UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

"Implementación de la Estructura del Sistema Académico con el Módulo de Matriculación para la Facultad de Ciencias Matemáticas y Físicas"

TESIS DE GRADO

Previo la Obtención del Título de:
INGENIERO EN SISTEMAS COMPUTACIONALES

Autores:

AYUQUINA CARABAJO MARTHA ELIZABETH
ARELLANO GUZMAN GREGORIO ALEJANDRO
CRUZ OCHOA WILSON
ZAMBRANO MOLINA DARWIN FLORENCIO

GUAYAQUIL - ECUADOR

Año: 2004

AGRADECIMIENTO

- A Dios por permanecer a cada instante a nuestro lado y guiar los pasos que he marcado en este tiempo.
- A nuestros padres por ser nuestro apoyo incondicional y porque con sus sabios consejos hemos logrado superarnos.
- A nuestros hermanos, de quienes hemos aprendido la importancia de tener una familia y de disfrutar ese privilegio.
- A la Universidad de Guayaquil, un gran templo de enseñanza, nos ha dado la oportunidad de sentirnos partícipes de sus reconocimientos.
- A los docentes de la Carrera de Ingeniería de Sistemas, por compartir sus conocimientos y prepararnos profesionalmente.
- A nuestros amigos, por estar juntos en los buenos y malos momentos y porque han levantado nuestro optimismo en tiempos de dificultad.
- A ti que estás leyendo nuestro agradecimiento y especialmente nuestra tesis.
- A los que conocimos, a los que conocemos y a los que conoceremos, porque de cada persona aprendemos algo nuevo. Muchas Gracias.

DEDICATORIA

A nuestros padres por permanecer siempre a nuestro lado y por cumplir uno de sus sueños: Ser profesionales.

A Haydee Guzmán por constituir el pilar de fortaleza y apoyo en nuestro integrante de grupo Gregorio.

A la memoria de José Ayuquina, hombre sencillo, respetuoso y respetable y de cuyas enseñanzas recordamos "El éxito se logra confiando en uno mismo y sin perder la humildad que han adquirido"

DECLARACION EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente al suscrito(s), perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma”

(Reglamento de Graduación de la Carrera de Ingeniería en Sistemas Computacionales, Art. 26)

INDICE GENERAL

Pág.

AGRADECIMIENTO II

DEDICATORIA III

DECLARACION EXPRESA IV

INDICE GENERAL V

INTRODUCCION 1

CAPITULO 1

1. ENFOQUE DEL PROYECTO 6

1.1. Visión 6

1.2. Misión 6

1.3. Objetivos 7

1.4. Metodología 9

CAPITULO 2

2. FASE DE ANALISIS 12

2.1. Escenario Actual 12

2.1.1. Recopilación de la Información 12

2.1.2. Definición de los procesos por área en Ingeniería Civil y Sistemas 15

2.1.3. Proceso de inscripción al curso Preuniversitario o Propedéutico 17

2.1.4. Análisis en la Carrera de Ingeniería Civil 18

2.1.5. Análisis en la Carrera de Ingeniería de Sistemas 33

2.1.6. Observaciones generales de los procesos en las unidades académicas 48

2.1.7. Identificación de necesidades 50

2.2. Escenario Futuro Propuesto 51

2.2.1. Características del Sistema 51

2.2.2. Definición de los módulos y procesos del sistema 53

2.2.3. Proceso de inscripción y matriculación en Ingeniería Civil 58

2.2.4. Proceso de inscripción y matriculación en Ingeniería de Sistemas Computacionales 60

2.2.5. Funciones recomendadas para los usuarios 62

2.2.6. Requerimientos de software y hardware para la etapa de desarrollo 63

2.2.7. Explicación de la Plataforma Tecnológica 64

CAPITULO 3

3. ETAPA DE DISEÑO 69

3.1. Estándares para el diseño de la Base de Datos 69
3.1.1. Explicación de la Plataforma Tecnológica 69
3.1.2. Formato del nombre de las tablas 69
3.1.3. Formato del nombre de los campos 70
3.2. Módulos del Sistema de Matriculación 71
3.3. Diseño de la Estructura del Sistema Académico 73
3.4. Estándares en el diseño de las pantallas 82
3.4.1. Ubicación de los controles en la forma 82
3.4.2. Propiedades Generales 84
3.4.3. Formas 84
3.4.4. Botones 85
3.4.5. TextBoxes 87
3.4.6. ComboBoxes 88
3.4.7. OptionButtons 89
3.4.8. CheckBoxes 89
3.4.9. GroupBoxes o Frames 89
3.4.10. Datagrid 89
3.4.11. MaskedBox 90
3.4.12. Labels 90

CONCLUSIONES 91
RECOMENDACIONES 92
GLOSARIO 94

ANEXOS 96

BIBLIOGRAFIA 99

INDICE DE FIGURAS

Pág.

Figura 2.1. Inscripción y Matriculación en la Carrera de Civil para estudiantes de primer año 24

Figura 2.2. Inscripción y Matriculación en la Carrera de Civil para estudiantes no principiantes 28

Figura 2.3. Sistema de Evaluación en la Carrera de Civil 31

Figura 2.4. Inscripción y Matriculación en la Carrera de Sistemas para estudiantes de primer año 40

Figura 2.5. Inscripción y Matriculación en la Carrera de Sistemas para estudiantes no principiantes 45

Figura 2.6. Sistema de Evaluación de la Carrera de Sistemas 46

INDICE DE TABLAS

Pág.

Tabla 3.1. Propiedades Generales 84

Tabla 3.2. Propiedades de la Forma 84

Tabla 3.3. Propiedades de los botones 86

Tabla 3.4. Propiedades de los Textboxes 87

INTRODUCCION

La Universidad de Guayaquil constituye la primera universidad en el Ecuador. Se define como una persona jurídica de derecho público que se rige por la Constitución, la Ley de Universidades y Escuelas Politécnicas y los estudios propios, en términos más simples, es labor de la comunidad y corresponde a sus intereses, proporcionando pedagogía, información y servicios.

Constantemente la entidad instaura grandes reformas y retos para una mayor recepción de personas. Entre ellos mencionamos:

- ✓ Modernas Herramientas Tecnológicas
- ✓ Enseñanza integral
- ✓ Motivación a los estudiantes sobresalientes mediante Becas, Exoneraciones, etc.
- ✓ Estudios a distancia
- ✓ Cursos extracurriculares

En la actualidad está compuesta por 17 facultades y 66 carreras, de las cuales el 45% tiene sus procesos totalmente automatizados y la diferencia todavía depende de ciertas inspecciones manuales.

Por tal motivo la Universidad está instituyendo carreras técnicas de manera que los profesionales formen parte del desarrollo tecnológico de la entidad y proporcionen ideas innovadoras que perfeccionen la educación en las unidades académicas, haciendo de cada una de ellas un ente investigativo y productivo que participe activamente en los cambios de la estructura del país.

Cada desafío que afronta la institución para pernoctar en la mente de las personas requiere que su gente esté predispuesta a su trabajo y sobretodo a un excelente conocimiento en los procesos internos otorgando de esta manera ligereza y habilidad al interactuar con el factor humano principal: “el educando”.

Durante el transcurso de nuestra carrera tuvimos la oportunidad de evidenciar estas características que distinguen al organismo y además el tiempo suficiente para analizar las necesidades primordiales en algunas facultades que harían de nuestra universidad una entidad admirable, como por ejemplo:

- Optimización del Sistema Académico
- Migración de las aplicaciones a las plataformas educativas que adquieren para la enseñanza de los estudiantes
- Atención de nuevos requerimientos en el Sistema de Recursos Humanos
- Control de Inventario

Dichas investigaciones ocasionaron un gran interés de nuestra parte: “Implementar un Sistema Académico integrado para la Universidad” debido a dos factores importantes:

1. La mayoría de los procesos académicos son automatizados e independientes entre facultad.
2. Control manual de datos de alumnos en el 10% de las facultades vigentes.

Así por ejemplo, desde que la Facultad de Ciencias Matemáticas y Físicas fue creada para los jóvenes interesados en la Carrera de Ingeniería Civil, sin duda alguna la envergadura más notable de la misma en la actualidad se constituye por el Sistema de Matriculación que ha permanecido intacto en 20 años con características desfavorables tales como:

- El procesamiento de información en un 80% es manual
- Lentitud en trámites de obtención de certificados, promociones, promedios u otro documento que proviene de la Secretaría de la Facultad
- Sobrecarga de trabajo al personal de Secretaría
- Falta de información básica y oportuna para los estudiantes

Los factores que nos obligan como parte de la Carrera de Ingeniería de Sistemas a responder con obras nuestros conocimientos son:

1. El avance tecnológico
2. Y la naturaleza humana que exige más perfeccionamiento y desafío

Para conceder un sistema automatizado con grandes beneficios, de manera que las inscripciones futuras se realicen eficientemente, descargando el trabajo del personal y mejorando la seguridad de la información deberían considerar las recomendaciones evidentes, como:

- Estudio exhaustivo de las necesidades apremiantes en la parte académica con el objetivo de entregar recomendaciones para la mejora del proceso actual.
- Bosquejo de los aspectos afines en el proceso académico de cada Facultad.
- Diseño de un sistema estándar y que se adecue a los requisitos de futuros procesos.
- Disminuir costos al no adquirir un sistema académico sino más bien motivar a los profesionales de la carrera a desarrollar e implementar las necesidades que tiene la Universidad para disponer de un Sistema de Información.

Nuestra meta es alcanzar prestigio universitario a nivel nacional, destacando la capacidad competitiva e intelectual adquirida en nuestra carrera y

prescindiendo de las dificultades que en un futuro puedan impedir el progreso de la Universidad.

CAPITULO 1

1. ENFOQUE DEL PROYECTO

1.1. Visión

Contribuir con la Universidad y la Facultad de Ciencias Matemáticas y Físicas en su crecimiento tecnológico a fin de cubrir con las necesidades actuales y futuras.

1.2. Misión

Implementación de un sistema académico estándar, confiable, seguro y flexible a los procesos internos que manejan cada una de las unidades académicas de la Universidad de Guayaquil considerando la implementación práctica en la Facultad de Ciencias Matemáticas y Físicas a fin de obtener la satisfacción total del personal administrativo como estudiantil.

1.3. Objetivos

1.3.1. Conocer la estructura académica de la Universidad de Guayaquil.

1.3.1.1. Constatar el número de Facultades y Carreras existentes en la Universidad de Guayaquil.

1.3.1.2. Reconocer las necesidades comunes de las facultades.

1.3.2. Evidenciar la utilización de Sistemas Académicos Automatizados.

1.3.2.1. Consultar las carreras que utilizan actualmente un sistema académico automatizado.

1.3.2.2. Realizar un estudio macro sobre los sistemas académicos automatizados y determinar si estos cubren las necesidades actuales de cada facultad de la Universidad de Guayaquil.

1.3.2.3. Obtener datos estadísticos de las carreras con procesos manuales y automáticos.

1.3.3. Estudiar el reglamento universitario actual para la matriculación de estudiantes.

1.3.3.1. Analizar el proceso de matriculación de las Facultades.

1.3.3.2. Determinar cuáles son los procedimientos comunes entre las Facultades para el proceso de Matriculación.

1.3.3.3. Aplicar las técnicas de investigación: Entrevistas, encuestas, observaciones para evaluar los procesos académicos involucrados.

1.3.4. Elaborar estándares para el proceso académico de la Universidad.

1.3.4.1. Optimizar el sistema académico en cada una de las unidades académicas.

1.3.4.2. Diseñar estructura estándar de Base de Datos para el Sistema Académico.

1.3.4.3. Dar a conocer la solución o el conjunto de soluciones para un sistema académico integrado.

1.3.5. Otorgar mayor comodidad, seguridad y confianza en el manejo de datos académicos a nivel universitario.

1.3.5.1. Facilitar las gestiones en los departamentos involucrados en el proceso.

1.3.5.2. Conseguir satisfacción por parte de los usuarios al interactuar con aplicaciones automatizadas.

1.3.6. Implementar el desarrollo del proyecto basado en los conocimientos adquiridos en la Universidad.

1.3.6.1. Incentivar al educando en el aprendizaje de la Carrera de Ingeniería en Sistemas.

1.3.6.2. Proyectar la competitividad existente en la Carrera de Ingeniería de Sistemas Computacionales a beneficio del crecimiento productivo nacional.

1.4. Metodología

Nuestros objetivos propuestos se cumplirán con la implementación del Sistema Académico en la Facultad de Ciencias Matemáticas y Físicas. Dicha implementación comprende las siguientes etapas:

1. Análisis de los procesos involucrados

2. Diseño de datos

3. Desarrollo del sistema

4. Implementación

Estos pasos son necesarios para la optimización de los procedimientos actuales. Considerando para las fases ya mencionadas las actividades que detallamos a continuación:

FASE I.- Análisis de los procesos involucrados

Levantamiento y estudio de la información

1. Coordinación de entrevistas con personal operativo, administrativo y autoridades de cada Facultad.
2. Entrevistas con autoridades de la Facultad: Decanos y Directores
3. Entrevistas con personal administrativo: Secretarios y Coordinadores
4. Entrevistas con personal operativo: Secretarias
5. Ordenamiento y clasificación de la información relevada.
6. Informe sobre las necesidades para la automatización del proceso académico.

FASE II.- Diseño de datos

1. Diseño preliminar del sistema.
2. Revisión del diseño preliminar
3. Elaboración del Modelo Lógico de Datos
4. Diseño Físico de los archivos y base de datos
5. Diseño de entradas y salidas de datos
6. Evaluación del Software
7. Evaluación del Hardware
8. Entrega del informe conteniendo los diseños y propuestas de Hardware y Software

FASE III.- Desarrollo del sistema

1. Creación de pantallas y reportes
2. Pruebas individuales (por módulo)
3. Prueba global del sistema con el director de tesis

FASE IV.- Implementación del sistema

1. Implementación inicial del sistema. Simulación
2. Depuración del sistema
3. Implementación y puesta en marcha del sistema

CAPITULO 2

2. FASE DE ANALISIS

2.1. Escenario Actual

2.1.1. Recopilación de la Información

Se han realizado entrevistas al personal administrativo de ambas carreras quienes aportaron con sus conocimientos sobre el proceso académico y mencionamos a continuación:

- Ing. Fernando Abad Montero: Decano de la Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería Civil:

- Ab. Stalin Ramírez: Secretario de la Carrera
- Sra. Gladys García Bueno: Coordinadora Académica
- Ing. Alex Jordán: Coordinador de los Cursos Extracurriculares
- Sra. Eleana: Secretaria

- Sra. Mónica Bazán: Secretaria

Carrera de Ingeniería de Sistemas Computacionales:

- Ing. Matilde Alvear: Directora de la Carrera
- Ab. Juan Chávez: Secretario de la Carrera

Centro de Cómputo de la Universidad de Guayaquil:

- Sra. Ketty Sar: Jefe de Sección Académica

Presentamos una propuesta al Ing. Abad Montero sobre la automatización del actual sistema académico en la Facultad de Ciencias Matemáticas y Físicas; luego de su aprobación, el ingeniero envió una autorización al personal administrativo para que colaboren con nosotros.

El Ab. Ramírez Chávez recibió la orden directa del Decano y delegó a todo el personal a su cargo para que aporten en la recopilación de información requerida. Además nos facilitó el documento “Valores y Tasas de Matriculación para las carreras no autofinanciadas”. (**Ver Anexo 2.1**)

Recibimos una explicación por parte de la Sra. García Bueno sobre el control de los cupos por grupo y el cruce de horarios por materia.

El Ing. Jordán fue quien nos motivó a emprender este proyecto comunicándonos las necesidades de la Facultad. Además, nos indicó cómo se maneja el proceso académico actual en la Carrera de Ingeniería Civil.

La Sra. Eleana nos proporcionó el modelo de la ficha personal de los estudiantes y los formularios para la inscripción de los estudiantes a la carrera. (**Ver Anexo 2.2**)

Con respecto a las órdenes de pago, la Sra. Bazán nos explicó su emisión mediante el programa de Recaudaciones realizado en FoxPro por el Centro de Cómputo de la Universidad de Guayaquil.

La Ing. Alvear hizo entrega del cuadro de las tasas y valores para la matriculación de carreras autofinanciadas. (**Ver Anexo 2.3**) Además explicó cómo calcular el monto de la matrícula por materia.

El Ab. Chávez Atocha explicó en forma general el proceso de calificaciones; aprobación de materia e información de los cursos extracurriculares.

La Sra. Ketty Sar contribuyó con la información del proceso académico de la Universidad, así como el procedimiento de recaudación de la universidad y la obtención de la tabla de “Rebajas y exoneraciones”.

2.1.2. Definición de los procesos por área en Ingeniería Civil y Sistemas

Para una mayor comprensión clasificamos los procesos involucrados en el sistema académico actual en los siguientes módulos:

Proceso de Matriculación

Se lleva a cabo al inicio de cada ciclo lectivo:

- Determinación de las categorías por colegio
- Cálculo del valor de la matrícula dependiendo de los aranceles y descuentos aprobados para el estudiante
- División del valor de la matrícula en cuotas
- Emisión de órdenes de pago
- Emisión de listas

- Anulación de la matrícula

Plan de Estudios

Planteamiento relevante de la malla curricular* para obtener el título que la unidad académica otorga. Las actualizaciones del Pénsum son esporádicas:

- Asignación de materias a un nivel
- Aprobación de Prerrequisitos para cursar una materia

Mantenimiento de datos personales de los estudiantes

Utilizan fichas para anotar la hoja de vida de cada alumno. Una ficha es un documento preimpreso con los campos personales y educativos del alumno. El registro de datos personales se actualiza al inicio de cada ciclo lectivo.

Proceso de Inscripción

El proceso se efectúa al inicio de cada periodo:

- Selección de materias por los alumnos
- Verificación de los promedios del nivel anterior
- Aprobación de las Materias a cursar de los alumnos
- Aviso de cruce de horarios
- Informe de cursos curriculares y extracurriculares

- Asignación de materias por grupo a los catedráticos

Registro de calificaciones

Las calificaciones se asientan en la ficha de cada estudiante y en cada parcial:

- Ingreso de notas
- Cálculo del promedio
- Consultas de notas
- Solicitud de recalificaciones
- Emisión de certificados de notas

2.1.3. Proceso de inscripción al curso Preuniversitario o Propedéutico

Detallamos los pasos que se desarrollan en el proceso de inscripción al Preuniversitario o Propedéutico en cualquier entidad académica:

1. El estudiante se acerca al Departamento de Recaudaciones para adquirir una especie valorada.
2. El estudiante elabora una solicitud para la inscripción al preuniversitario dirigida al Decano o Director de la entidad académica donde estudiará. La solicitud se realiza en la especie valorada.

