagar.Enabled = False
    imprimir.Enabled = False
    Frame_pension.Enabled = False
    menu.Enabled = True
    salir.Enabled = True
    cancelar.Enabled = False
    nuevo.Enabled = True
    grilla.Visible = True
    Call buscar_datos
End Sub

```

#### **Guarda en la base de datos**

```

Private Sub registrar_pension()
'On Error GoTo error_añadir
'Nuevo registro
Set RSnuevo = New Recordset
Set RSSaldo = New Recordset
RSnuevo.Open "SELECT
codigo_pension,fecha_pago,lugar_pago,cantidad,numero_meses,codigo_alumno
FROM PENSIONES", comm, adOpenStatic, adLockOptimistic
If RSnuevo.RecordCount > 0 Then
RSnuevo.MoveFirst
End If

```

```

RSnuevo.AddNew
RSnuevo!codigo_pension = lblalumno.Item(6).Caption
RSnuevo!fecha_pago = lblalumno.Item(10).Caption
RSnuevo!lugar_pago = "UEGS"
RSnuevo!cantidad = lblalumno.Item(9).Caption
RSnuevo!numero_meses = lblalumno.Item(7).Caption
RSnuevo!codigo_alumno = codigo_seleccionado
'actualizar
RSnuevo.Update
RSnuevo.Requery
RSnuevo.MoveLast
RSsaldo.Open "SELECT saldo FROM ALUMNOS WHERE codigo_alumno= '"
& codigo_seleccionado & "'", comm, adOpenStatic, adLockOptimistic
RSsaldo!saldo = lblalumno.Item(13).Caption
MsgBox "DATOS GUARDADOS CORRECTAMENTE", vbInformation,
"BASE DE DATOS"
RSsaldo.Update
RSsaldo.Requery
RSsaldo.Close
RSnuevo.Close
'Exit Sub
'error_añadir:
' MsgBox "Error inesperado al buscar datos", vbInformation, "Base de Datos"
End Sub

```

### **Imprime los recibos**

```

Private Sub imprimir_recibo_pensiones()
    Dim i As Integer
    On Error GoTo errores:
    Printer.ScaleMode = vbCharacters 'establezco caracteres para controlar la
impresion
    Printer.TrackDefault = True
        'siempre apunta a la impresora predeter
        'el punto y coma (;) hace que la linea se imprima en el mismo renglon
        'para el logo y datos fijos
    i = 1
    For i = 1 To 9
        Printer.Print " "
    Next
    Printer.Font.Size = 5
    Printer.Print " "
    Printer.Font.Name = "Courier"
    Printer.Font.Size = 10
    Printer.FontBold = True
    Printer.DrawWidth = 1
    'APELLIDOS Y NOMBRES
    Printer.Print Tab(26); lblalumno.Item(1).Caption; Tab(97); Date
    Printer.Font.Size = 5

```

```

Printer.Print " "
Printer.Font.Size = 10
Printer.Print Tab(21); lblalumno.Item(12).Caption; Tab(76);
    lblalumno.Item(3).Caption
Printer.Font.Size = 5
Printer.Print " "
Printer.Font.Size = 10
Printer.Print Tab(28); lblalumno.Item(5).Caption + " " +
    lblalumno.Item(4).Caption
Printer.Print " "
Printer.Print " "
Printer.Print " "
Printer.FontBold = False
Printer.Font.Size = 5
Printer.Print " "
Printer.Font.Size = 10
'pensiones a pagar
If meses_ya_pagados = 0 Then
    Call meses_pagados_0
End If
If meses_ya_pagados = 1 Then
    Call meses_pagados_1
End If
If meses_ya_pagados = 2 Then
    Call meses_pagados_2
End If
If meses_ya_pagados = 3 Then
    Call meses_pagados_3
End If
If meses_ya_pagados = 4 Then
    Call meses_pagados_4
End If
If meses_ya_pagados = 5 Then
    Call meses_pagados_5
End If
If meses_ya_pagados = 6 Then
    Call meses_pagados_6
End If
If meses_ya_pagados = 7 Then
    Call meses_pagados_7
End If
If meses_ya_pagados = 8 Then
    Call meses_pagados_8
End If
If meses_ya_pagados = 9 Then
    Call meses_pagados_9
End If

```