3. El estudiante debe acudir al Departamento de Secretaría de la unidad académica presentando los siguientes documentos:

a. Fotocopia de la cédula de identidad (anverso y reverso) para estudiantes con nacionalidad ecuatoriana o pasaporte si se trata de persona extranjera.

b. Fotocopia del título de Bachiller o certificado de estar cursando el sexto año. En el caso de profesionales se solicita una fotocopia del título universitario.

c. Solicitud de ingreso en especie valorada donde el estudiante llena los datos específicos requeridos por la unidad.

2.1.4. Análisis en la Carrera de Ingeniería Civil

2.1.4.1. Proceso de inscripción y matriculación

a) Estudiantes nuevos (Primera vez a Primer año)

1. El estudiante una vez aprobado el preuniversitario se presenta al Departamento de Bienestar Estudiantil de la Universidad de Guayaquil para efectuarse los exámenes médicos, odontológicos y sicológicos previo el pago de una tasa de salud equivalente al 10% del salario mínimo vital. Cabe recalcar que este es un requisito obligatorio para toda persona que ingrese por primera vez a una unidad académica de la Universidad de Guayaquil.

2. Después de dichos exámenes el Departamento de Bienestar Estudiantil otorga en base a los resultados, un certificado al estudiante indicando idoneidad o incapacidad.
3. El estudiante se acerca al Departamento de Recaudaciones para adquirir una especie valorada.
4. El estudiante elabora una solicitud -en la especie valorada- para la inscripción a primer año siguiendo un modelo dado por la unidad académica donde desea estudiar.
5. El estudiante entrega a la secretaria de la Carrera los siguientes documentos:
   - Una foto tamaño carnet actualizada
   - Una fotocopia de la cédula o pasaporte
   - Solicitud de inscripción
   - 500 hojas tamaño A4
6. La secretaria recepta los documentos que se encuentran completos y en orden, e indica al estudiante que debe presentarse después de una semana.
7. Transcurrido ese tiempo, la secretaria entrega un formulario con dos copias del mismo. El documento a llenar por el estudiante contiene:
   a. Datos personales

      Apellidos y Nombres

Lugar de Nacimiento (Ciudad y Provincia)

Fecha de Nacimiento

Cédula de Ciudadanía

Título de Grado

Colegio en donde se graduó (indicando si es Nacional, Particular u otros)

Provincia del colegio

Fecha de Graduación

Nota de Grado

Nombre del Padre

Nombre de la Madre

Domicilio

Teléfono del domicilio

Lugar de Trabajo

Teléfono del trabajo

b. Cuadro de las materias a cursar. El reglamento universitario indica que está obligado a cursar todas las materias de primer nivel.

c. Número de Suspensos y Arrastres

8. El estudiante está obligado a llenar y entregar a la secretaria, los tres documentos descritos en el paso anterior. En el cuadro de banco de materias, el estudiante

selecciona todas las materias no los grupos. Esto se conoce como preinscripción.

9. La secretaria asienta estos datos en una ficha personal y adjunta una de las tres hojas.

Cada estudiante tiene una ficha personal única que describe sus datos personales y las materias que el alumno cursa según su decisión particular, y la aprobación de materias en niveles anteriores y/o el tema de tesis, director, su nota de grado y la fecha de incorporación.

Las fichas se llenan manualmente y se las guarda en un archivador con acceso sólo para el personal de Secretaría.

10. De acuerdo al número de materias a cursar y de la categoría del colegio de donde proviene, se calcula el valor de la matrícula considerando intereses o descuentos que adquiera el estudiante de acuerdo al Reglamento de Matriculación para las Carreras no autofinanciadas.

11. Si el estudiante solicita crédito, se divide el monto de la matrícula de acuerdo al número de cuotas que las autoridades hayan aprobado.

12. En ambos casos, es decir, de existir o no crédito, se emiten las órdenes de pago, las mismas que serán firmadas por el abogado de la entidad académica.

13. La secretaria envía la información de las matrículas al Centro de Cómputo de la Universidad de Guayaquil, siendo este departamento el encargado de entregar la información al Banco y al Departamento de Recaudaciones.

14. El estudiante retira las órdenes de pago después de una semana de la preinscripción. Adjunto a las órdenes de pago están los dos formularios.

15. El estudiante debe esperar 48 horas para cancelar la primera orden de pago, en caso de no hacerlo, la secretaria procede a la anulación de la matrícula, por ende no constará en lista. Se exceptúa la anulación de la matrícula previa aprobación de las autoridades de la Carrera.

16. El estudiante entrega a la secretaria, la primera orden de pago con tres fotocopias de la orden y anexa los dos formularios.

17. La secretaria archiva las copias de la orden de pago devolviendo la original al estudiante y entrega la información al Centro de Cómputo de la Universidad.

18. El Banco y el Departamento de Recaudaciones entregan la información del cobro de matrícula al Centro de Cómputo de la Universidad para legalizar su permanencia como alumno de la Carrera.

19. La secretaria asigna un código único al estudiante.

20. La secretaria establece al estudiante un grupo por cada materia a cursar.

21. La secretaria entrega al estudiante, un documento que especifica las materias y los grupos seleccionados junto con el nombre de los docentes.

Para una mayor comprensión, observar el gráfico de este proceso. (**Ver Figura 2.1**)

b) Estudiantes no principiantes

1. El estudiante se acerca al Departamento de Recaudaciones para adquirir una especie valorada.

2. El estudiante elabora una solicitud -en la especie valorada- para la inscripción a un nivel siguiendo un modelo proporcionado por la unidad académica donde desea estudiar.

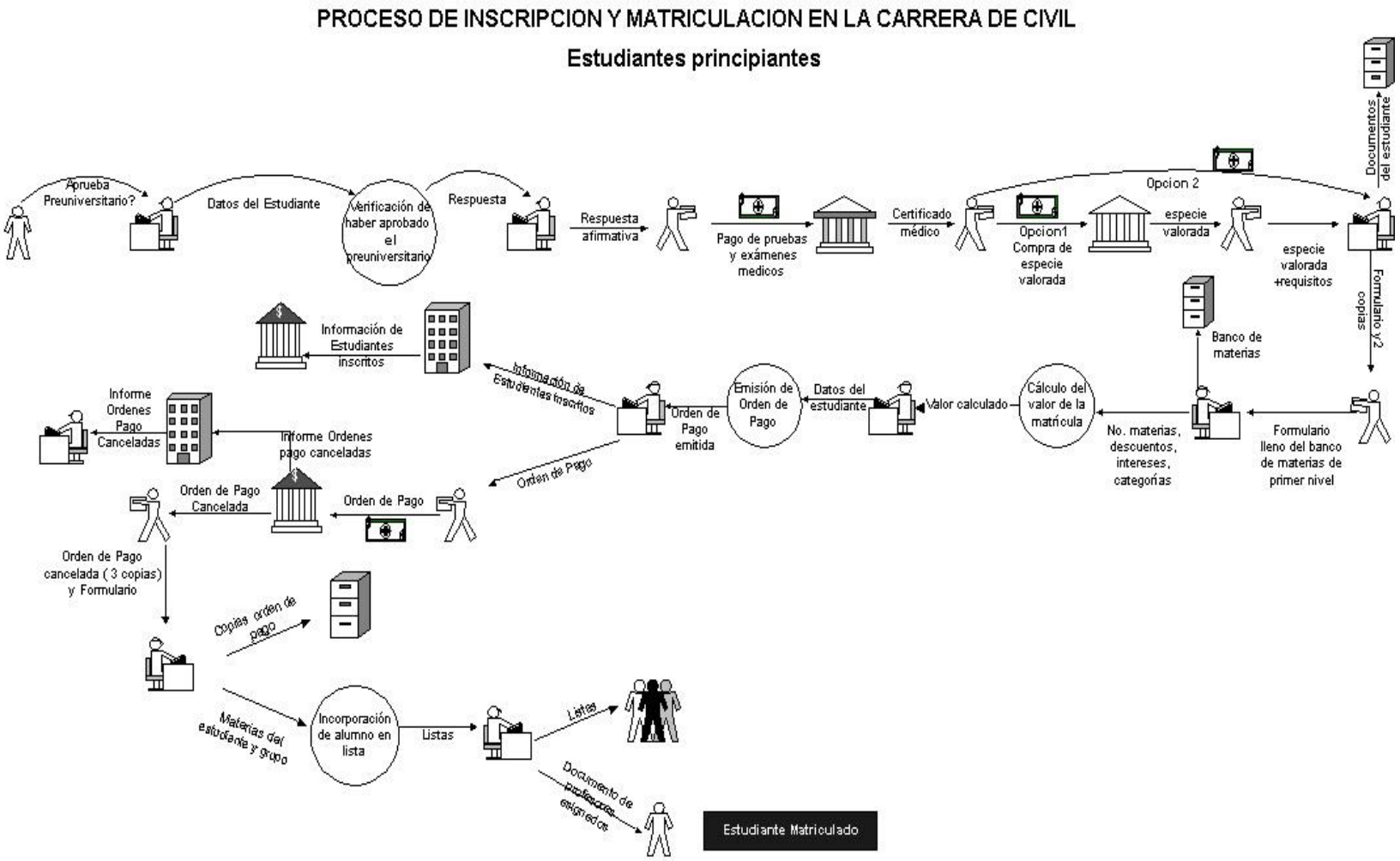
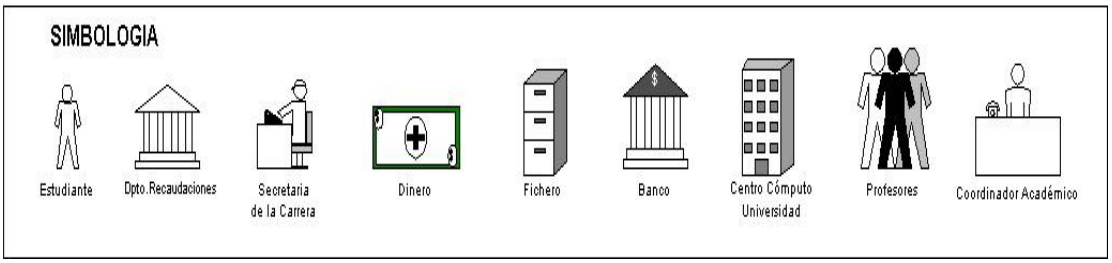


Figura 2.1. Inscripción y Matriculación en la Carrera de Civil para estudiantes de primer año



3. El estudiante entrega a la secretaria de la Carrera los siguientes documentos:

- Una foto tamaño carnet actualizada
- Una fotocopia de la cédula o pasaporte
- Solicitud de inscripción
- 500 hojas tamaño A4

4. La secretaria recepta los documentos que se encuentran completos y en orden, e indica al estudiante que debe presentarse después de una semana.

5. Transcurrido ese tiempo, la secretaria entrega un formulario con dos copias del mismo. El documento a llenar por el estudiante contiene:

a. Datos personales

Apellidos y Nombres

Lugar de Nacimiento (Ciudad y Provincia)

Fecha de Nacimiento

Cédula de Ciudadanía

Título de Grado

Colegio en donde se graduó (indicando si es Nacional, Particular u otros)

Provincia del colegio

Fecha de Graduación

Nota de Grado

Nombre del Padre

Nombre de la Madre

Domicilio

Teléfono del domicilio

Lugar de Trabajo

Teléfono del trabajo

b. Cuadro de las materias a seleccionar

c. Número de Suspensos y Arrastres

6. La secretaria verifica las materias aprobadas en el nivel anterior y los requisitos del nivel a cursar.

7. La secretaria entrega el grupo de materias junto con los nombres de los docentes asignados y horarios de los grupos.

8. El estudiante está obligado a llenar y entregar al Departamento de Secretaría, los tres formularios descritos

anteriormente. En el cuadro de banco de materias, el estudiante selecciona las materias no los grupos. Esto se conoce como preinscripción.

9. La secretaria asienta estos datos en una ficha personal y adjunta uno de los tres formularios.

10. Dependiendo de la decisión que tome el estudiante sobre el número de materias a cursar y de la categoría del colegio de donde proviene, se calcula el valor de la matrícula considerando intereses o descuentos que adquiera el estudiante de acuerdo al Reglamento de Matriculación para las Carreras no autofinanciadas.

11. Si el estudiante solicita crédito, se divide el monto de la matrícula de acuerdo al número de cuotas que las autoridades hayan aprobado.

12. En ambos casos, es decir, de existir o no crédito, se emiten las órdenes de pago, las mismas que serán firmadas por el abogado de la entidad académica.

13. La secretaria envía información de las matrículas al Centro de Cómputo de la Universidad de Guayaquil, siendo este departamento el encargado de entregar la información al Banco y al Departamento de Recaudaciones.

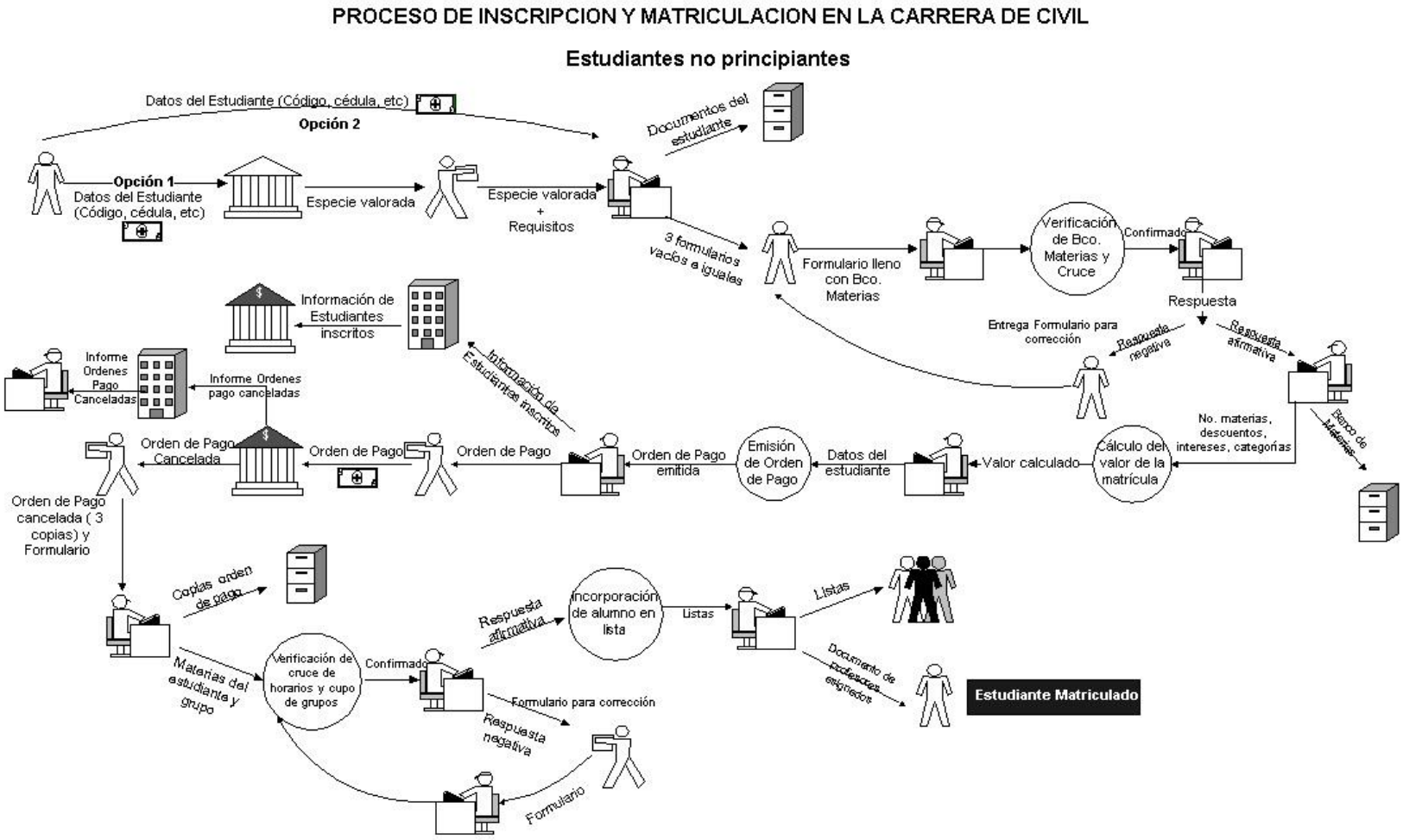


Figura 2.2. Inscripción y Matriculación en la Carrera de Civil para estudiantes no principiantes

14. El estudiante retira las órdenes de pago después de una semana de la preinscripción. Adjunto a las órdenes de pago están los dos formularios.
15. El estudiante debe esperar 48 horas para cancelar la primera orden de pago, en caso de no hacerlo, la secretaria procede a la anulación de la matrícula, por ende no constará en lista. Se exceptúa la anulación de la matrícula previa aprobación de las autoridades de la Carrera.
16. El estudiante entrega a la secretaria de la Carrera, la primera orden de pago con tres fotocopias de la misma y anexa los dos formularios, indicando de acuerdo a la comodidad del horario, el grupo y subgrupo por cada materia a cursar.
17. La secretaria archiva las copias de la orden de pago devolviendo la original al estudiante y entrega la información al Centro de Cómputo de la Universidad.
18. El Banco y el Departamento de Recaudaciones entregan la información del cobro de matrícula al Centro de Cómputo de la Universidad para legalizar su permanencia como alumno de la Carrera.

19. La secretaria verifica el cruce de materias, cruce de horarios, cupo en los grupos previa a la asignación del estudiante a un grupo.
20. La secretaria entrega al estudiante, un documento que especifica las materias y los grupos seleccionados junto con el nombre de los docentes.

Para un mayor entendimiento, observar el gráfico de este proceso. (**Ver Figura 2.2**)

2.1.4.2. Sistema de Evaluación

El ciclo de estudio en esta carrera es anual y está dividido en tres parciales; cada parcial es evaluado sobre 10 puntos. El estudiante aprueba cada materia obteniendo hasta un mínimo de 20 puntos en la suma de los tres parciales.

La Carrera otorga un periodo de gracia para efectuar dos exámenes adicionales: Recuperación y Suspenso.

Al examen de recuperación o también denominado examen de mejoramiento asisten todos los estudiantes que desean mejorar la nota o que han reprobado cualquier materia. No es

obligatorio. Para quienes rindan este examen, se considera la suma de las tres notas más altas de las cuatro calificaciones –entre los tres parciales y el examen de recuperación-, verificando si el alumno cumple con el mínimo de puntos requeridos para la aprobación de la materia.

En caso de no completar 20 puntos, el estudiante cuenta con el examen de suspenso, siempre y cuando haya obtenido hasta un mínimo de 14 puntos.