```

Printer.EndDoc
errores:
If Err.Number = 482 Then MsgBox "Error de Impresora!"
End Sub

```

**Para la impresión de los meses pagados se ha seguido la siguiente estructura:**

```

Private Sub meses_pagados_8()
    If Int(lblalumno.Item(7).Caption) = 1 Then
        Printer.Print Tab(12); "1"; Tab(25); "Pensión Mayo"; Tab(93);
            lblalumno.Item(8).Caption; Tab(110); lblalumno.Item(8).Caption
        Printer.Print Tab(85); "-";
            i = 1
        For i = 1 To 10
            Printer.Print " "
        Next
        Printer.Print Tab(110); lblalumno.Item(11).Caption;
        Printer.Print Tab(110); "-";
        Printer.Print Tab(110); "-";
        Printer.Print Tab(110); lblalumno.Item(11).Caption;
    End If

    If Int(lblalumno.Item(7).Caption) = 2 Then
        Printer.Print Tab(12); "1"; Tab(25); "Pensión Mayo"; Tab(93);
            lblalumno.Item(8).Caption; Tab(110); lblalumno.Item(8).Caption
        Printer.Print Tab(12); "1"; Tab(25); "Pensión Junio"; Tab(93);
            lblalumno.Item(8).Caption; Tab(110); lblalumno.Item(8).Caption
        Printer.Print Tab(85); "-";
            i = 1
        For i = 1 To 9
            Printer.Print " "
        Next
        Printer.Print Tab(110); lblalumno.Item(11).Caption;
        Printer.Print Tab(110); "-";
        Printer.Print Tab(110); "-";
        Printer.Print Tab(110); lblalumno.Item(11).Caption;
    End If
End Sub

```

## Código del modulo notas

### RECUPERA LOS DATOS DEL ALUMNO Y SUS MATERIAS

```
Private Sub recuperar_datos_alumno()  
Set RSver1 = New Recordset  
Set RSver2 = New Recordset  
Set RSver3 = New Recordset  
Set RSver4 = New Recordset  
Set RSmaterias = New Recordset  
Dim año_actual As Integer  
Call vaciar_texto  
materias_si = "no"  
'Tabla Alumnos  
RSver1.Open "SELECT  
codigo_alumno,num_cedula_alumno,nombres_alumno,apellidos_alumno,fecha_n  
acimiento_alumno,edad_alumno,sexo_alumno,telefono_alumno,direccion_alumn  
o,nombre_padre,ocupacion_padre,direccion_padre,nombre_madre,ocupacion_ma  
dre,direccion_madre,becado_alumno,bautismo,primera_comunion,confirmacion,  
matrimonio,observaciones_sacramentos,pago_matricula,valor_matricula,esta_mat  
riculado,esta_inscrito,saldo from ALUMNOS WHERE codigo_alumno="" &  
codigo_seleccionado & """, comm, adOpenStatic, adLockOptimistic  
alumno(1).Text = RSver1!apellidos_alumno + " " + RSver1!nombres_alumno  
RSver1.Close  
'tabla Matriculas  
RSver2.Open "SELECT  
codigo_matricula,folio,tipo_matricula,periodo_lectivo,procedencia,paralelo,fecha  
_matricula,nombre_representante,telefono_representante,ocupacion_representante  
,direccion_representante,parentesco,observaciones,codigo_curso,codigo_alumno,c  
odigo_especialidad,codigo_matricula_final from MATRICULAS WHERE  
codigo_alumno="" & codigo_seleccionado & """, comm, adOpenStatic,  
adLockOptimistic  
alumno(4).Text = RSver2!paralelo  
codigo_matri = RSver2!codigo_matricula  
'curso  
RSver3.Open "SELECT nombre_curso FROM CURSOS WHERE  
codigo_curso="" & RSver2!codigo_curso & """, comm, adOpenStatic,  
adLockOptimistic  
alumno(5).Text = RSver2!codigo_curso  
alumno(2).Text = RSver3!nombre_curso  
If RSver2!codigo_curso = "C00011" Or RSver2!codigo_curso = "C00012" Or  
RSver2!codigo_curso = "C00013" Then  
RSver4.Open "SELECT nombre_especialidad FROM ESPECIALIDADES  
WHERE codigo_especialidad="" & RSver2!codigo_especialidad & """, comm,  
adOpenStatic, adLockOptimistic  
alumno(6).Text = RSver2!codigo_especialidad  
alumno(3).Text = RSver4!nombre_especialidad  
RSver4.Close  
'materias
```

```

    RSmaterias.Open "SELECT nombre_mater FROM AYUDA_MATERIAS
WHERE codigo_cur=" & RSver2!codigo_curso & " and codigo_especialidad="
& RSver2!codigo_especialidad & "", comm, adOpenStatic, adLockOptimistic
    If RSmaterias.RecordCount > 0 Then
        ' si tiene materias
        materias_si = "si"
        For i = 0 To RSmaterias.RecordCount - 1 Step 1
            Combo_materias.List(i) = RSmaterias!nombre_mater
            RSmaterias.MoveNext
        Next
        RSmaterias.MoveFirst
        Combo_materias.Text = RSmaterias!nombre_mater
        RSmaterias.Close
    End If
    Frame_BUSCAR.Visible = False
Else
    'materias
    RSmaterias.Open "SELECT nombre_mater FROM AYUDA_MATERIAS
WHERE codigo_cur=" & RSver2!codigo_curso & "", comm, adOpenStatic,
adLockOptimistic
    If RSmaterias.RecordCount > 0 Then
        ' si tiene materias
        materias_si = "si"
        For i = 0 To RSmaterias.RecordCount - 1 Step 1
            Combo_materias.List(i) = RSmaterias!nombre_mater
            RSmaterias.MoveNext
        Next
        RSmaterias.MoveFirst
        alumno(3).Text = "NINGUNA"
        Combo_materias.Text = RSmaterias!nombre_mater
        RSmaterias.Close
    End If
    Frame_BUSCAR.Visible = False
End If
RSver3.Close
RSver2.Close
End Sub

```

### **RECUPERA LAS NOTAS DE LA MATERIA YA SELECCIONADA**

```

Private Sub btn_notas_Click()
    Set RSmat = New Recordset
    btn_notas.Enabled = False
    Frame2.Enabled = True
    RSmat.Open "SELECT codigo_materia,nombre_materia FROM
MATERIAS WHERE nombre_materia=" & Combo_materias.Text & "
", comm, adOpenStatic, adLockOptimistic
    cod_mater = RSmat!codigo_materia

```

```

    RSmат.Close
    Combo_materias.Enabled = False
    Labelmat.Caption = Combo_materias.Text
    Call recuperar_notas
End Sub

```

```

Private Sub recuperar_notas()
    Set RSguardarnotas = New Recordset
    RSguardarnotas.Open "select
codigo_matricula,codigo_materia,nota1,nota2,nota3,asistencia1,asistencia
2,asistencia3,examen_supletorio,codigo_alumno,nota1_t1,nota2_t1,nota3_
t1,nota1_t2,nota2_t2,nota3_t2,nota1_t3,nota2_t3,nota3_t3,examen1,exame
n2,examen3,nota_final,aprobado,observacion_nota from NOTAS WHERE
codigo_matricula=" & codigo_matri & " and codigo_materia=" &
cod_mater & """, comm, adOpenStatic, adLockOptimistic
    If RSguardarnotas.RecordCount > 0 Then
        nota(1).Text = Trim(Str(RSguardarnotas!nota1_t1))
        nota(2).Text = Trim(Str(RSguardarnotas!nota2_t1))
        nota(3).Text = Trim(Str(RSguardarnotas!nota3_t1))
        nota(4).Text = Trim(Str(RSguardarnotas!examen1))
        nota(5).Text = Trim(Str(RSguardarnotas!nota1_t2))
        nota(6).Text = Trim(Str(RSguardarnotas!nota2_t2))
        nota(7).Text = Trim(Str(RSguardarnotas!nota3_t2))
        nota(8).Text = Trim(Str(RSguardarnotas!examen2))
        nota(9).Text = Trim(Str(RSguardarnotas!nota1_t3))
        nota(10).Text = Trim(Str(RSguardarnotas!nota2_t3))
        nota(11).Text = Trim(Str(RSguardarnotas!nota3_t3))
        nota(12).Text = Trim(Str(RSguardarnotas!examen3))
        nota(13).Text = Trim(Str(RSguardarnotas!asistencia1))
        nota(14).Text = Trim(Str(RSguardarnotas!asistencia2))
        nota(15).Text = Trim(Str(RSguardarnotas!asistencia3))
    End If
    RSguardarnotas.Close
End Sub

```

### **ALAMCENA LOS DATOS DE LAS NOTAS MODIFICADAS**