Para aprobar la materia en el suspenso, la nota mínima es 7. (**Ver Figura 2.3**)

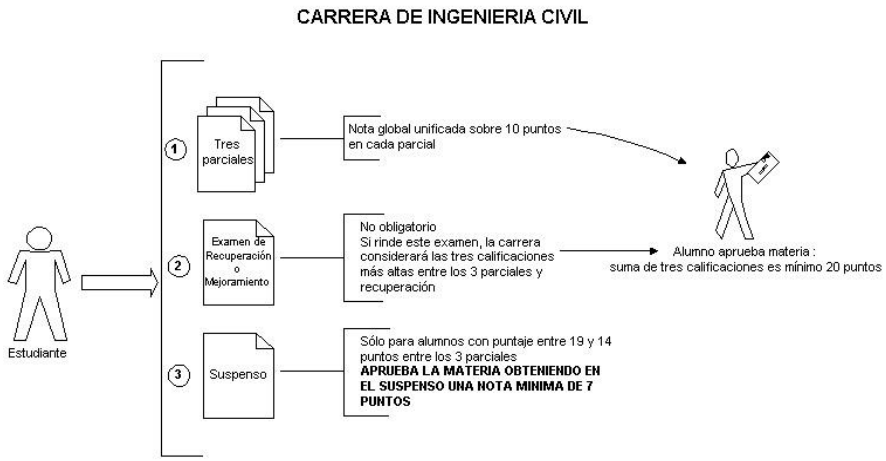


Figura 2.3. Sistema de Evaluación en la Carrera de Civil

2.1.4.3. Especificaciones Funcionales

- Anualmente el Decano y Secretario reúnen al personal de Secretaría para delegar las funciones que realizarán en el proceso de inscripción y matriculación. En la actualidad todo el personal operativo no tiene funciones específicas y desempeñan servicios varios para los alumnos.
- Un grupo de personas están en ventanilla para la recepción de documentación y pueden inscribir a los estudiantes.
- Tres personas se encargan de elaborar el Banco de Materias -asignaturas que cada estudiante puede cursar-.
- En las órdenes de pago, una persona es encargada de su emisión; otra secretaria verifica el pago y revisa los cupos disponibles por curso.
- El Secretario es quien firma las órdenes emitidas, revisa la convalidación de las materias y se encarga de la parte legal de la Carrera.
- Existe otro personal que se encarga de asentar las notas de los alumnos y proveen de certificados si el estudiante lo requiere.

2.1.4.4. Especificaciones Técnicas

- No hay red, todas las máquinas son independientes entre sí.
- Los respaldos se llevan en diskette.
- Las máquinas están actualizadas.

2.1.5. Análisis en la Carrera de Ingeniería de Sistemas Computacionales

2.1.5.1. Proceso de inscripción y matriculación

a) Estudiantes nuevos (Primera vez a Primer año)

1. El estudiante una vez aprobado el preuniversitario se presentará al Departamento de Bienestar Estudiantil de la Universidad de Guayaquil para efectuarse los exámenes médicos, odontológicos y sicológicos previo el pago de una tasa de salud equivalente al 10% del salario mínimo vital. Cabe recalcar que este es un requisito obligatorio para toda persona que ingrese por primera vez a una unidad académica de la Universidad de Guayaquil.

2. Después de dichos exámenes el Departamento de Bienestar Estudiantil otorgará en base a los resultados, un certificado al estudiante indicando idoneidad o incapacidad.

3. El estudiante se acercará al Departamento de Recaudaciones para adquirir una especie valorada.
4. El estudiante elaborará una solicitud -en la especie valorada- para la inscripción a primer año siguiendo un modelo proporcionado por la unidad académica donde desea estudiar.
5. Al inicio del periodo de matriculación la Carrera publicará la información de los docentes asignados a cada materia y el horario de los grupos.
6. El estudiante entregará a la secretaria de la Carrera los siguientes documentos:
   - Tres fotos tamaño carnet actualizada
   - Una fotocopia de la cédula o pasaporte
   - Solicitud de inscripción
   - Una carpeta colgante manila
   - En caso de que el alumno desee estudiar en horario nocturno deberá anexar una fotocopia y el original del carnet de afiliación del Seguro Social, si no posee este documento podrá presentar un certificado de trabajo.
7. La secretaria receptará los documentos que se encuentran completos y en orden.
8. La secretaria entregará al estudiante:

- Un formulario donde se llenará los siguientes datos personales:

  Cédula de Identidad

  Apellidos

  Nombres

  Nacionalidad

  Sexo

  Provincia y Ciudad de Nacimiento

  Fecha de Nacimiento

  Estado Civil

  Lugar de Residencia

  Dirección Domiciliaria

  Teléfono de domicilio

  Nombre y Apellido del Padre

  Nombre y Apellido de la Madre

  Colegio en donde se graduó

  Tipo de Colegio

  Localización

  Costo anual de la matrícula

  Título de Especialización

  Año de graduación

  Nota de Graduación

Lugar de Trabajo (Nombre de la empresa)

Dirección del Trabajo

Teléfono del Trabajo

Carnet de Afiliación al IESS

Email

Grupo Sanguíneo

- Una solicitud detallando las materias y grupo (s) seleccionados por el estudiante
- La malla curricular actualizada

9. La secretaria asignará un código único al educando.

10. El estudiante llenará y entregará a la secretaria, la hoja de datos personales y la solicitud detallando todas las materias de primer nivel. El reglamento universitario indica que está obligado a cursar todas las materias.

11. El Coordinador Académico revisará la solicitud de cada estudiante con la finalidad de que curse todas las materias de primer nivel. Esto se conoce como preinscripción.

12. La secretaria entregará al estudiante un documento indicando las materias que cursará.

13. La secretaria archivará los documentos entregados por el estudiante y aprobados por el Coordinador Académico.

14. De acuerdo al número de materias a cursar y de la categoría del colegio de donde proviene, se calculará el valor de la matrícula considerando intereses o descuentos que adquiera el estudiante de acuerdo al Reglamento de Matriculación para las Carreras Autofinanciadas.

15. El monto de la matrícula se dividirá en cuatro cuotas siendo la distribución del valor de la siguiente manera: 40% para la primera cuota y el 20% en cada una de las restantes. En la primera cuota se incluye los valores extras tales como: cobro de especie valorada (en caso de omitir el paso 3), uso de Internet o emisión del carnet estudiantil.

16. La secretaria entregará información de las matrículas al Centro de Cómputo de la Universidad de Guayaquil, siendo este departamento el encargado de entregar la información al Banco y al Departamento de Recaudaciones.

17. El estudiante retirará las órdenes de pago después de una semana de la preinscripción.

18. El estudiante debe esperar 48 horas para cancelar la primera orden de pago, en caso de no hacerlo, la secretaria procede a la anulación de la matrícula, por ende no constará en lista. Se exceptúa la anulación de la

matrícula previa aprobación de las autoridades de la Carrera.

19. El estudiante entregará a la secretaria de la Carrera, la primera orden de pago con una fotocopia de la misma.
20. La secretaria sellará la copia y entregará la misma al estudiante. El documento original se archiva en la Carrera.
21. La secretaria registrará la orden de pago y entregará la información al Centro de Cómputo de la Universidad.
22. El Banco y el Departamento de Recaudaciones entregarán la información del cobro de matrícula al Centro de Cómputo de la Universidad para legalizar su permanencia como alumno de la Carrera.

Para una mayor comprensión, observar el gráfico de este proceso. (**Ver Figura 2.4**)

b) Estudiantes no principiantes

1. El estudiante se acercará al Departamento de Recaudaciones para adquirir una especie valorada.
2. El estudiante elaborará una solicitud -en la especie valorada- para la inscripción a un nivel superior siguiendo

un modelo proporcionado por la unidad académica donde desea estudiar.

3. Al inicio del periodo de matriculación la Carrera publicará la información de los docentes asignados a cada materia y el horario de los grupos.
4. El estudiante entregará a la secretaria de la Carrera los siguientes documentos:
   - Tres fotos tamaño carnet actualizada
   - Una fotocopia de la cédula o pasaporte
   - Solicitud de inscripción
   - Una carpeta colgante manila
   - En caso de que el alumno desee estudiar en horario nocturno deberá anexar una fotocopia y el original del carnet de afiliación del Seguro Social, si no posee este documento podrá presentar un certificado de trabajo.
5. La secretaria receptará los documentos que se encuentran completos y en orden.

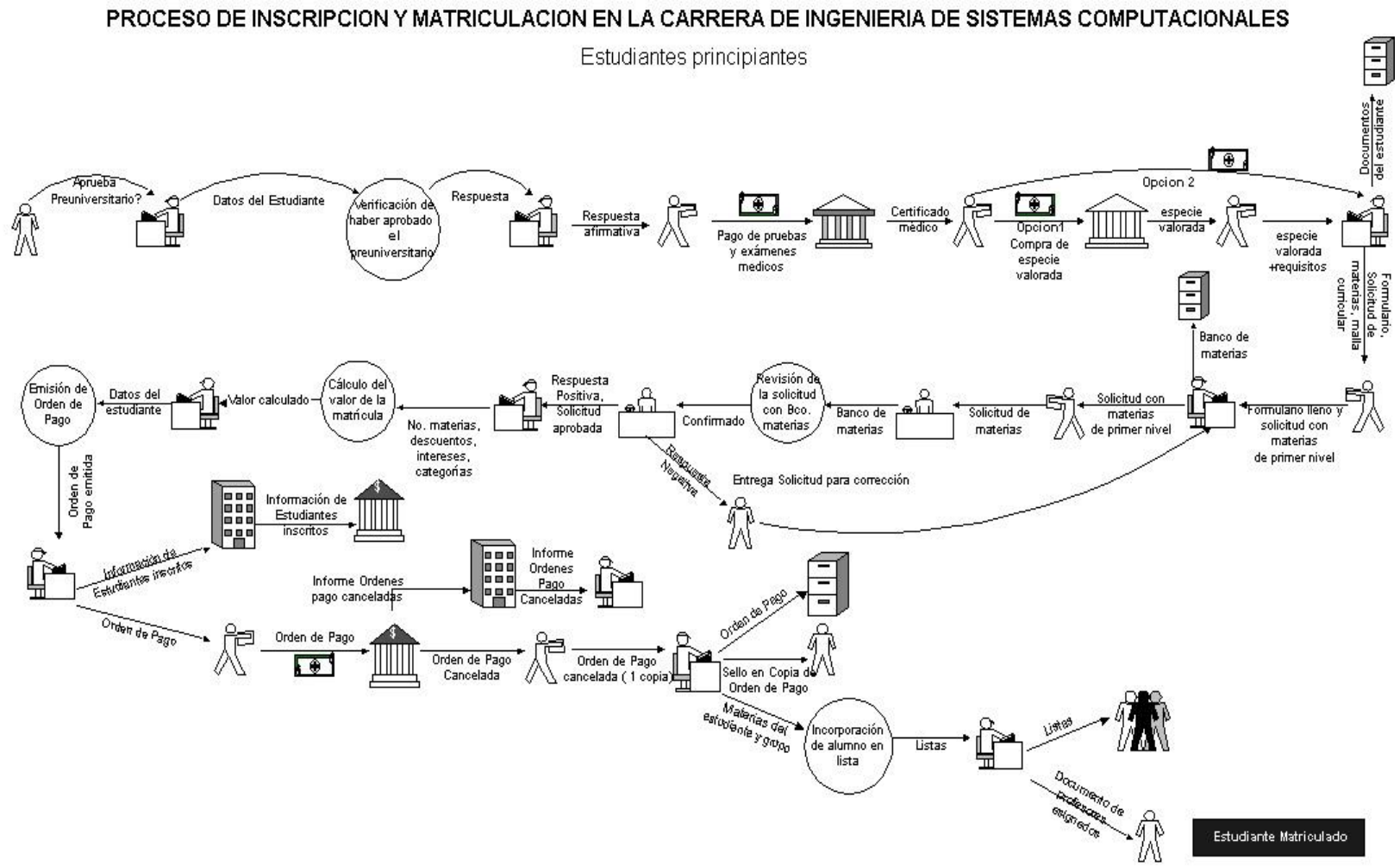
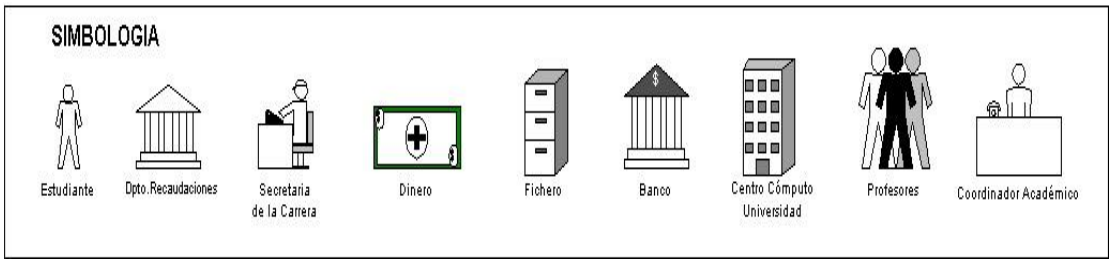


Figura 2.4. Inscripción y Matriculación en la Carrera de Sistemas para estudiantes de primer año



6. La secretaria entregará al estudiante:

- Un formulario donde se llenará los siguientes datos personales:

  Cédula de Identidad

  Apellidos

  Nombres

  Nacionalidad

  Sexo

  Provincia y Ciudad de Nacimiento

  Fecha de Nacimiento

  Estado Civil

  Lugar de Residencia

  Dirección Domiciliaria

  Teléfono de domicilio

  Nombre y Apellido del Padre

  Nombre y Apellido de la Madre

  Colegio en donde se graduó

Tipo de Colegio

Localización

Costo anual de la matrícula

Título de Especialización

Año de graduación

Nota de Graduación

Lugar de Trabajo (Nombre de la empresa)

Dirección del Trabajo

Teléfono del Trabajo

Carnet de Afiliación al IESS

Email

Grupo Sanguíneo

- Una solicitud detallando las materias y grupo (s) seleccionados por el estudiante
- La malla curricular actualizada
- Un documento especificando el promedio de las materias aprobadas en el nivel anterior.

7. El estudiante llenará y entregará a la secretaria, la hoja de datos personales y la solicitud detallando todas las materias del nivel a cursar.

8. El Coordinador Académico revisará la solicitud de cada estudiante con el propósito de verificar los requisitos del nivel a cursar. Esto se conoce como preinscripción.
9. La secretaria entregará al estudiante un documento indicando las materias que cursará.
10. La secretaria archivará los documentos entregados por el estudiante y aprobados por el Coordinador Académico.
11. Dependiendo del número de materias a cursar y de la categoría del colegio de donde proviene, se calculará el valor de la matrícula considerando intereses o descuentos que adquiera el estudiante de acuerdo al Reglamento de Matriculación para las Carreras Autofinanciadas.
12. El monto de la matrícula se dividirá en cuatro cuotas siendo la distribución del valor de la siguiente manera: 40% para la primera cuota y el 20% en cada una de las restantes. En la primera cuota se incluye los valores extras tales como: cobro de especie valorada (en caso de omitir el paso 1), uso de Internet o emisión del carnet estudiantil.
13. La secretaria entregará información de las matrículas al Centro de Cómputo de la Universidad de Guayaquil, siendo este departamento el encargado de entregar la información al Banco y al Departamento de Recaudaciones.

14. El estudiante retirará las órdenes de pago después de una semana de la preinscripción.

15. El estudiante debe esperar 48 horas para cancelar la primera orden de pago, en caso de no hacerlo, la secretaria procede a la anulación de la matrícula, por ende no constará en lista. Se exceptúa la anulación de la matrícula previa aprobación de las autoridades de la Carrera.

16. El estudiante entregará a la secretaria de la Carrera, la primera orden de pago con una fotocopia de la misma.

17. La secretaria sellará la copia y entregará la misma al estudiante. El documento original se archiva en la Carrera.

18. La secretaria registrará la orden de pago y entregará la información al Centro de Cómputo de la Universidad.

19. El Banco y el Departamento de Recaudaciones entregarán la información del cobro de matrícula al Centro de Cómputo de la Universidad para legalizar su permanencia como alumno de la Carrera.

Observar el gráfico de este proceso para un mejor entendimiento del análisis. (**Ver Figura 2.5**)

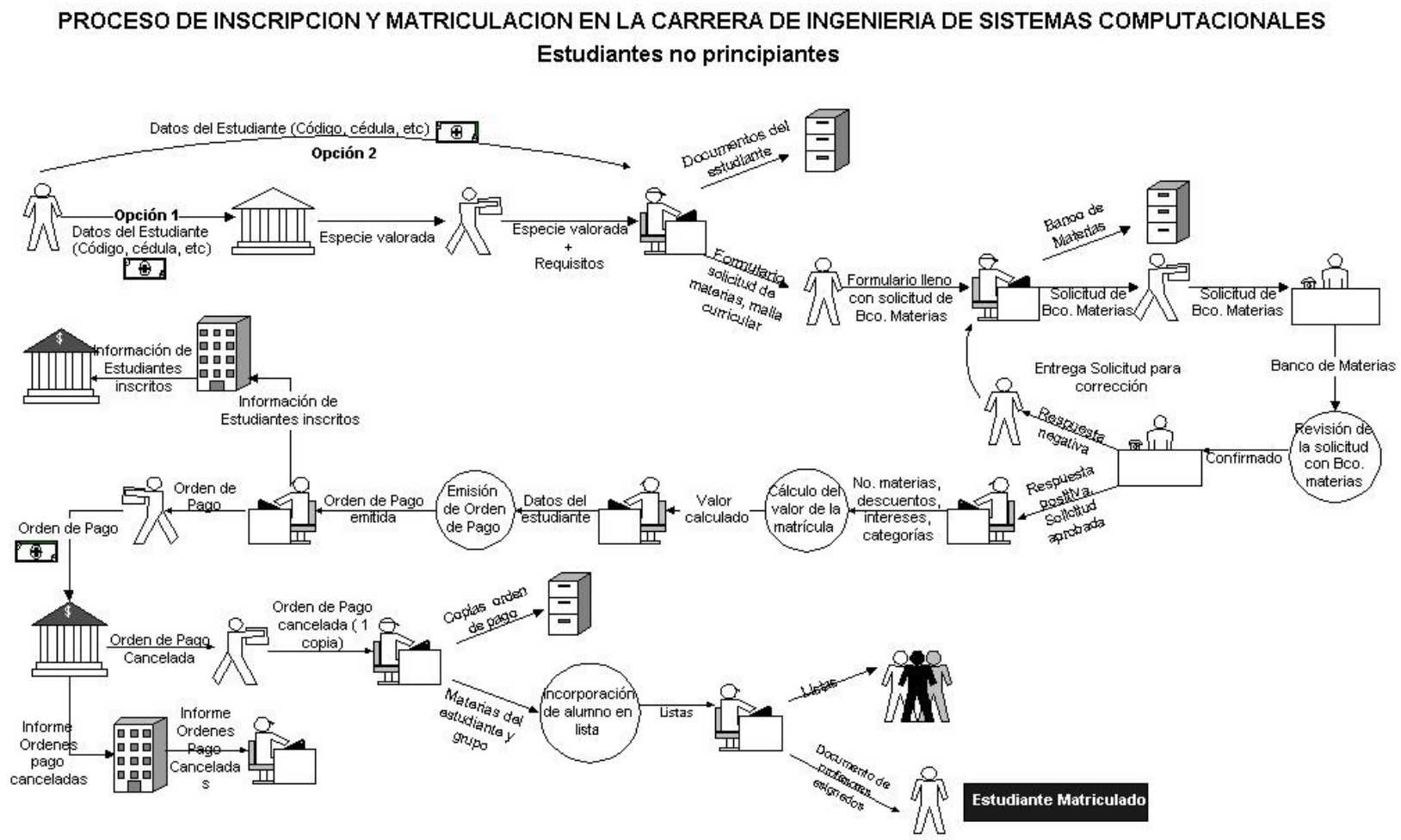


Figura 2.5. Inscripción y Matriculación en la Carrera de Sistemas para estudiantes no principiantes

2.1.5.2. Sistema de Evaluación

El ciclo de estudio en esta carrera es semestral y esta dividido en dos parciales; cada parcial es evaluado sobre 10 puntos. El estudiante aprueba cada materia obteniendo hasta un mínimo de 14 puntos en la suma de los dos parciales.