```

Private Sub guardar_Click()
    Dim not_valida As Integer
    Set RSguardarnotas = New Recordset
    not_valida = 0
    For i = 1 To 12 Step 1
        If nota(i).Text = "" Then
            nota(i).Text = "0"
        End If
        If Int(nota(i).Text) > 20 Then
            not_valida = 1
        End If
    Next

```

```

For i = 13 To 15 Step 1
    If nota(i).Text = "" Then
        nota(i).Text = "0"
    End If
    If Int(nota(i).Text) > 100 Then
        not_valida = 1
    End If
Next
If not_valida = 0 Then
    'Guardar datos
    RSGuardarnotas.Open "select
codigo_matricula,codigo_materia,nota1,nota2,nota3,asistencia1,asistencia
2,asistencia3,examen_supletorio,codigo_alumno,nota1_t1,nota2_t1,nota3_
t1,nota1_t2,nota2_t2,nota3_t2,nota1_t3,nota2_t3,nota3_t3,examen1,exame
n2,examen3,nota_final,aprobado,observacion_nota from NOTAS WHERE
codigo_materia=" & cod_mater & " and codigo_matricula=" &
codigo_matri & """, comm, adOpenStatic, adLockOptimistic
    If RSGuardarnotas.RecordCount > 0 Then
        RSGuardarnotas!nota1_t1 = Trim(nota(1).Text)
        RSGuardarnotas!nota2_t1 = Trim(nota(2).Text)
        RSGuardarnotas!nota3_t1 = Trim(nota(3).Text)
        RSGuardarnotas!examen1 = Trim(nota(4).Text)
        RSGuardarnotas!nota1_t2 = Trim(nota(5).Text)
        RSGuardarnotas!nota2_t2 = Trim(nota(6).Text)
        RSGuardarnotas!nota3_t2 = Trim(nota(7).Text)
        RSGuardarnotas!examen2 = Trim(nota(8).Text)
        RSGuardarnotas!nota1_t3 = Trim(nota(9).Text)
        RSGuardarnotas!nota2_t3 = Trim(nota(10).Text)
        RSGuardarnotas!nota3_t3 = Trim(nota(11).Text)
        RSGuardarnotas!examen3 = Trim(nota(12).Text)
        RSGuardarnotas!asistencia1 = Trim(nota(13).Text)
        RSGuardarnotas!asistencia2 = Trim(nota(14).Text)
        RSGuardarnotas!asistencia3 = Trim(nota(15).Text)
    End If
    RSGuardarnotas.Update
    RSGuardarnotas.Requery
    RSGuardarnotas.Close
    MsgBox "DATOS GUARDADOS CORRECTAMENTE",
vbInformation, "BASE DE DATOS"
    Call vaciar_notas
    Call vaciar_texto
    Labelmat.Caption = ""
    Frame_BUSCAR.Visible = True
Else
    MsgBox "Una de las notas ingresadas no es válida, Ingrese
correctamente!", vbInformation, "Notas mal Ingresadas"
End If
End Sub

```

## **6.4 Pruebas**

Para la realización de pruebas se ha tomado en consideración dos formas de realizarlas.

La primera que se ha realizado con datos simulados los cuales permitieron obtener un informe de errores y corregirlos a tiempo. Estas pruebas se fueron realizando de acuerdo a la culminación de cada modulo, con la supervisión del tutor de pasantía.

La segunda se realizo en la culminación del desarrollo del proyecto, se utilizaron datos reales, que constan dentro de la información que posee la institución. Esta prueba se realizo en compañía de las personas que van a utilizar el Sistema permitiendo realizar ciertos cambios estéticos, corrección de errores y comprobando el correcto funcionamiento de validaciones, funciones, procedimientos y demás componentes utilizados para facilitar el uso del Sistema.

## **CAPITULO VII**

### **IMPLANTACIÓN**

#### **7.3 Puesta en Marcha**

Para la creación de la base de datos se ha creado un archivo (.mdf) y (.ldf) que contiene un esquema con todas las tablas, reglas y relaciones. La base de datos creada para que el sistema funcione se encuentra guardada en el CD de instalación del sistema, y los pasos para su instalación se encuentran dentro del manual del administrador.

Para la instalación y ejecución del sistema estudiantil se deben seguir los pasos que se encuentran en el manual de instalación.