La Carrera otorga un examen adicional denominado examen de suspenso o mejoramiento, porque un estudiante lo puede rendir sin necesidad de haber reprobado la materia. El requisito para rendir dicho examen es obtener en uno de los dos parciales hasta mínimo 4 puntos.

El promedio final será sobre las dos notas más altas de las tres calificaciones. (**Ver Figura 2.6**)

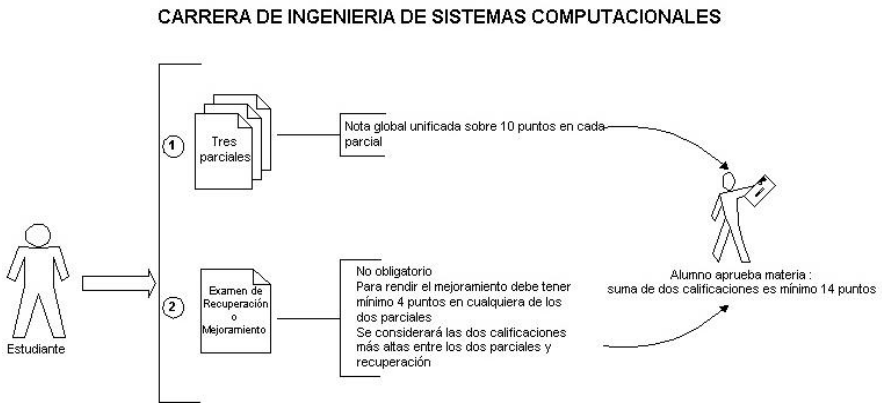


Figura 2.6. Sistema de Evaluación de la Carrera de Sistemas

2.1.5.3. Especificaciones Funcionales

- Actualmente existen una secretaria encargada para dos semestres
- Existe una secretaria encargada del preuniversitario.
- Existe otra secretaria que recepta y entrega documentos generales tales como: órdenes de pago, solicitudes y certificados.
- Una secretaria se encarga de emitir las órdenes de pago para cada estudiante.
- Una secretaria personal para la Directora, quien coordina reuniones y recepta las solicitudes dirigidas a la Directora.
- Existe otro personal que se encarga de asentar las notas de los alumnos y proveen de certificados si el estudiante lo requiere.
- El Secretario es quien firma las órdenes emitidas, revisa la convalidación de las materias y de la parte legal de la Carrera.
- Un economista se encarga de la administración financiera de la Carrera, por ejemplo: de las cotizaciones de los equipos, verifica la cancelación de las órdenes de pago, etc.

- El Coordinador Académico es aquel que verifica los cruces de materias, que cada estudiante esté en el nivel que corresponda, realiza el cronograma de actividades, realiza gestiones para lograr convenios académicos con empresas privadas para beneficio de la Facultad.

2.1.5.4. Especificaciones Técnicas

- Sí hay red porque todas las máquinas de las secretarias se interconectan.
- Existe un sistema diseñado en FoxPro para respaldar las notas de los estudiantes en las materias que hayan tomado durante los semestres.
- Todas las máquinas de secretaría están actualizadas.

2.1.6. Observaciones generales de los procesos en las unidades académicas

a) Proceso de matriculación:

- Ninguna persona puede cursar un nivel a menos que todos los montos pendientes del nivel anterior se cancelen en el Departamento de Recaudaciones de la Universidad.
- Se considera a un estudiante matriculado cuando éste ya ha cancelado la orden de pago y entregado la orden al

Departamento de Secretaría de la unidad, es decir cuando ya consta en listas y tiene asignado un grupo para cada materia.

Existen otros tipos de cursos adicionales que anualmente se imparten, tales como: seminarios de arrastre o avance de materias y son aquellos que se inician en el periodo de invierno siempre y cuando se llene un mínimo cupo para la materia determinada por las autoridades.

b) Pasos para la anulación de la matrícula

a. Cuando el estudiante desea anular su matrícula:

Elabora una solicitud en una especie valorada dirigida al Decano o Director de la unidad pidiendo la anulación de la matrícula con el propósito de no perder el crédito* en cada una de las materias en las que se inscribió en ese periodo lectivo.

b. Cuando el estudiante desea anular una(s) materia(s):

Este caso no aplica para las personas que cursan primer año. El estudiante tiene que elaborar una solicitud dirigida al Decano o Director de la unidad indicando la(s) materia(s) que desea anular para no considerarla como un crédito adicional. En caso de no haber cancelado la orden de pago, se hace el recálculo del valor a pagar emitiéndose la nueva orden.

2.1.7. Identificación de necesidades

Mediante la información obtenida por las entrevistas, observaciones y análisis exhaustivo del proceso actual y sobretodo por el énfasis de nuestra misión, detallamos a continuación las necesidades requeridas en un sistema académico moldeable y acorde a las políticas solicitadas por las unidades académicas:

- Datos centralizados en un servidor pero segmentado en cada entidad académica.
- Registro de hoja de vida de los educandos
- Historial de las calificaciones de los estudiantes
- Calificaciones actualizadas y disponibles para su consulta
- Estadísticas de promedios por materia y por grupo
- Administración de malla curricular
- Control cruce de materias
- Inscripción de los estudiantes
- Cálculo automático del monto de la matrícula
- Control de crédito estudiantil
- Generación automática de listas por materia y por grupo

2.2. Escenario Futuro Propuesto

Como la eficiencia en el proceso académico exige una mayor facilidad en la labor del usuario, entonces, el nuevo sistema proporcionará la información en un tiempo menor en comparación al proceso actual.

2.2.1. Características del sistema

El Sistema Académico otorga mayor permeabilidad para las unidades académicas de la Universidad, es decir, que el manejo de los procesos es abierto y de fácil manejo.

A continuación enfatizamos las propiedades más sobresalientes:

Adaptabilidad

Un sistema es adaptable cuando los cambios exigidos por el medio externo no afectan el funcionamiento del sistema. En nuestro sistema, los factores externos constituyen los reglamentos o políticas actualizadas por las autoridades.

Escalabilidad

Al Sistema pueden incorporarse otros sistemas o módulos a futuro con la finalidad de diseñar un sistema integrado de todas las entidades académicas de la Universidad de Guayaquil.

Modularidad

En un sistema modular, los procesos se dividen en módulos. El Sistema Académico contiene módulos y cada módulo es considerado como ”componente de alto nivel” que en caso de existir un cambio o mantenimiento del mismo, sólo afectará a una parte de la aplicación pero no al resto.

Consistencia e Integridad

Una de las mayores ventajas del Sistema Académico es la consistencia e integridad de datos que al ser modificados no afectan la estabilidad de otros datos existiendo coherencia entre los mismos.

Seguridad

De acuerdo a las políticas de la unidad académica, es recomendable definir las funciones que realiza el personal operativo y de esta manera conocer los roles de acceso al sistema.

Cabe recalcar que el control de acceso a la información no depende exclusivamente del sistema o de los privilegios que un superior otorgue sino también del Administrador de la Base de Datos.

2.2.2. Definición de los módulos y procesos del sistema

Describiremos las actividades que conforman la sistematización académica, las cuales son:

1. **Módulo General**

   - **Registro de parámetros básicos**

     Permite ingresar al sistema información necesaria para determinar la procedencia de los estudiantes y docentes.

2. **Módulo de Matriculación**

   - **Categorías de colegios**

     El sistema permitirá actualizar periódicamente la categoría a la que pertenece el colegio para el cálculo automático de la matrícula.

   - **Actualización de aranceles y descuentos**

     De acuerdo al reglamento universitario en cuanto a matriculación se refiere, se consideran rebajas e intereses que inciden al valor de la matrícula, dependiendo de ciertos casos especiales.

   - **Cálculo del valor de la matrícula**

     Está relacionado con la categoría del colegio, con los aranceles y descuentos que le asignan al estudiante.

   - **Control de cuotas por crédito estudiantil**

Consiste en la partición del valor de la matrícula en varias cuotas mensuales.

- **Emisión de órdenes de pago**

  Es la impresión del valor a cancelar por concepto de matrícula y/o pensiones.

- **Emisión de listas**

  Cuando el estudiante entrega la orden de pago cancelada, es asignado a un grupo para luego constar en lista.

- **Anulación de la matrícula**

  El alumno por cualquier evento puede solicitar la anulación de la misma, y esto se contempla en el sistema.

- **Manejo de Períodos Lectivos**

  Cada vez que se abra un ciclo, un curso extracurricular o haya transcurrido un año lectivo, se puede considerar la apertura de un periodo lectivo y esa información es necesaria también automatizarla. El sistema contemplará la creación de periodos lectivos con fechas tentativas y podrá manipularlos mediante estados.

3. Módulo de Administración de Currículo Docente

Se refiere al mantenimiento de la hoja de vida del catedrático, esto implica la actualización de los datos del docente.

4. **Módulo de Administración de Asignaturas**

   Control del Pénsum existente en cada unidad académica durante el ingreso de la malla curricular (materias y prerrequisitos)

5. **Módulo de estudiantes**

   Mantenimiento de la ficha personal del educando, esto comprende la actualización de datos.

6. **Módulo de Inscripción de Asignaturas**

   - **Selección de materias por los estudiantes**

     En el proceso de inscripción, el sistema facilitará una malla curricular con las materias aprobadas en el nivel anterior y las materias que puede cursar.

   - **Asignación de materias a los docentes**

     La secretaria tendrá que especificar al sistema la materia que le corresponda impartir a cada docente, con el propósito de emitir listas y como información para otros controles, tales como las cargas horarias.

   - **Control de los tipos de cursos curriculares y extracurriculares**

Administrar la información de los cursos (normales, seminarios de arrastre, preuniversitarios, etc.) tales como número de parciales, número de exámenes adicionales (suspenso y/o recuperación), mínima nota para aprobar las materias, mínima nota para alcanzar el suspenso, mínima nota para aprobar el suspenso y mínimo porcentaje de asistencia.

- **Control de cruces de horarios**

  El sistema mostrará un mensaje de advertencia cuando se detecte que un estudiante se ha registrado en materias que coincidan en días, horas clase y sección (matutina, vespertina y nocturna). Nosotros planteamos manejar las horas clase por números (1, 2, 3, 4, 5...) dentro de cada sección.

7. Módulo de Calificación de Asignaturas

- **Ingreso de calificaciones**

  Comprende la digitación de las notas parciales dependiendo de la materia, grupo, año lectivo, tipo de curso y ciclo (en el caso de niveles semestrales son dos ciclos en el año lectivo y en el caso de niveles anuales es un ciclo).

- **Cálculo del promedio**

  El sistema facilitará en base a las notas ingresadas, la generación de promedios por materia.

- **Recalificación**

  En el caso de existir una autorización de parte de las autoridades para la modificación de una nota ingresada, el sistema proveerá la opción de corregir dicha nota.

8. Módulo de Incorporación

- **Ingreso de proyectos de graduación**

  Cuando los estudiantes estén realizando proyectos para culminar sus estudios universitarios en una subunidad, es necesario guardar información sobre cada uno de los proyectos realizados, así como quiénes lo han realizado y qué nota obtuvieron.

9. Módulo de Seguridad

- **Seguridad de acceso al sistema (control de usuarios)**

  Cada usuario tendrá privilegios diferentes para acceder al sistema. El administrador tendrá acceso a todas las opciones y otorgará accesos al resto del personal dependiendo de sus funciones. Además cada usuario tendrá una contraseña que puede ser modificada a su conveniencia.

El nivel de seguridad estará dado a nivel del motor de la Base de Datos y a las autorizaciones que se otorguen a cada usuario dependiendo de su perfil o cargo en la parte académica.

Un perfil es un grupo de valores que define las preferencias y opciones de personalización de un usuario determinado, incluido el acceso a las pantallas, si se le permite grabar cierta información, si puede imprimir o sólo se refiere a consultas. El administrador del sistema define el perfil y le asigna a cada usuario uno, según su cargo en la carrera.

- **Auditoria del sistema**

  Por cada transacción realizada en el sistema se almacenará el nombre del usuario y la fecha que ingresa o modifica cualquier dato.

2.2.3. Proceso de inscripción y matriculación en Ingeniería Civil

1. El estudiante principiante después de constatar que aprobó el preuniversitario y de efectuarse los exámenes en el Departamento de Bienestar Estudiantil, se acercará al Departamento de Secretaría de la Unidad para entregar en una carpeta, el certificado de acuerdo al resultado y anexando los

otros documentos que exige la carrera (solicitud en especie valorada, fotos tamaño carnet, fotocopia de cédula, etc).

Cabe recalcar que si el estudiante pertenece a la carrera, no es necesario realizarse los exámenes médicos, omitiéndose la entrega del certificado pero sí tendrá la obligación de entregar los demás documentos al momento de inscribirse.

2. La secretaria revisará estos documentos y los archivará si están en orden.
3. La secretaria entregará un formulario de datos personales con dos copias incluido el banco de materias, una vez constatadas las materias aprobadas por el estudiante en el sistema.
4. El estudiante deberá llenar los formularios, así como también seleccionar las materias que va a cursar en ese periodo lectivo. En el caso de ser principiante está obligado a cursar todas las asignaturas de primer nivel.
5. La secretaria receptará los formularios entregados al estudiante y revisará que los datos sean correctos para ingresarlos al sistema. Para los estudiantes no principiantes tendrá que verificar mediante el mismo sistema el cruce de materias.
6. El sistema calculará el valor de la matrícula el cual se hará en base a la categoría del colegio, descuentos aprobados, el número

de veces que cursa alguna materia (arrastre) y al número de materias a cursar por el alumno.

7. El sistema emitirá la orden de pago a cancelar por el estudiante. Si éste solicita crédito para cancelar el valor de la matrícula, el sistema permitirá dividir dicho monto en las cuotas que permitan las autoridades.
8. El sistema generará un archivo que contiene los datos de los estudiantes cuya orden de pago han sido emitidas, para ser enviado al centro de cómputo de la Universidad de Guayaquil.
9. Cuando el estudiante cancele el valor de la matrícula, entregará a la secretaria la orden de pago con tres fotocopias de la misma.
10. Si el alumno es principiante, el sistema generará un código aleatorio para el estudiante y la secretaria le asignará un grupo dependiendo del cupo. En el caso de que no sea principiante, el sistema verificará si hay cupo en cada grupo que el alumno seleccionó cursar por materia y avisará mediante un mensaje si hay cruce de horarios.
11. Mediante el sistema, la secretaria emitirá las listas.

2.2.4. Proceso de inscripción y matriculación en Ingeniería de Sistemas Computacionales

1. El estudiante principiante después de constatar que aprobó el preuniversitario y de efectuarse los exámenes en el Departamento de Bienestar Estudiantil, se acercará al Departamento de Secretaría de Sistemas para entregar en una carpeta, el certificado dependiendo del resultado del mismo y anexando los otros documentos que le pide la entidad (solicitud en la especie valorada, fotos tamaño carnet, fotocopia de la cédula, etc.). Cabe recalcar que si el estudiante pertenece a la escuela, no será necesario realizarse los exámenes médicos, pero sí tendrá que entregar los demás documentos al momento de inscribirse.
2. La secretaria revisará estos documentos y los archivará si están en orden.
3. La secretaria entregará un formulario de datos personales con el banco de materias, una vez constatadas las materias aprobadas por el estudiante en el sistema.
4. El estudiante llenará dichos formularios, así como también seleccionará las materias que va a cursar en ese periodo lectivo. En el caso de ser principiante está obligado a cursar todas las asignaturas de primer nivel.
5. El coordinador académico revisará con la ayuda del sistema el banco de materias que llenó el estudiante para constatar que las

materias seleccionadas están acorde al nivel que el estudiante cursará.

6. El coordinador aprobará dicha solicitud.
7. La secretaria registrará esas materias en el sistema.
8. El sistema calculará el valor de la matrícula el cual se hará en base a la categoría del colegio, descuentos aprobados, el número de veces que cursa alguna materia (arrastre) y al número de materias a cursar por el alumno.
9. El sistema emitirá las órdenes de pago a cancelar por el estudiante.
10. El sistema generará un archivo que contiene los datos de los estudiantes cuyas órdenes de pago han sido emitidas, para ser enviado al centro de cómputo de la Universidad de Guayaquil.
11. Cuando el estudiante cancele el valor de la matrícula, entregará a la secretaria la orden de pago con una fotocopia de la misma. El sistema avisará mediante un mensaje si hay cruce de horarios.
12. Mediante el sistema, la secretaria emitirá las listas.

2.2.5. Funciones recomendadas para los usuarios

El sistema académico tiene la ventaja de ser flexible al acceso al sistema a través de una persona que ocupará el cargo de supervisor

o administrador, es decir, aquel que tendrá el acceso a todas las interfaces e informes de la aplicación.