Solo existen dos equipos para la instalación de este sistema, el primero se esta como servidor y a la vez como cliente, y el otro equipo esta solo instalado como cliente.

La puesta en marcha del sistema de control de matriculas, pensiones y notas, se ha ido realizando paulatinamente y se culminara al final del año lectivo. Ya que se es un sistema que tiene varios módulos y los datos que se ingresan en cada uno dependen del tiempo por ejemplo las notas y pago de pensiones de los estudiantes.

Se inicio con el uso del modulo de matriculas y pago de pensiones, pero para efectos de comprobación y hasta que exista una familiarización con el sistema, durante este año tanto la matriculación, pago de pensiones y control de notas se realizará de forma manual como hasta ahora se ha venido llevando y también con el sistema.

Nota.- El manual de usuario, del administrador y de instalación se encuentra en este documento como Anexo.

## 7.4 Capacitación

**OBJETIVO.-** Preparar a todos los usuarios que van utilizar el sistema estudiantil, para que la manejo de este nuevo sistema sea fácil, agradable y satisfactorio.

**DIRIGIDO A.-** Ya que el sistema fue desarrollado para satisfacer las necesidades de la institución la capacitación esta dirigida al personal administrativo y profesores que laboran en la institución, es decir a las personas que van a utilizar el sistema estudiantil desarrollado.

### CRONOGRAMA

Cada uno de los temas planteados está orientado a la explicación de la forma de uso de los diferentes módulos existentes, así como también la generación correcta de reportes.

<b>Tema</b>	<b>Duración</b>	<b>Responsable</b>	<b>Dirigido a</b>
Presentación	½ hora	Gladys Martínez	Personal Administrativo
Ingreso al Sistema y Gestión de Permisos	1 hora	Gladys Martínez	Administrador
Manejo de Matriculas	1 hora	Gladys Martínez	Secretaría
Manejo de Pensiones	½ hora	Gladys Martínez	Colecturía
Manejo de Notas	1 hora	Gladys Martínez	Maestros
Agradecimiento	½ hora	Gladys Martínez	Personal de la institución

## CAPITULO VIII

### CONCLUSIONES Y RECOMENDACIONES

#### 9.1 Conclusiones

- El sistema estudiantil realizado reduce tiempo en cantidad considerable tanto para el personal administrativo como para los alumnos, estudiantes y profesores, generando a la vez un ahorro económico.
- Al trabajar con bases de datos, se garantiza la integridad de los datos dando así mayor confianza a la hora de obtener consultas e informes.
- Las herramientas de desarrollo elegidas fueron apropiadas para el sistema estudiantil realizado en la UEGS por su rapidez y confiabilidad.
- La correcta utilización de los recursos tecnológicos en este proyecto informático permitieron asegurar un buen funcionamiento de la aplicación.
- El proyecto propuesto cumple con los requerimientos que la institución dispuso, mejorando así la agilidad en los procesos que se ejecutan.

## 9.2 Recomendaciones

- Se recomienda que se facilite el manual de usuario a todos los empleados que requieren usar el sistema, para facilitar el aprendizaje de su funcionamiento.
- Utilizar una impresora matricial, para reducir los costos impresión.
- Capacitar al personal administrativo con un curso intensivo de computación para que el personal tenga conocimientos básicos y con ello pueda manejar el sistema de manera apropiada.
- Contar con una persona capacitada en la administración de la base de datos para inicio y mantenimiento de la base de datos.
- Capacitar de forma apropiada a todos los maestros que van a manipular las notas de los estudiantes para obtener un buen funcionamiento del sistema.
- Habilitar dos maquinas exclusivamente para las matriculas y pensiones que se realizan en la institución.
- Tener siempre instalada una impresora en la maquina donde se ejecuta el sistema para imprimir los reportes.
- Realizar respaldos de información periódicamente de manera que en caso de presentarse incoherencia en los datos estos pueden ser recuperados, garantizando de esta manera que los mismos estén siempre disponibles.
- Crear políticas de seguridad y asegurar correctamente los permisos a los usuarios que van a manejar la aplicación para mantener la integridad de los datos.
- Consultar la ayuda general con la que cuenta el Sistema, para solucionar posibles problemas que se pueden presentar.

## **BIBLIOGRAFÍA**

### **Libros**

- KROENKE, David    Procesamiento De Bases De Datos  
Octava Edición
- PRESSMAN, Roger   Ingeniería de Software  
Tercera Edición
- SILBERSCHATZ, Abrahán; KART , Henry; SUDARSHAN  
Fundamentos De Base De Datos, Cuarta Edición
- SONN, James A.     Análisis y Diseño de Sistemas de Información  
Edición 2<sup>da</sup> 1997

### **Internet**

<http://es.wikipedia.org/wiki/Inform%C3%A1tica>  
<http://www.monografias.com/trabajos/seguinfo/seguinfo.shtml>  
<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>  
<http://sopa.dis.ulpgc.es/diplomatura/practicass9798/ipc/interc02.htm>  
<http://www.maestrosdelweb.com/editorial/phpmysqlap/>  
<http://www.webtaller.com/manual-mysql/creando-usando-base-datos.php>  
[http://es.wikipedia.org/wiki/Visual\\_Basic](http://es.wikipedia.org/wiki/Visual_Basic)  
<http://www.canalvisualbasic.net/manual/tema1.asp>  
<http://www.monografias.com/trabajos10/visual/visual.shtml>  
<http://www.monografias.com/trabajos2/guiavb/guiavb.shtml>  
[http://es.wikipedia.org/wiki/SQL\\_Server](http://es.wikipedia.org/wiki/SQL_Server)  
[http://www.mundotutoriales.com/tutoriales\\_sql\\_server-mdtema119.htm](http://www.mundotutoriales.com/tutoriales_sql_server-mdtema119.htm)  
<http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>

## GLOSARIO

**Base de Datos.-** Es un almacenamiento de datos formalmente definidos y controlados centralmente para intentar servir a múltiples y diferentes aplicaciones. Una Base de Datos no es solamente una colección de archivos, es una fuente de datos significativos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones.

**Cliente.-** Los clientes son las máquinas o procesos que piden información, recursos y servicios a un servidor.

**Formulario.-** Los formularios son las zonas de la pantalla sobre las que se diseña el programa y sobre las que se sitúan los controles o herramientas.

**Interfaz.-** La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador.

**Método.-** Los métodos son funciones que realizan ciertas tareas y que son llamadas desde un programa.

**Proceso.-** Capacidad del Sistema para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida.

**Sistema.-** Es una parte del mundo real a la cual una persona o grupo de personas durante un cierto intervalo de tiempo o por alguna razón, lo escoge para enfocar como un todo formado por componentes.

**Sistemas de información.-** Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

**Servidor.-** Los servidores son los procesos que proporcionan información recursos y servicios a los clientes.

**Variable.-** Una variable es un nombre que designa a una zona de memoria (se trata por tanto de un identificador), que contiene un valor de un tipo de información.

**Visual Basic.-** es un lenguaje de programación visual, también llamado lenguaje de cuarta generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

# **ANEXOS**