De las funciones recomendadas para los usuarios del sistema tenemos:

- Administrador del Sistema
- Matriculación
- Registro de Datos de Estudiantes y Banco de Materias
- Registro y Consulta de Notas
- Emisión de las Ordenes de Pago
- Sólo Consulta de Notas

2.2.6. Requerimientos de software y hardware para la etapa de desarrollo

Las herramientas tanto de hardware como de software que solicitamos para el desarrollo del sistema se detallan a continuación:

Hardware

Servidor de las siguientes características:

- Procesador Pentium 4 de 1.70GHz
- Memoria 256 MB
- Disco duro de 80 GB

- Impresora Lexmark E210
- Impresora Epson LX-300

Cliente de las siguientes características:

- Procesador Pentium 3 de 533Mhz
- Memoria 128 MB
- Disco duro de 40 GB

Software

- Sistema Operativo Microsoft Windows 2000 Server y Professional
- Erwin 3.5 como modelador de datos para el diagrama Entidad-Relación.
- Visio Professional 5.0
- Motor de Base de Datos SQL Server 2000
- Microsoft Visual Basic .Net
- Crystal Reports para Visual Studio .Net

2.2.7. Explicación de la Plataforma Tecnológica

Previo a la selección de estas herramientas hemos considerado los siguientes puntos.

SQL Server 2000

- SQL Server 2000 disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos y seguridad integrada.
- Las innovaciones permiten que SQL Server 2000 sea el líder en varias de las categorías de aplicación de rápido crecimiento en la industria de base de datos.
- SQL Server 2000 es la primera base de datos en escalar desde la computadora portátil hasta la empresa utilizando la misma base de código y ofrecer el 100% de compatibilidad de código.
- Contiene la más amplia gama de opciones de replicación de cualquier base de datos.
- Incluye la mejor integración con Windows NT Server y Microsoft Transaction Server.
- Revoluciona el concepto de Base de Datos para la Empresa. Reúne en un sólo producto la potencia necesaria para cualquier aplicación empresarial crítica junto con unas herramientas de gestión que reducen al mínimo el coste de propiedad. Con Microdoft SQL Server, la empresa tiene todo de serie.

- Miles de Soluciones Disponibles: Tendrá libertad de elección, ya que todas las aplicaciones de gestión del mercado corren sobre Microsoft SQL Server
- Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sites*.
- Potencia: Microsoft SQL Server es la mejor base de datos para Windows NT Server.
- Gestión: Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.
- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.

Visual Studio .Net 2002

- Visual Studio .NET permite a los programadores generar con rapidez aplicaciones para Internet de próxima generación orientadas a cualquier dispositivo y que se integran en cualquier plataforma. Con el más moderno y completo entorno de programación.

- Los programadores de Windows encontrarán los nuevos formularios Windows Forms intuitivos y muy eficaces. Estos formularios son compatibles con cualquier lenguaje basado en .NET, incluidos Microsoft Visual Basic .NET y Microsoft Visual C#.NET.
- Para el mayor alcance posible de dispositivos compatibles con Internet, Visual Studio .NET proporciona características de Mobile Internet que permiten a los programadores generar una única interfaz Web móvil compatible con una amplia gama de dispositivos, HTML compacto (cHTML) para teléfonos i-mode y HTML para Pocket PC, dispositivos de mano y localizadores (pagers). Los controles móviles de servidor generan de manera inteligente la reproducción y paginación apropiada para cada dispositivo Web, proporcionando una experiencia completa y coherente al usuario a la vez que mantiene la flexibilidad del programador.

Además de las características mencionadas con anterioridad, también la selección de estas herramientas se deben a:

- Uniformidad en el uso de las herramientas de desarrollo con otros grupos de tesis involucrados con el Sistema Académico.
- Facilidad en el aprendizaje de la herramienta:

- Capacitación proporcionada por expertos en el manejo de la herramienta
- Disponibilidad de material didáctico.

El convenio establecido entre la Universidad y Microsoft permiten que el apoyo de la Carrera de Sistemas referente al aprendizaje de las herramientas, a través del Ing. Milton Maridueña sean una realidad; cumpliendo de esta manera uno de nuestros objetivos propuestos en la tesis.

CAPITULO 3

3. FASE DE DISEÑO

3.1. Estándares para el diseño de la Base de Datos

3.1.1. Prefijos de los módulos

El sistema académico es amplio por lo que se ha procedido dividirlo en varios módulos, por esta razón cada tabla que se ha creado pertenece a un módulo diferente, y para identificarlos se han tomado las iniciales de cada uno de ellos como prefijo.

Así tenemos, el prefijo del **M**ódulo **G**eneral es MG, el prefijo del **M**ódulo de **M**atriculación es MM, el prefijo del **M**ódulo de **I**nscripción es MI, el prefijo del **M**ódulo de I**n**corporación es MN (para diferenciarlo del anterior), etc.

3.1.2. Formato del nombre de las tablas

El nombre de la tabla y su reconocimiento como tal se hará con el identificador "TB", considerando el siguiente formato:

"TB" + Prefijo del Módulo + "_" + Identificación de la Tabla

La identificación de la tabla consiste en:

a) si es una palabra: los primeros 10 caracteres en singular.

b) 2 palabras: 5 primeros caracteres de la primera palabra + "_" + 4 caracteres de la segunda palabra.

c) 3 palabras: 3 primeros caracteres de las primera palabra + 2 primeros caracteres de la segunda palabra + "_" + 4 primeros caracteres de la tercera palabra.

Por ejemplo: TBME_Estudiante (tabla Estudiante correspondiente al módulo ME), TBMC_Calif_Parc (tabla calificación parcial correspondiente al MC), etc.

3.1.3. Formato del nombre de los campos

Para cada campo se debe seguir el siguiente formato:

Identificación de la Tabla + "_" + Identificación del Campo

La identificación de la tabla consiste en:

a) 1 palabra: 4 primeras letras de la tabla.

b) más de 1 palabra: 2 primeras letras del primer nombre + 2 primeras letras del segundo nombre.

La identificación del campo consiste en:

a) 1 palabra: 10 primeros caracteres.

b) 2 palabras: 5 primeros caracteres de la primera palabra + "_" + 4 primeros caracteres de la segunda palabra.

c) 3 palabras: 3 primeros caracteres de la primera palabra + 2 primeros caracteres de la segunda palabra + "_" + 4 primeros caracteres de la tercera palabra.

Por ejemplo, campos de la tabla TBME_Estudiante: Estu_Codigo, Estu_Apellido, Estu_Nombre, etc. Campos de la tabla TBMC_Calif_Parc: CaPa_Estud_Subu, CaPa_Asignatura, CaPa_Tipo_Calif, etc.

3.2. Módulos del Sistema Académico

- **Módulo General**

La abreviatura para este módulo es MG y comprende tablas referentes a: País, Provincia, Ciudad y Universidad.

- **Módulo de Asignaturas**

La abreviatura para este módulo es MA y comprende tablas referentes a: Facultad, Tipo de Subunidad, Subunidad Académica, Área Académica, Asignatura, Pénsum, Motivos de Cambio de Pénsum y Prerequisitos.

- **Módulo de Docentes**

La abreviatura para este módulo es MD y comprende tablas referentes a: Título Universitario y Docente.

- **Módulo de Estudiantes**

La abreviatura para este módulo es ME y comprende tablas referentes a: Especialización, Foto Estudiante y Estudiante.

- **Módulo de Inscripción**

La abreviatura para este módulo es MI y comprende tablas referentes a: Tipo de Curso, Docente-Asignatura, Horario-Grupo, Estudiante-Subunidad, Estudiante-Asignatura, Estudiante-Nivel, Horario- Estudiante, Estudiante-Pénsum.

- **Módulo de Matriculación**

La abreviatura para este módulo es MM y comprende tablas referentes a: Categorías, Colegios, Valores de Matrícula Global, Valores de Matrícula por Asignatura, Incremento de Matrícula, Descuentos y Aranceles, Conceptos de Cobros, Bancos, Órdenes de Pago, Detalle de Matrícula, Cuotas, Autoridades y Períodos Lectivos.

- **Módulo de Calificaciones**

La abreviatura para este módulo es MC y comprende tablas referentes a: Tipo de Calificación, Calificación Parcial, Calificación Promedio, Entrega de Calificaciones, Tipo de Solicitudes y Solicitudes Especiales.

- **Módulo de Incorporación**

La abreviatura para este módulo es MN y comprende tablas referentes a: Tipo de Proyectos y Proyecto de Grado.

- **Módulo de Seguridad**

La abreviatura para este módulo es MS y comprende las tablas referentes a: Pantalla, Perfil, Permiso y Usuario.

3.3. Diseño de la estructura del Sistema Académico

- **TBMG_PAIS**

Esta tabla guarda información sobre cada país indicando la nacionalidad correspondiente. Nos servirá de mucha ayuda para señalar la procedencia de un docente o de un estudiante.

- **TBMG_PROVINCIA**

La tabla almacena información sobre las provincias o estados de cada país.

- **TBMG_CIUDAD**

Esta tabla archiva información sobre las ciudades de cada país.

- **TBMG_UNIVERSIDA**

Esta tabla guarda información sobre las universidades más importantes. Ésta puede ayudar en caso de convalidaciones o en el caso que un estudiante sea un profesional de otra universidad. Es más usada al llenar la ficha de los docentes.

- **TBMA_FACULTAD**

Esta tabla guarda información sobre cada facultad que exista en la universidad.

- **TBMA_TIPO_SUBU**

Esta tabla guarda información sobre los tipos de subunidad que existan en la universidad, tales como: Escuelas, Carreras, Institutos, etc.

- **TBMA_SUBUN_ACAD**

Esta tabla guarda información sobre las subunidades académicas especialmente a qué facultad o a qué unidad corresponde.

- **TBMA_AREA_ACAD**

Esta tabla guarda información sobre las áreas académicas a las que pertenece cada asignatura.

- **TBMA_ASIGNATURA**

Esta tabla guarda información sobre las asignaturas que se deben estudiar para completar una carrera universitaria.

- **TBMA_PENSUM**

Esta tabla guarda información sobre cada malla curricular que se vaya creando durante la vida de una subunidad académica, lo más importante es señalar las fechas de vigencia y de caducidad.

- **TBMA_MOTCA_PENS**

Esta tabla guarda información sobre todos las razones o motivos posibles por la que un estudiante deba cambiar un pénsum. Es muy importante para una subunidad que contenga varias mallas curriculares.

- **TBMA_PREREQUISI**

Esta tabla guarda las cadenas de cada materia, es decir, los requisitos necesarios para cursar una determinada asignatura.

- **TBMD_TITUL_UNIV**

Esta tabla contiene información sobre títulos universitarios. Es usada cuando se llena la ficha de docentes.

- **TBMD_DOCENTE**

Esta tabla guarda información sobre los docentes que pertenecen a la universidad.

- **TBME_ESPECIALIZ**

Esta tabla contiene información de todas las especializaciones del bachillerato. Es usada cuando se llena la ficha de estudiantes.

- **TBME_FOTO_ESTU**

Esta tabla guarda imágenes correspondientes a las fotos de los estudiantes.

- **TBME_ESTUDIANTE**

Esta tabla guarda información sobre los estudiantes que pertenecen a la universidad.

- **TBMI_TIPO_CURS**

Esta tabla contiene información sobre los tipos de curso que se van abriendo en una subunidad. La información que se guarda es el total de puntos que un estudiante debe reunir para aprobar cada materia dentro de ese tipo de curso, el mínimo porcentaje de asistencia, el número de parciales, el número de exámenes adicionales, etc. Es muy importante porque se considera como parte de cada período lectivo que se cree dentro del sistema.

- **TBMI_DOCEN_ASIG**

Esta tabla almacena información sobre los distributivos de las cargas horarias a los docentes. Guarda un histórico de todos los docentes que hayan dado alguna asignatura en cada subunidad.

- **TBMI_HORAR_GRUP**

Esta tabla guarda información temporalmente sobre los horarios de cada grupo abiertos en cada período lectivo. Cabe destacar que esa información se borra cuando se cierra dicho período lectivo.

- **TBMI_ESTUD_SUBU**

Esta tabla guarda información sobre los estudiantes cuando pertenecen a una determinada subunidad académica. También guarda un histórico referente a su promedio de egresado y de graduación, así como también las fechas de ingreso y egreso y qué proyecto realizó para su graduación.

- **TBMI_ESTUD_ASIG**

Esta tabla contiene información acerca de las asignaturas que haya tomado un alumno durante su vida estudiantil en una determinada subunidad.

- **TBMI_ESTUD_NIVE**

Esta tabla guarda información acerca de cada estudiante en los niveles que va cursando y/o repitiendo.

- **TBMI_HORAR_ESTU**

Esta tabla guarda información temporalmente sobre el horario que se va armando cuando un estudiante toma varias asignaturas. Así como en la tabla de “Horarios por Grupos”, esta información se borra cuando se cierra un período lectivo.

- **TBMI_ESTUD_PENS**

Esta tabla guarda información sobre todos los pénsum que haya tomado un estudiante por algún motivo específico.

- **TBMM_CATEGORÍA**

Esta tabla contiene todas las categorías en las que se dividan los colegios. Otras categorías que se guardan son: bachilleres extranjeros, profesionales nacionales y profesionales extranjeros.

- **TBMM_COLEGIO**

Esta tabla guarda información sobre todos los colegios que existen en nuestro país y las categorías a las que pertenecen.

- **TBMM_MATRI_GLOB**

Esta tabla guarda los valores de la matrícula global dividida en categorías.

- **TBMM_MATRI_ASIG**

Esta tabla guarda los valores de la matrícula por cada asignatura dividida en categorías.

- **TBMM_INCRE_MATR**

Esta tabla guarda los porcentajes de incremento por cada tipo de matrícula.

- **TBMM_DESCU_ARAN**

Esta tabla contiene información de los descuentos y aranceles que existan en la universidad.

- **TBMM_CONCE_COBR**

Esta tabla almacena todos los conceptos de cobros por los que se emita una orden de pago.

- **TBMM_BANCO**

Esta tabla guarda información de los bancos y las cuentas que tiene la universidad en cada uno de ellos.

- **TBMM_ORDEN_PAGO**

Esta tabla almacena información sobre las órdenes de pago emitidas por concepto de matriculación.

- **TBMM_DETAL_MATR**

Esta tabla guarda todos los detalles que se van ingresando en cada orden de pago emitida por concepto de matriculación.

- **TBMM_CUOTA**

Esta tabla contiene información sobre todas las cuotas creadas en la emisión de órdenes de pago.

- **TBMM_AUTORIDAD**

Esta tabla almacena información acerca de las autoridades de cada subunidad y de una firma digitalizada para cada autoridad.

- **TBMM_PERIO_LECT**

Esta tabla tiene información respecto a cada período lectivo que se aperture en el sistema. Cabe destacar que el período lectivo está formado en año, el ciclo y tipo de curso; cada vez que se inicie un nuevo curso se tiene que abrir un período lectivo. En esta tabla se consideran estados para cada período, así como también la fecha de inicio y la fecha de fin.

- **TBMC_TIPO_CALI**

Esta tabla tiene información sobre los tipos de calificaciones que existan en una subunidad. Es muy importante para determinar el parcial cuando se proceda a registrar las calificaciones de los estudiantes, y también cuando se vaya a generar los promedios para determinar si una calificación se trata de suspenso o de mejoramiento o de ambas.

- **TBMC_CALIF_PARC**

Esta tabla registra todas las calificaciones parciales de todos los estudiantes.

- **TBMC_CALIF_PROM**

Esta tabla guarda los promedios de cada estudiante en cada asignatura tomada.

- **TBMC_ENTRE_CALI**

Esta tabla almacena información sobre las fechas topes de entrega de notas parciales por parte de los docentes.

- **TBMC_TIPO_SOLI**

Esta tabla registra todos los tipos de solicitudes por lo que un estudiante pueda pedir una recalificación, un cambio de categoría, etc.

- **TBMC_SOLIC_ESPE**

Esta tabla almacena todas las solicitudes de los estudiantes que hayan sido aprobadas por las autoridades; es más usada cuando se ingresan las recalificaciones.

- **TBMN_TIPO_PROY**

Esta tabla contiene los tipos de proyectos tales como seminarios, tesis, etc.

- **TBMN_PROYE_GRAD**

Esta tabla contiene todos los proyectos que se vayan realizando en la subunidad para la culminación de los estudios de cada alumno de la subunidad.

- **TBMS_PANTALLA**

Esta tabla tienen todas las pantallas del sistema académico.

- **TBMS_PERFIL**

Esta tabla contiene los perfiles o cargos para poder accesar al sistema.

- **TBMS_PERMISO**

Esta tabla tiene los permisos sobre las pantallas otorgados a cada perfil.

- **TBMS_USUARIO**

Esta tabla tiene información acerca de los usuarios de la base de datos.

3.4. Estándares en el diseño de las pantallas

La interfaz de usuario es el sistema que permite al usuario interactuar con el programa o aplicación que está usando. El uso de interfaces gráficas tales como Windows, ha facilitado el aprendizaje y el manejo de los programas, al ser más humanizados, sencillos, intuitivos y atractivos con respecto a sus antecesores basados en texto.

El diseño tiene como objetivo primordial el encontrar la mejor representación para poder comunicar algún tipo específico de información. El diseñador se ve constantemente requerido a resolver demandas conflictivas entre el problema, el plan de trabajo, el presupuesto y el nivel de calidad deseado. Entre las principales fuentes de conflictos está la aparente oposición entre los criterios funcionales y estéticos. El diseño visual orientado a la comunicación ve a estas fuerzas no como dos oponentes irreconciliables, sino más bien como componentes cooperativos de toda solución de alta calidad.

Los siguientes puntos describen algunas de las reglas y técnicas de diseño que se han aplicado de manera más directa al diseño de las pantallas del sistema académico.

3.4.1. Ubicación de los controles en la forma*

- Los labels* o etiquetas van alineados a la derecha.
- Los textBoxes* o cajas de texto se alinean tanto horizontal como verticalmente. Así mismo sólo tienen tamaños predeterminados.
- El color de fondo de los textBoxes diferencia a los campos de ingreso de datos con los de sólo salida.
- Los frames* sólo se emplean discretamente y llevan un título.
- Todos los botones van situados en el lado derecho de la forma.
- Los labels dentro de cada frame van alineados con los que están fuera del mismo.
- Todas las formas tienen los botones para minimizar, maximizar y cerrar la ventana.
- Se puede desplazar de un control a otro dentro de un mismo formulario, por medio de la tecla “Tab” o de “Enter”.
- Todas las formas aparecen con el mismo estilo de borde.
- El primer carácter de los labels y menús está en mayúscula.
- Todos los controles están alineados correctamente.
- Los OptionButtons* siempre vienen encerrados en un Frame.
- En cada grupo de OptrionButtons hay uno asignado por defecto.
- Los campos de fecha tienen el mismo formato (aaaa/MM/dd).

3.4.2. Propiedades generales

Las siguientes propiedades se establecen para todos los controles dentro de los formularios:

Propiedad	Descripción	Valor
FontName	Nombre de la Fuente	MS-Sans Serif
FontSize	Tamaño de la Fuente	8.25
FotntBold	Con Negrita	False

Tabla 3.1. Propiedades Generales

3.4.3. Formas

Llamamos formas a las ventanas de la aplicación, las mismas que deben ser movibles, esto le permite al usuario pueda abrir más de una forma a la vez. En el sistema académico, el tamaño de las formas depende de la cantidad de información que necesite ser visualizada.

A continuación, se detallan las propiedades comunes en las formas:

Propiedad	Descripción	Valor
Width	Ancho	550 (pantalla de mantenimiento)

Height	Alto	270 (pantalla de mantenimiento)
FormBorderStyle	Estilo	FixedSingle
GridSize	Cuadrícula	4; 4
StartPosition	Posición en Monitor	Manual
Name	Nombre	(empieza con el prefijo Frm)
Icon	Icono	Logo de la Universidad

Tabla 3.2. Propiedades de la Forma

3.4.4. Botones

Permiten ejecutar una acción representada por el texto que tenga el botón. Se ubican verticalmente en la parte lateral derecha, situando en la parte superior la opción por omisión (Nuevo, Aceptar, Seleccionar, etc) para dicha transacción o forma. Se dejan al final los botones de Cancelar, Retornar y Salir. En todas las formas, el botón Salir, están separados del resto.

Las dimensiones de los botones están dadas de la siguiente manera:

Propiedad	**Descripción**	**Valor**
Width	Ancho	80
Height	Alto	32
Name	Nombre del Botón	(empieza con el prefijo Cmd)
Image	Imagen a agregar	(dependiendo de la acción que va a realizar el botón)
Location	Posición	Las coordenadas del primer botón en la mayoría de las formas son 460;16. La distancia entre los botones es de 30

Tabla 3.3. Propiedades de los botones

Los textos en los botones por lo general son verbos que denotan la acción para el que están destinados. Por ejemplo: Aceptar, Modificar, Guardar, Cancelar, Nuevo, Buscar, Procesar, Dividir, Imprimir, Retornar, Salir, etc.

Se puede acceder a todos los botones desde el teclado sin utilizar el mouse. Para ello se han empleado los Short Cuts o caminos

cortos de acceso, como por ejemplo la combinación de **ALT** y **A** nos permite acceder directamente al botón de Aceptar. Para representar las teclas disponibles para cada botón, se ha subrayado la letra que la ejecuta.

3.4.5. TextBoxes

Los TextBoxes son controles de edición dentro de los cuales el usuario escribe información. El usuario puede aceptar el contenido del texto que ya se encontraba en el campo, borrarlo, cambiarlo o reemplazarlo. Cabe recalcar que estos controles se alinean por la izquierda.

Todos los controles de este tipo tienen las siguientes propiedades:

Propiedad	**Descripción**	**Valor**
BackColor	Color de Fondo	Blanco (si es de ingreso) Gris (si es de salida)
CharacterCasing	Caracteres permitidos	Upper
Name	Nombre	(empieza con el prefijo Txt)

BorderStyle	Estilo de Borde	Fixed3D
ForeColor	Color de Frente	Negro
Height	Alto	20
Location	Posición	La distancia entre los botones es de 31

Tabla 3.4. Propiedades de los Textboxes

3.4.6. ComboBoxes

El Combo Box es un caja de texto (texbox) con una lista integrada, relacionada e interdependiente. Son útiles cuando el usuario requiera ingresar datos y pueda ver una lista de los posibles datos que puede ingresar.

Estos incluyen un textbox y un Standard list (lista de selección). La lista debe aparecer indentada con respecto al texto para que quede plenamente identificada. Las entradas en la lista pueden estar ordenadas por código o bien en orden alfabético.

La propiedad más importante es DropDownStyle (estilo de combo) que en el sistema tiene el valor DropDownList y el nombre de cada control de este tipo empieza con el prefijo Cbo.

3.4.7. Option Buttons

Permiten al usuario escoger sólo una de varias opciones posibles. Debe marcar la que haya elegido. Las dimensiones dependen de la cantidad de opciones que existan para determinada acción. El nombre de cada control de este tipo empieza con el prefijo Opt.

3.4.8. CheckBoxes*

Permite al usuario tomar o no varias opciones o atributos a la vez. Cuando la opción es tomada, se marca con un visto; cuando no, queda en blanco. Al igual que el caso anterior, la dimensión dependerá de la cantidad de opciones que existan. El nombre de cada control de este tipo comienza con el prefijo Chk.

3.4.9. GroupBoxes o Frames

Los GroupBoxes técnicamente son considerados controles, pero no procesan ninguna entrada ni del mouse ni del teclado. Ellos son sólo provistos para agrupar controles relacionados. Un GroupBox consiste en un frame rectangular con una etiqueta en la parte superior izquierda que lo describe. El título por lo general va de color azul oscuro.

3.4.10. DataGrid

El DataGrid es un control que sirve para las consultas a la base de datos. Se emplean las propiedades que vienen por defecto en el Visual Basic .Net.

3.4.11. MaskedBox

Este control es muy similar a las cajas de texto pero son más usadas para ingresar fechas. El formato para las fechas a emplear es: 4 dígitos del año / 2 dígitos del mes / 2 dígitos del día.

3.4.12. Labels

Los labels (etiquetas) identifican la función del control y proveen acceso directo del teclado al control. La primera letra en dichas etiquetas es mayúscula. Todos los labels están alineados a la derecha.

CONCLUSIONES

- La Facultad de Ciencias Matemáticas y Físicas debe contar con un Sistema Académico que cumpla con todas sus necesidades y acorde a las nuevas tecnologías informáticas existentes en nuestro mercado.

- El Sistema Académico proporcionará mayor agilidad al proceso de matriculación y se adaptará a los cambios futuros en los reglamentos académicos de la Facultad.

- Permitirá el manejo de perfiles para mantener la información lo más segura posible.

RECOMENDACIONES

Técnicas

- ✓ Cada persona que maneje el sistema requerirá de una computadora personal y todas estas computadoras estarán conectadas a su vez a una computadora de plataforma tipo servidor, es decir, con mejores y mayores características que una computadora común porque en el servidor estará almacenada la base de datos, todos estos ordenadores formarán parte de una red local.

- ✓ Además como nuestro sistema generará informes es necesario la adquisición de impresoras.

Generales

- ✓ Si la Carrera de Sistemas continúa presentando sus aplicaciones en plataformas nuevas y actualizadas con el afán

de incursionar en nuevos ambientes informáticos incrementará el compromiso de sus estudiantes para mejorar los servicios de la Universidad de Guayaquil.

- ✓ El Sistema Académico desarrollado en Visual Studio .Net servirá como modelo e incentivo para que futuros proyectos de la Escuela se basen en esta infraestructura tecnológica que nos proporciona seguridad y disponibilidad de la información.

GLOSARIO

Checkbox.- Casilla de Verificación.

Crédito.- Existen tres definiciones:

1. Crédito en las Ordenes de Pago: Describe las divisiones del valor de la matrícula en varias cuotas.
2. Créditos ganados en materias aprobadas: Se maneja en la carrera de Civil, son puntos acumulativos a favor del estudiante como requisito previo de la graduación, es decir, cada materia tiene un valor diferente y para egresar debe completar un determinado puntaje.
3. Crédito en la inscripción: Crédito también se refiere al número de veces que uno cursa una materia. Recordemos que hasta un tercer crédito es válido, a partir del cuarto crédito se necesita la autorización de la directiva.

Forma.- Pantalla que requiere de datos y de un evento para generar funcionalidad.

Frame.- Control de Visual Basic.net que permite agrupar otros controles como: Label, Textbox.

Label.- Control de Visual Basic .net utilizado para mostrar datos.

Malla Curricular.- Plan de Estudios, Pensum Académico establecido por la subunidad académica.

OptionButton.-Botón de Opción. Pertenece a Visual Basic .net

Sites.- Sitios o lugares de cobertura.

Textbox.- Control de Visual Basic .net utilizado para mostrar e ingresar datos.

Anexo 2.1

CARRERAS NO AUTOFINANCIADAS
ARANCELES DE MATRÍCULA

Para las matriculas de segundo curso en adelante.

Matricula ordinaria para estudiantes nacionales:

CATEGORÍA	
1	US $ 77.00
2	69.00
3	60.00
4	52.00
5	42.00
6	34.00
7	26.00

Las categorías, para el *año* lectivo 2003-2004 se sustentan en las tablas de aranceles que para este mismo ano lectivo estarán autorizados los colegios particulares por las autoridades de educación. ya que en el ano lectivo 2002-2003 se dieron. Varios cambios en el transcurso del mismo. Si el aspirante presenta los recibos de pensiones de octubre, noviembre y diciembre del año 2002. la clasificación será sobre el respectivo valor mensual efectivamente pagado.

Categoría 1, bachilleres de colegios de US $36.01 en delante de pensión mensual.

Categoría 2, bachilleres de colegios entre US $ 28.01 y US $ 36, de pensión mensual.

Categoría 3, bachilleres de colegios entre US $ 20.01 y US $ 28, de pensión mensual.

Categoría 4*,* bachilleres de colegios de US $ 12.01 y US $ 20 de pensión mensual.

Categoría 5, bachilleres de colegios entre US $ 8. 01 y US $ 12, de pensión mensual.

Categoría 6, bachilleres de colegios entre US $ 4.01 y US $ 8, de pensión mensual

Categoría 7, bachilleres de colegios públicos (fiscales u municipales) y particulares gratuitos o de pensión mensual de US $ 4 o menos

Si el estudiante indica que cuando era estudiante de colegio el arancel era menor, deberá presentar prueba de lo pagado en
sexto curso, y el calculo será en base de tipo de cambio al 31 de diciembre del año calendario en que aprobó sexto curso.

Si el estudiante prefiere, en primera matricula, no tomar todas las materias de un curso, sino algunas, puede hacerlo; en ese caso la matricula por cada materia tiene los siguientes valores.

CATEG	
1	US$ 19.20
2	17.20
3	15.00
4	13.00
5	10.50
6	8.50
7	6.50

Si en un ciclo (año o semestre) hay más de 4 materias, se pagará por 4. si se toma de otro curso, se paga la materia adicional.

Si en un año lectivo toma materias de curso diferentes, el estudiante deberá pagar siempre por las materias que sean de curso diferente, además del valor de la matricula por la o las
materia(s) del curso al que pertenece el estudiante.

La repetición de una asignatura es por materia, no incrementa el valor de la materia del año siguiente que cursaría

Si la materia no es tomada en un año lectivo, no causará recargo por repetición cuando se la tome en otro curso lectivo

Para estudiante extranjeros, el valor será de US $ 110 por materia en primera matrícula De 4 o más materias será de US $ 440

Para egresados y/o profesionales nacionales el valor de la matrícula será de US $ 88 y para egresados y/o profesionales extranjeros, el valor será de US $ 440. Por primera vez, cada materia tendrá un calor de US $ 22 para egresados y profesionales y para egresados y profesionales extranjeros será de US $ 110, el principio de las 4

materias por curso se mantiene, así como las demás disposiciones sobre la matrícula de estudiantes nacionales.

El estudiante que se retire de una materia sin ser reprobado en el examen, no tiene derecho a devolución, pero pagará la materia en segunda matrícula con el valor de primera matrícula, si se trata de una sola materia, cuando la tome nuevamente, por una sola vez adicional. Fuera del caso indicado. Para segunda matrícula,
el incremento es del 100% sobre el valor de la primera matrícula, en la Universidad debemos evitar se llegue a terceras matrícula. Un ejemplo, si la primera matricula es de US $ 26.00 la segunda será US $ 52.00 y la tercera $ 104.00.

Sí en un mismo curso repite 4 o más materias, pagará hasta 4, por las de ese curso, a más de pagar por materias la de otros cursos que quisiera tomar.

La matrícula a crédito de hasta 4 pagos, solo es para los estudiantes que tomen de cuatro materias en adelante. Los que tomen menos de 4 materias, deberán pagar en una o en dos cuotas.
La matrícula deberá obtenerse obligatoriamente durante los 30 días de la convocatoria realizada, para el efecto, aun cuando sea a crédito, solicitando en ese plazo el crédito si es el caso.

Los segmentos a crédito de la matrícula causarán un interés del 1 % mensual sobre el saldo.

Para primer curso, el valor de la matricula de estudiante ecuatoriano de primera carrera, será del 50% del valor correspondiente al segundo curso en adelante, si el pre-universitario lo siguió y aprobó en el mismo año con las exoneraciones que considera el instructivo del Curso Pre-universitario.

Cada Facultad debe publicitar en forma adecuada el periodo de matricula

Transcurrido los 30 días del periodo de matricula, se cierra el periodo para el hipotético evento que tenga que reabrirse
matricula en alguna unidad académica, automáticamente habrá un recargo del 30% de su valor Si el incumplimiento de docentes
o empleados administrativos causa el retraso, ese 30 % deberá pagarlo el responsable del retraso.
En los casos de segunda carrera, pero al mismo tiempo, por la segunda pagará el costo que se determine, no menor de US $ 200 sin derecho a exoneraciones en esa segunda carrera.
Para los servidores de la Universidad de Guayaquil y sus hijos no mayores de 25 años, sólo respecto al valor de la matricula, habrá
rebaja del 20% en primera matricula y del 10% en segunda matricula. No habrá rebaja para tercera matricula. Esta rebaja es adicional a la que por otras circunstancias tengan derecho. El tratamiento es especial de este numeral es sólo para primera carrera. El valor de la matricula puede ser apagado en doce mensualidades con orden de descuento. El cupo es de 3 personas por servidor universitario. Se excluye las carreras autofinanciadas, para las que habrá una normativa especial

EXONERACIONES Y REBAJAS
(De pregrado, segundo curso en adelante)

Todas las exoneraciones y rebajas serán en función de los estudiantes nacionales de primera carrera que aprueben el curso
regular, en un año lectivo, es decir, en todas las materias, y en primera matricula. Estas son.

a) 100% del valor de la matricula cuando el promedio de calificaciones del año anterior, en primera matricula fue 9,5 puntos o más

- 75% cuando fue de 9 a 9,49 puntos

- 50% cuando fue de 8,5 a 8,99 puntos
- 25% cuando fue de 8 a 8,49 puntos.

Si el estudiante es servidor universitario, tendrá una rebaja adicional de 25% en cada uno de los supuestos anteriores. Si es hijo de servidor universitario el 10% adicional.

b) 20% si la asistencia a clases programadas fue el 90% en el año anterior; y, 15% por la asistencia entre e! 85% y &' 89,99% de las clases programadas.

c) En el caso de discapacidades, depende de su nivel, exoneración hasta por el 100%.

d) 5% de rebaja para todo estudiante que practique deporte en forma regular, 10% de rebaja para el que representa a las unidades académicas; y, rebajas que van hasta el 50%, dependiendo que sean campeones o vice-campeones universitarios; y, aun cuando no se trate de torneos universitarios, en las competencias provinciales, nacionales e internacionales.

En los casos de discapacidades y de los estudiantes deportistas, puede aplicarse para segunda matricula, pero la aplicación de la rebaja o exoneración dependerá del Rector, siempre bajo el supuesto de aprobación de todas las materias del curso tomadas efectivamente en el año lectivo anterior. También aplicable a primer curso.

e) Rebaja del 5% para los estudiantes que rindan exámenes de primera o segunda llamada, dentro de los horarios regulares y aprueben todas las materias del curso.

f) En caso de pobreza critica, rebaja hasta del 80% del pago de la matricula por primera vez, aun cuando no tenga el estudiante las notas previstas en el literal a). También es aplicable a primer curso. Se requiere informe fundamentado del Departamento de Bienestar Universitario y la decisión del Rector.

Las rebajas son acumulativas, salvo que lleguen al 100% del valor de la matricula.

ARANCELES

Aranceles, con vigencia desde el lunes 20 de Enero del 2003.

Especies Valoradas US$ 0.80

Derecho de copia para estudiantes (a más del costo de las fotocopias, así el caso) US$1.00

Derecho de copia para egresados o profesionales
(a más del costo de las fotocopias, asÍ el caso)
US$4.00

Derecho de duplicado del título profesional o grado académico US$ 7.00

Derecho de pase a otra universidad US$ 44.00

Derecho de pase a otra unidad académica US$ 7.00

Certificado de salud US$5.50

Certificado de egresado US$5.50

Derecho de examen de recuperación (suspenso o aplazado) por materia US$ 2.50

Derecho de equiparación o nivelación

(por materia) el mismo valor de cada materia en
la matricula, según el colegio de origen.

Derecho de solicitud de matrícula a crédito 1.00

Revalidación, homologación o equiparación de título Profesional o grado académico para nacionales US$ 220.00

Revalidación, homologación o equiparamiento de título profesional o grado académico para extranjeros US$ 440.00

Tasa de emisión y liquidación del impuesto dos por mil US$ 3.00

Derecho de participación en concurso para
profesores universitarios US$ 16.00

Derecho de certificado final para aux. de enfermería US$ 16.00

DERECHO DE TÍTULOS PARA NACIONALES

a. Titulo de pregrado US $ 68.00

b. Otro, segundo o posterior titulo de pregrado
US $83.00

c. Títulos de postgrado
- Diplomado y especialidad US $ 104.00
- Maestría US $ 32.00
- Doctorado US $ 660.00

DERECHO DE TÍTULOS PARA EXTRANJEROS

a. Titulo de pregrado US $ 412.00

b. Otro, segundo o posterior titulo de pregrado
US $ 462.00

c. Títulos de postgrado
- Diplomado y especialidad US $ 660.00
- Maestría US $ 770.00

***Derecho de titulo**:* Quienes hayan cursado en el año lectivo 2002-2003 o se encuentren cursando el último curso o semestre de pregrado, postgrado, diplomado, especialidad y de maestría en la Universidad de Guayaquil, que paguen los derechos al 31 de marzo del 2002, solo deberán cubrir el valor que estuvo vigente en la tabla anterior, es decir el año 2002.

Los derechos podrán pagarse aun cuando el 31 de marzo no hayan cumplido el año de estudios o rendido los exámenes, declarándose el pago provisional.

Anexo 2.2

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
ESCUELA DE INGENIERIA CIVIL
2002-2003

FOLIO..... MATRICULA.....
CATEGORIA DEL COLEGIO..... 4
PROMEDIO AÑO ANTERIOR..... 8,57
FORMA DE PAGO: (CONTADO) (CREDITO)

Señor Estudiante
Sírvase llenar el siguiente **BANCO DE MATERIAS**, con todos los datos que se solicitan:

APELLIDOS Y NOMBRES... AYUQUINA CARABAJO WALTER RAMIRO

LUGAR DE NACIMIENTO... Guayaquil (Ciudad) Guayas (Provincia) FECHA: 26 de Marzo de 1983

CEDULA DE CUIDADANIA... 0921916862 TITULO DE GRADO... FIMA

GRADUADO EN EL COLEGIO... Academia Naval Almirante Illingworth (Particu
(Indicar si es Nacional, Particular u otros)

PROVINCIA... Guayas ...EN FECHA... 21 de Enero del 2001 CON NOTA... 18

NOMBRE DEL PADRE... Jose Ayuquina ...NOMBRE DE LA MADRE: Rosa Carabajo

DOMICILIO... Floresta II Mz 191 V16 ...TELEFONO... 2439770

LUGAR DE TRABAJO... ...TELEFONO...

CODIGO	NOMBRE DE LA ASIGNATURA	VEZ	TOMO	GRUPO	SUBGRUPO
201	Dibujo Técnico II Computarizado	1era.	Si		
202	Matemáticas II	1era.	Si		
203	Topografía II	1era.	Si		
204	Física II	1era.	Si		
205	Estática	1era.	Si		
206	Geología	1era.	Si		
207	Física II Laboratorio	1era.	Si		
208	Hidráulica I	1era.	Si		
210	Computación II	1era.	Si		
	Cobrar Módulo II de Idiomas		Si		

SUSPENSOS:
ARRASTRES:

Elaborado por Eleana Revisado por ... Entregado por ... Recibido por ME

Walter Ayuquina
FIRMA DEL ESTUDIANTE

Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Físicas
SECRETARIO

Anexo 2.3

Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Físicas
Carrera de Ingeniería de Sistemas Computacionales

Cuadro de Valores de Matrículas para el I Ciclo del Periodo Lectivo 2002 - 2003 por Categoría de colegios

Matrícula global tomando 4 o más Asignaturas del mismo nivel

- Los estudiantes que tomen Asignaturas de Primero a Cuarto Semestre deberán cancelar en la PRIMERA CUOTA el 40% del valor total de la matrícula y el 60% restante en 3 cuotas de igual valor. Además se le sumará a la primera cuota los recargos por vez a tomar de la(s) asignatura(s), derecho de examen de recuperación del semestre anterior ($2.50 por asignatura), el valor del carnet
  $ 1.70 (para ler. Semestre y para aquellos alumnos que se caduca el mismo, durante el semestre), $ 6.00 por uso libre de Internet y por valor de especie valorada $ 1.00.

Categoría	Valor Global de la Matrícula	1 ra. Cuota 40%	3 cuotas 20%
1	147.00	58.80	29.40
2	142.80	57.12	28.56
3	134.40	53.76	26.88
4	130.20	52.08	26.04
5	126.00	50.40	25.20
6 y 7	117.60	47.04	23.52

- Los estudiantes que toman asignaturas de Quinto Semestre en adelante pagarán el valor de la matrícula en 5 cuotas de 20% cada una. Además se le sumará a la primera cuota los recargos por vez
  a tomar de la(s) asignatura(s), derecho de examen de recuperación del semestre anterior ($2.50 por
  asignatura), el valor del carnet $ 1.70 (para los alumnos que se caduca el mismo, durante el semestre), $ 6.00 por uso libre de Internet y por valor de especie valorada $ 1.00.

Categoría	Valor Global de la Matrícula	1ra. Cuota 20%	4 cuotas 20%
1	147.00	29.40	29.40
2	142.80	28.56	28.56
3	134.40	26.88	26.88
4	130.20	26.04	26.04
5	126.00	25.20	25.20

6 y 7	**117.60**	**23.52**	**23.52**

• Valor por Asignatura a cursar por primera vez. Menos de 4 asignaturas del mismo nivel ó de diferente nivel

NO TIENE RECARGO ALGUNO

Categoría	Valor por Asignaturas
1	**36,75**
2	**35.70**
3	**33.60**
4	**32.55**
5	**31.50**
6 y 7	**29.40**

VALOR POR ASIGNATURAS A CURSAR POR SEGUNDA VEZ

Recargo de 50%, I ciclo 2002 - 2003

Categoría	Valor por Asignaturas	Valor del Recargo 50%	Total
1	**36.75**	**18.37**	**55.12**
2	**35.70**	**17.85**	**53.55**
3	**33.60**	**16.80**	**50.40**
4	**32.55**	**16.27**	**48.82**
5	**31.50**	**15.75**	**47.25**
6 y 7	**29.40**	**14.70**	**44.10**

VALOR POR ASIGNATURAS A CURSAR POR TERCERA Y ÚLTIMA VEZ

Recargo del 150%, I ciclo 2002 - 2003

Categoría	Valor por Asignaturas	Valor del Recargo 150%	Total
1	36.75	55.12	91.87
2	35.70	53.55	89.25
3	33.60	50.40	84.00
4	32.55	48.82	81.37
5	31.50	47.25	78.75
6 y 7	29.40	44.10	73.50

VALOR POR ASIGNATURAS A CURSAR POR CUARTA VEZ
(Casos por excepción, considerados por el Consejo Universitario)

Recargo del 250%, I ciclo 2002 – 2003

Categoría	Valor por Asignaturas	Valor del Recargo 250%	Total
1	36.75	91.87	128.62
2	35.70	89.25	124.95
3	33.60	84.00	117.60
4	32.55	81.37	113.92
5	31.50	78.75	110.25
6 y 7	29.40	73.50	102.90

EGRESADOS Y/O PROFESIONALES NACIONALES

Ciclo I del Período Lectivo 2002-2003

Matrícula global tomando 4 o más asignaturas por primera vez del mismo nivel. $ **152.00**

Valor por asignatura a cursar por primera vez, del mismo nivel ó de diferente nivel $ **38.00**

EXTRANJEROS

Bachilleres, matrícula global tomando 4 o más asignaturas por primera vez del mismo nivel. $ **360.00**

Bachilleres, valor por asignatura a cursar por primera vez, del mismo nivel ó de diferente nivel. $ **90.00** c/u

Egresados y/o Profesionales,
Matrícula global tomando 4 o más asignaturas por primera vez del mismo nivel. $ **400.00**

Egresados y/o Profesionales,
Valor por asignatura a cursar por primera vez, del mismo nivel ó de diferente nivel. $ **100.00**

El valor por asignatura a cursar por segunda vez tendrá un recargo del 50% por tercera vez el recargo es 150% y por cuarta vez 250%.

En la PRIMERA CUOTA los estudiantes deberán cancelar el 40% del valor total de la matrícula y el 60% restante en 3 cuotas de igual valor. Además se le sumará a la primera cuota los recargos por vez a tomar de la(s) asignatura(s), derecho de examen de recuperación del Semestre anterior ($ 2.50 por asignatura), el valor del carnet $ 1.70 (para ler. Semestre y para aquellos alumnos que se les caduca durante el semestre), $ 6.00 por uso libre de Internet y por valor de especie valorada $ 1.00.

BIBLIOGRAFIA

- www.monografias.com
- www.lawebdelprogramador.com
- http://www.willydev.net/DesktopDefault.Aspx
- http://RustemSoft.com/DataGridColumns.htm
- Microsoft Visual Studio .net Professional

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

"Implementación de la Estructura del Sistema Académico con el Módulo de Matriculación para la Facultad de Ciencias Matemáticas y Físicas"

MANUAL TECNICO

Autores:

AYUQUINA CARABAJO MARTHA ELIZABETH
ARELLANO GUZMAN GREGORIO ALEJANDRO
CRUZ OCHOA WILSON
ZAMBRANO MOLINA DARWIN FLORENCIO

GUAYAQUIL – ECUADOR

Año: 2004

INDICE GENERAL

INTRODUCCION 1

CAPITULO 1

1. MODELO ESTRUCTURAL DEL SISTEMA 3

1.1. Diagrama Entidad-Relación 3

1.2. Detalle Descriptivo de las Tablas 3

1.3. Descripción Técnica de las Pantallas del Sistema 32

1.3.1. Pantallas de Acceso 32

1.3.2. Pantallas de Mantenimiento 33

1.3.3. Pantallas de Control Académico 34

1.3.4. Pantallas de Períodos Lectivos 35

1.3.5. Pantallas de Control de Horarios 35

1.3.6. Pantallas de Inscripción y Matriculación 35

1.3.7. Pantallas de Registro de Calificaciones 36

1.3.8. Pantallas de Registro de Solicitudes 37

1.3.9. Pantallas de Reportes 37

1.3.10 Pantallas de Seguridad y Administración 38

CAPITULO 2

2. DICCIONARIO DE DATOS 39

2.1. Tablas de la Base de Datos 39

2.2. Scripts de la Creación de las Tablas 80

2.3. Stored Procedures 99

2.4. Scripts y Triggers para Auditoria del Sistema 193

CAPITULO 3

3. REQUERIMIENTOS TECNICOS DEL SISTEMA 210

3.1. Requerimientos de Hardware 211

3.2. Requerimientos de Software 211

3.3. Requerimientos Generales 211

CAPITULO 4

4. INSTALACION DE COMPONENTES 216

4.1. Instalación de SQL Server en el Servidor 216

4.2. Registro de Servidor SQL 234

4.3. Creación de la base de datos en SQL Server 244

4.4. Creación de las tablas en la base de datos 252

4.5. Creación de Stored Procedures 255

4.6. Creación de las tablas de auditoría y triggers 255

4.7. Creación del usuario ADM_SYS 256

4.8. Instalación de la aplicación en el cliente 260

4.9. Precaución con el archivo Base_Acad.ini 267

CAPITULO 5

5. MIGRACION 269

5.1. Herramientas para exportación de datos 269

5.2. Tablas proporcionadas para migración 270

5.3. Depuración de datos 271

5.4. Importación de datos 290

CAPITULO 6

6. COPIAS DE SEGURIDAD BACKUP 294

6.1. Copia de seguridad de SQL Server 2000 294

6.2. Pasos para realizar backup de la base de datos 295

CAPITULO 7

7. ADMINISTRACION DE USUARIOS Y ACCESOS 308

7.1. Acceso a las pantallas 308

7.2. Mantenimiento de perfiles 309

7.3. Mantenimiento de pantallas 310

7.4. Mantenimiento de usuarios 311

7.5. Mantenimiento de permisos 313

INDICE DE FIGURAS

Figura 1.1. Diagrama Entidad – Relación 3

Figura 3.1. Configuración Regional – Números 212

Figura 3.2. Configuración Regional- Moneda 213

Figura 3.3. Configuración Regional- Fecha 214

Figura 4.1. Menú Inicial para instalación de SQL Server 217

Figura 4.2. Instalar componentes 219

Figura 4.3. Pantalla inicial del instalador 220

Figura 4.4. Nombre del equipo 221

Figura 4.5. Selección de instalación 222

Figura 4.6. Información del usuario 223

Figura 4.7. Contrato de licencia del software 224

Figura 4.8. Definición de la instalación 225

Figura 4.9. Nombre de la instancia 226

Figura 4.10. Tipo de instalación 227

Figura 4.11. Cuentas de servicio 228

Figura 4.12. Modo de autenticación 229

Figura 4.13. Iniciar la copia de archivos 230

Figura 4.14. Fin de la instalación 232

Figura 4.15. Menú Microsoft SQL Server 232

Figura 4.16. Administrador de servicios SQL-Detenido 233

Figura 4.17. Administrador de servicios SQL-Ejecución 234

Figura 4.18. SQL Server Enterprise Manager 235

Figura 4.19. Asistente para registro de servidor SQL 236

Figura 4.20. Seleccionar un servidor SQL 237

Figura 4.21. Agregar un servidor SQL 238

Figura 4.22. Seleccionar un modo de autenticación 239

Figura 4.23. Seleccionar una opción de conexión 240

Figura 4.24. Seleccionar un grupo de servidores SQL 241

Figura 4.25. Asistente registro de servidor-Finalizado 242

Figura 4.26. Nuevo servidor SQL –Sistema Academico 243

Figura 4.27. Conectar a SQL Server 244
Figura 4.28. Creación de una base de datos 246
Figura 4.29. Propiedades de la nueva base de datos 247
Figura 4.30. Base de datos Universidad-General 248
Figura 4.31. Base de datos Universidad-Arch. de datos 249
Figura 4.32. Base de datos Universidad-Transacciones 251
Figura 4.33. Analizador de consultas SQL 253
Figura 4.34. Selección de bases de datos 253
Figura 4.35. Script de creación de tablas 254
Figura 4.36. Nuevo inicio de sesión 256
Figura 4.37. Propiedades de inicio de sesión 257
Figura 4.38. Inicio de sesión-Autenticación 258
Figura 4.39. Inicio de sesión-Funciones del servidor 259
Figura 4.40. Inicio de sesión-Acceso a bases de datos 260
Figura 4.41. Archivo de instalación Sistema Academico 262
Figura 4.42. Pantalla inicial del instalador del sistema 263
Figura 4.43. Seleccionar carpeta de instalación 264
Figura 4.44. Confirmar la instalación del producto 265
Figura 4.45. Instalando el Sistema Académico 266

Figura 4.46. Instalación completada 267

Figura 4.47. Archivo Base_Acad.ini 268

Figura 5.1. Asistente para la importación/exportación 290

Figura 5.2. Seleccionando origen de datos 291

Figura 5.3. Seleccionando el destino de datos 291

Figura 5.4. Seleccionando una opción de copia 292

Figura 5.5. Señalando los archivos a migrar 292

Figura 5.6. Ejecutar la migración 293

Figura 5.7. Iniciar el proceso de migración 293

Figura 6.1. Acceso a la configuración del backup 294

Figura 6.2. Acceso directo del Administrador corporat. 295

Figura 6.3. Pantalla SQL server Enterprise Manager 295

Figura 6.4. Servidor de la base de datos 296

Figura 6.5. Acceso al backup 297

Figura 6.6. Ficha general del backup 297

Figura 6.7. Ficha de modificación de programación 300

Figura 6.8. Ficha de modificación periódica 302

Figura 6.9. Ficha opciones del backup (1) 304

Figura 6.10. Ficha opciones del backup (2) 307

Figura 7.1. Acceso al menú Administración de usuarios 308

Figura 7.2. Mantenimiento de perfiles-Pantalla inicial 309

Figura 7.3. Mantenimiento de perfiles-Nuevo perfil 310

Figura 7.4. Mantenimiento de perfiles-Perfiles creados 310

Figura 7.5. Mantenimiento de pantallas 311

Figura 7.6. Creación de un nuevo usuario del sistema 312

Figura 7.7. Consulta de un usuario existente 312

Figura 7.8. Mantenimiento de permisos 313

INDICE DE TABLAS

Tabla 1.1. Tablas que se relacionan con TBMG_PAIS 3

Tabla 1.2. Tablas que se relacionan con TBMG_PROVINCIA 4

Tabla 1.3. Tablas que se relacionan con TBMG_CIUDAD 4

Tabla 1.4. Tablas que se relacionan con TBMG_UNIVERSIDA 5

Tabla 1.5. Tablas que se relacionan con TBMA_FACULTAD 5

Tabla 1.6. Tablas que se relacionan con TBMA_TIPO_SUBU 6

Tabla 1.7. Tablas que se relacionan con TBMA_SUBUN_ACAD 7

Tabla 1.8. Tablas que se relacionan con TBMA_AREA_ACAD 8

Tabla 1.9. Tablas que se relacionan con TBMA_ASIGNATURA 8

Tabla 1.10. Tablas que se relacionan con TBMA_PÉNSUM 9

Tabla 1.11.Tablas que se relacionan con TBMA_MOTCA_PENS 9

Tabla 1.12. Tablas que se relacionan con TBMA_PREREQUISI 10

Tabla 1.13. Tablas que se relacionan con TBMD_TITUL_UNIV 10

Tabla 1.14. Tablas que se relacionan con TBMD_DOCENTE 11

Tabla 1.15. Tablas que se relacionan con TBME_ESPECIALIZ 12

Tabla 1.16. Tablas que se relacionan con TBME_FOTO_ESTU 12

Tabla 1.17. Tablas que se relacionan con TBME_ESTUDIANTE 13

Tabla 1.18. Tablas que se relacionan con TBMI_TIPO_CURS 13

Tabla 1.19. Tablas que se relacionan con TBMI_DOCEN_ASIG 14

Tabla 1.20. Tablas que se relacionan con TBMI_HORAR_GRUP 14

Tabla 1.21. Tablas que se relacionan con TBMI_ESTUD_SUBU 15

Tabla 1.22. Tablas que se relacionan con TBMI_ESTUD_ASIG 16

Tabla 1.23. Tablas que se relacionan con TBMI_ESTUD_NIVE 17

Tabla 1.24. Tablas que se relacionan con TBMI_HORAR_ESTU 17

Tabla 1.25. Tablas que se relacionan con TBMI_ESTUD_PENS 18

Tabla 1.26. Tablas que se relacionan con TBMM_CATEGORÍA 19

Tabla 1.27. Tablas que se relacionan con TBMM_COLEGIO 19

Tabla 1.28. Tablas que se relacionan con TBMM_MATRI_GLOB 20

Tabla 1.29. Tablas que se relacionan con TBMM_MATRI_ASIG 20

Tabla 1.30. Tablas que se relacionan con TBMM_INCRE_MATR 21

Tabla 1.31.Tablas que se relacionan con TBMM_DESCU_ARAN 21

Tabla 1.32.Tablas que se relacionan con TBMM_CONCE_COBR 22

Tabla 1.33. Tablas que se relacionan con TBMM_BANCO 22

Tabla2.34.Tablas que se relacionan con TBMM_ORDEN_PAGO 23

Tabla 1.35.Tablas que se relacionan con TBMM_DETAL_MATR 24

Tabla 1.36. Tablas que se relacionan con TBMM_CUOTA 24

Tabla 1.37. Tablas que se relacionan con TBMM_AUTORIDAD 25

Tabla 1.38. Tablas que se relacionan con TBMM_PERIO_LECT 25

Tabla 1.39. Tablas que se relacionan con TBMC_TIPO_CALI 26

Tabla 1.40. Tablas que se relacionan con TBMC_CALIF_PARC 27

Tabla 1.41. Tablas que se relacionan con TBMC_CALIF_PROM 27

Tabla 1.42. Tablas que se relacionan con TBMC_ENTRE_CALI 28

Tabla 1.43. Tablas que se relacionan con TBMC_TIPO_SOLI 28

Tabla 1.44. Tablas que se relacionan con TBMC_SOLIC_ESPE 29

Tabla 1.45. Tablas que se relacionan con TBMC_SOLIC_ESPE 30

Tabla 1.46. Tablas que se relacionan con TBMC_SOLIC_ESPE 30

Tabla 1.47. Tablas que se relacionan con TBMS_PANTALLA 31

Tabla 1.48. Tablas que se relacionan con TBMS_PERFIL 31

Tabla 1.49. Tablas que se relacionan con TBMS_PERMISO 32

Tabla 1.50. Tablas que se relacionan con TBMS_USUARIO 32

Tabla 1.51. Descripción Técnica de la Pantalla de Acceso 32

Tabla 1.52. Descripción Técnica de las Pantallas de Mantenimiento (1) 33

Tabla 1.53. Descripción Técnica de las Pantallas de Mantenimiento (2) 34

Tabla 1.54. Descripción Técnica de las Pantallas de Control 34

Tabla 1.55. Descripción Técnica de las Pantallas de Períodos Lectivos 35

Tabla 1.56. Descripción Técnica de las Pantallas de Control de Horarios 35

Tabla 1.57. Descripción Técnica de las Pantallas de Inscripción y Matriculación (1) 35

Tabla 1.58. Descripción Técnica de las Pantallas de Inscripción y Matriculación (2) 36

Tabla 1.59. Descripción Técnica de las Pantallas de Registro de Calificaciones 36

Tabla 1.60. Descripción Técnica de las Pantallas de Solicitudes 37

Tabla 1.61. Descripción Técnica de las Pantallas de Reportes 37

Tabla 1.62. Descripción Técnica de las Pantallas de Seguridad 38

Tabla 2.1. Campos de la Tabla TBMA_FACULTAD 39

Tabla 2.2. Campos de la Tabla TBMA_SUBUN_ACAD 40

Tabla 2.3. Campos de la Tabla TBMA_TIPO_SUBU 41

Tabla 2.4. Campos de la Tabla TBMA_AREA_ACAD 41

Tabla 2.5. Campos de la Tabla TBMA_PREREQUISI 42

Tabla 2.6. Campos de la Tabla TBMA_MOTCA_PENS 42

Tabla 2.7. Campos de la Tabla TBMA_ASIGNATURA 43

Tabla 2.8. Campos de la Tabla TBMA_ PENSUM 44

Tabla 2.9. Campos de la Tabla TBMC_TIPO_CALI 45

Tabla 2.10. Campos de la Tabla TBMC_CALIF_PROM 46

Tabla 2.11. Campos de la tabla TBMC_CALIF_PARC (1) 47

Tabla 2.12. Campos de la tabla TBMC_CALIF_PARC (2) 48

Tabla 2.13. Campos de la Tabla TBMC_TIPO_SOLI 48

Tabla 2.14. Campos de la Tabla TBMC_SOLIC_ESPE 49

Tabla 2.15. Campos de la Tabla TBMD_DOCENTE (1) 50

Tabla 2.16. Campos de la Tabla TBMD_DOCENTE (2) 51

Tabla 2.17. Campos de la Tabla TBMD_TITUL_UNIV 51

Tabla 2.18. Campos de la Tabla TBME_ESTUDIANTE (1) 52

Tabla 2.19. Campos de la Tabla TBME_ESTUDIANTE (2) 53

Tabla 2.20. Campos de la Tabla TBME_ESTUDIANTE (3) 54

Tabla 2.21. Campos de la Tabla TBME_ESPECIALIZ 54

Tabla 2.22. Campos de la Tabla TBME_FOTO_ESTU 55

Tabla 2.23. Campos de la Tabla TBMG_UNIVERSIDA (1) 55

Tabla 2.24. Campos de la Tabla TBMG_UNIVERSIDA (2) 56

Tabla 2.25. Campos de la Tabla TBMG_PAIS 56

Tabla 2.26. Campos de la Tabla TBMG_CIUDAD 57

Tabla 2.27. Campos de la Tabla TBMG_PROVINCIA 57

Tabla 2.28. Campos de la Tabla TBMI_DOCEN_ASIG 58

Tabla 2.29. Campos de la Tabla TBMI_TIPO_CURS (1) 59

Tabla 2.30. Campos de la Tabla TBMI_TIPO_CURS (2) 60

Tabla 2.31. Campos de la Tabla TBMI_ESTUD_PENS 60

Tabla 2.32. Campos de la Tabla TBMI_HORAR_GRUP 61

Tabla 2.33. Campos de la Tabla TBMI_ESTUD_ASIG (1) 62

Tabla 2.34. Campos de la Tabla TBMI_ESTUD_ASIG (2) 63

Tabla 2.35. Campos de la Tabla TBMI_ESTUD_SUBU (1) 63

Tabla 2.36. Campos de la Tabla TBMI_ESTUD_SUBU (2) 64

Tabla 2.37. Campos de la Tabla TBMI_ESTUD_NIVE (1) 65

Tabla 2.38. Campos de la Tabla TBMI_ESTUD_NIVE (2) 66

Tabla 2.39. Campos de la Tabla TBMI_HORAR_ESTU 66

Tabla 2.40. Campos de la Tabla TBMM_MATRI_ASIG 67

Tabla 2.41. Campos de la Tabla TBMM_AUTORIDAD 68

Tabla 2.42. Campos de la tabla TBMM_PERIO_LECT (1) 68

Tabla 2.43. Campos de la tabla TBMM_PERIO_LECT (2) 69

Tabla 2.44. Campos de la tabla TBMM_MATRI_GLOB 70

Tabla 2.45. Campos de la tabla TBMM_CATEGORIA 70

Tabla 2.46. Campos de la tabla TBMM_DETAL_MATR 71

Tabla 2.47. Campos de la tabla TBMM_DESCU_ARAN 72

Tabla 2.48. Campos de la tabla TBMM_COLEGIO (1) 72

Tabla 2.49. Campos de la tabla TBMM_COLEGIO (2) 73

Tabla 2.50. Campos de la tabla TBMM_ORDEN_PAGO 73

Tabla 2.51. Campos de la tabla TBMM_CUOTA 74

Tabla 2.52. Campos de la tabla TBMM_INCRE_MATR 75

Tabla 2.53. Campos de la tabla TBMS_PERFIL 75

Tabla 2.54. Campos de la tabla TBMS_PERMISO 76

Tabla 2.55. Campos de la tabla TBMS_USUARIO 76

Tabla 2.56. Campos de la tabla TBMS_PANTALLA 77

Tabla 2.57. Campos de la tabla TBMN_TIPO_PROY 77

Tabla 2.58. Campos de la tabla TBMN_ PROYE_GRAD 78

Tabla 2.59. Campos de la tabla TBMM_CONCE_COBR 78

Tabla 2.60. Campos de la tabla TBMM_BANCO 79

Tabla 2.61. Campos de la tabla TBMM_ENTRE_CALI 79

Tabla 5.1. Datos inconsistentes de la Tabla Colegio (1) 272

Tabla 5.2. Datos inconsistentes de la Tabla Colegio (2) 273

Tabla 5.3. Datos inconsistentes de la Tabla País 274

Tabla 5.4. Datos no encontrados en la Tabla Estudiante (1) 275

Tabla 5.5. Datos no encontrados en la Tabla Estudiante (2) 276

Tabla 5.6. Datos inconsistentes de la Tabla Estudiante 277

Tabla 5.7. Datos de la tabla Especialización (1) 277

Tabla 5.8. Datos de la tabla Especialización (2) 278

Tabla 5.9. Datos de la tabla Especialización (3) 279

Tabla 5.10. Datos de la tabla Especialización (4) 280

Tabla 5.11. Datos de la tabla Especialización (5) 281

Tabla 5.12. Datos no encontrados en la Tabla Docente (1) 282

Tabla 5.13. Datos no encontrados en la Tabla Docente (2) 283

Tabla 5.14. Datos no encontrados en la Tabla Asignatura 284

Tabla 5.15. Datos no encontrados en la Tabla Estud-Subu 287

Tabla 5.16. Datos no encontrados en la Tabla Estud-Asignat 288

Tabla 5.17. Datos no encontrados en la Tabla Estud-Nivel (1) 289

Tabla 5.18. Datos no encontrados en la Tabla Estud-Nivel (2) 290

INTRODUCCION

Actualmente la informática ha incursionado en el mercado del Desarrollo de Sistemas para automatizar la información facilitando el trabajo del hombre en diversas áreas; por esta razón muchas empresas ahorran tiempo y espacio utilizando sistemas informáticos en áreas como: Contabilidad, Tesorería, Recursos humanos, Servicios al cliente, etc. Este proyecto cumple las necesidades de la Facultad de Ciencias Matemáticas y Físicas para la matriculación de los estudiantes, manteniendo la información segura y agilitando dicho proceso para el bienestar del elemento más importante de una institución educativa: el cuerpo estudiantil.

Este sistema fue desarrollado sobre una plataforma Windows con herramienta Visual Studio.Net e interactúa de manera directa con la base de datos SQL Server. Las herramientas utilizadas nos han brindado soportes para implementar la estructura que más adelante se detalla, a más de manejar una base de datos sólida en la cual se pueden aplicar restricciones y permisos para cada usuario que se conectan por medio de la red.

El Sistema que presentamos está basado en parámetros con la finalidad de poder adaptarse a las necesidades que se presenten en la Facultad de

Ciencias Matemáticas y Físicas y pueda acoplarse fácilmente a cualquier otra unidad académica.

CAPITULO 1

1. MODELO ESTRUCTURAL DEL SISTEMA

1.1. Diagrama Entidad-Relación

El modelo ha quedado de la forma como se presenta en la figura 1.1, incorporando parámetros para lo más importante del sistema.

1.2. Detalle Descriptivo de las tablas

- **TBMG_PAIS**

  Esta tabla guardará información sobre cada país. Su clave primaria es el código del país → Pais_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMG_PROVINCIA	Prov_Pais

Tabla 1.1. Tablas que se relacionan con TBMG_PAIS

- **TBMG_PROVINCIA**

Esta tabla guardará información sobre cada provincia. Su clave primaria está formada por el código de la provincia → Prov_Codigo y el código del país → Prov_Pais. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMG_PAIS	Pais_Codigo
TBMG_CIUDAD	Ciud_Provincia

Tabla 1.2. Tablas que se relacionan con TBMG_PROVINCIA

- **TBMG_CIUDAD**

Esta tabla guardará información sobre cada ciudad. Su clave primaria está formada por el código de la ciudad → Ciud_Codigo, el código de la provincia → Ciud_Provincia y el código del país → Ciud_Pais. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMG_PROVINCIA	Prov_Codigo

Tabla 1.3. Tablas que se relacionan con TBMG_CIUDAD

- **TBMG_UNIVERSIDA**

Esta tabla guardará información sobre cada universidad. Su clave primaria es el código de la universidad → Univ_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMD_DOCENTE	Doce_Universida
TBME_ESTUDIANTE	Estu_Universida

Tabla 1.4. Tablas que se relacionan con TBMG_UNIVERSIDA

- **TBMA_FACULTAD**

Esta tabla guardará información sobre cada facultad. Su clave primaria es el código de la facultad → Facu_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Facultad
TBMD_DOCENTE	Doce_Codigo

Tabla 1.5. Tablas que se relacionan con TBMA_FACULTAD

- **TBMA_TIPO_SUBU**

Esta tabla guardará información sobre los tipos de subunidades. Su clave primaria es el código del tipo de subunidad → TiSu_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Tipo

Tabla 1.6. Tablas que se relacionan con TBMA_TIPO_SUBU

- **TBMA_SUBUN_ACAD**

Esta tabla guardará información sobre cada subunidad académica. Su clave primaria es el código de la subunidad → SuAc_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_TIPO_SUBU	TiSu_Codigo
TBMM_INCRE_MATR	InMa_Subunidad
TBMA_FACULTAD	Facu_Codigo
TBMD_DOCENTE	Doce_Codigo
TBMM_MATRI_ASIG	MaAs_Subunidad
TBMM_MATRI_GLOB	MaGl_Subunidad
TBMA_AREA_ACAD	ArAc_Subunidad
TBMI_TIPO_CURS	TiCu_Subunidad
TBMA_ASIGNATURA	Asig_Subunidad
TBMA_PENSUM	Pens_Subunidad
TBMC_TIPO_CALI	TiCa_Subunidad
TBMI_ESTUD_SUBU	EsSu_Subunidad
TBMN_PROYE_GRAD	PrGr_Subunidad
TBMM_AUTORIDAD	Auto_Subunidad

Tabla 1.7. Tablas que se relacionan con TBMA_SUBUN_ACAD

- **TBMA_AREA_ACAD**

Esta tabla guardará información sobre cada área académica. Su clave primaria está formada por el código del área académica → ArAc_Codigo y el código de la subunidad → ArAc_Subunidad. La

relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMD_DOCENTE	Doce_Codigo

Tabla 1.8. Tablas que se relacionan con TBMA_AREA_ACAD

- **TBMA_ASIGNATURA**

Esta tabla guardará información sobre cada asignatura. Su clave primaria es el código de la asignatura → Asig_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMI_DOCEN_ASIG	DoAs_Asignatura
TBMI_HORAR_GRUP	HoGr_Asignatura
TBMI_ESTUD_ASIG	EsAs_Asignatura
TBMA_PREREQUISI	Prer_Asignatura Prer_Requisito

Tabla 1.9. Tablas que se relacionan con TBMA_ASIGNATURA

- **TBMA_PENSUM**

Esta tabla guardará información sobre cada pénsum. Su clave primaria es el código del pénsum → Pens_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMI_ESTUD_PENS	EsPe_Pensu_Actu EsPe_Pensu_Ante
TBMA_PREREQUISI	Prer_Pensum

Tabla 1.10. Tablas que se relacionan con TBMA_PÉNSUM

- **TBMA_MOTCA_PENS**

Esta tabla guardará información sobre los motivos de cambio de pénsum. Su clave primaria es el código del motivo de cambio de pénsum → MoPe_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMI_ESTUD_PENS	EsPe_MotCa_Pens

Tabla 1.11. Tablas que se relacionan con TBMA_MOTCA_PENS

- **TBMA_PREREQUISI**

Esta tabla guardará información sobre cada prerrequisito. No tiene clave primaria. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_ASIGNATURA	Asig_Codigo
TBMA_PÉNSUM	Pens_Codigo

Tabla 1.12. Tablas que se relacionan con TBMA_PREREQUISI

- **TBMD_TITUL_UNIV**

Esta tabla guardará información sobre los títulos universitarios. Su clave primaria es el código del título universitario → TiUn_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMD_DOCENTE	Doce_Titul_Univ

Tabla 1.13. Tablas que se relacionan con TBMD_TITUL_UNIV

- **TBMD_DOCENTE**

Esta tabla guardará información sobre los docentes. Su clave primaria es el código del docente → Doce_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_FACULTAD	Facu_Decano
TBMA_SUBUN_ACAD	SuAc_Director
TBMA_AREA_ACAD	ArAc_Coordinado
TBMI_DOCEN_ASIG	DoAs_Docente
TBMG_UNIVERSIDA	Univ_Codigo
TBMD_TITUL_UNIV	TiUn_Codigo

Tabla 1.14. Tablas que se relacionan con TBMD_DOCENTE

- **TBME_ESPECIALIZ**

Esta tabla guardará información sobre cada especialización. Su clave primaria es el código de la especialización → Espe_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBME_ESTUDIANTE	Estu_Especializ

Tabla 1.15. Tablas que se relacionan con TBME_ESPECIALIZ

- **TBME_FOTO_ESTU**

Esta tabla guardará la foto de cada estudiante. Su clave primaria es el código del estudiante → FoEs_Estudiante. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBME_ESTUDIANTE	Estu_Codigo

Tabla 1.16. Tablas que se relacionan con TBME_FOTO_ESTU

- **TBME_ESTUDIANTE**

Esta tabla guardará información sobre los estudiantes. Su clave primaria es el código del estudiante → Estu_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBME_ESPECIALIZ	Espe_Codigo
TBME_FOTO_ESTU	FoEs_Estudiante
TBMM_COLEGIO	Cole_Codigo
TBMG_UNIVERSIDA	Univ_Codigo
TBMI_ESTUD_SUBU	EsSu_Estudiante

Tabla 1.17. Tablas que se relacionan con TBME_ESTUDIANTE

- **TBMI_TIPO_CURS**

Esta tabla guardará información sobre cada tipo de curso. Su clave primaria es el código del tipo de curso → TiCu_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMM_PERIO_LECT	PeLe_Tipo_Curs
TBMI_ESTUD_NIVE	EsNi_Tipo_Curs

Tabla 1.18. Tablas que se relacionan con TBMI_TIPO_CURS

- **TBMI_DOCEN_ASIG**

Esta tabla guardará información sobre las cargas horarias de un docente. No tiene clave primaria.

La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMD_DOCENTE	Doce_Codigo
TBMA_ASIGNATURA	Asig_Codigo

Tabla 1.19. Tablas que se relacionan con TBMI_DOCEN_ASIG

- **TBMI_HORAR_GRUP**

Esta tabla guardará información sobre los horarios de los grupos. Cabe recalcar que esta tabla es temporal y se borra cuando se cierra un periodo lectivo. Su clave primaria está formada por el código del horario → HoGr_Codigo y el código del grupo → HoGr_Grupo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_ASIGNATURA	Asig_Codigo
TBMI_HORAR_ESTU	HoEs_Horar_Grup HoEs_Grupo

Tabla 1.20. Tablas que se relacionan con TBMI_HORAR_GRUP

- **TBMI_ESTUD_SUBU**

Esta tabla guardará información sobre el estudiante en la subunidad. Su clave primaria es el código del estudiante dentro de esa subunidad (4 dígitos de la subunidad + 6 dígitos del estudiante) → EsSu_CodEs_Subu. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMI_ESTUD_ASIG	EsAs_Estud_Subu
TBMI_HORAR_ESTU	HoEs_Estud_Subu
TBMI_ESTUD_PENS	EsPe_Estud_Subu
TBMI_ESTUD_NIVE	EsNi_Estud_Subu
TBME_ESTUDIANTE	Estu_Codigo

Tabla 1.21. Tablas que se relacionan con TBMI_ESTUD_SUBU

- **TBMI_ESTUD_ASIG**

Esta tabla guardará información sobre las asignaturas tomadas por un estudiante. Su clave primaria está formada por el código del estudiante en la subunidad→ EsAs_Estud_Subu, el código de la asignatura → EsAs_Asignatura y el número de crédito →

EsAs_Numer_Cred. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMC_CALIF_PROM	CaPr_Estud_Subu
TBMC_CALIF_PARC	CaPa_Estud_Subu
TBMA_ASIGNATURA	Asig_Codigo
TBMI_ESTUD_SUBU	EsSu_CodEs_Subu
TBMC_SOLIC_ESPE	SoEs_Estud_Subu

Tabla 1.22. Tablas que se relacionan con TBMI_ESTUD_ASIG

- **TBMI_ESTUD_NIVE**

Esta tabla guardará información sobre el desenvolvimiento de cada estudiante en los niveles que va cursando. Su clave primaria es el código del estudiante en la subunidad → EsNi_Estud_Subu y un número de matrícula (número generado por el sistema cuando se están inscribiendo, consiste en: año + mes + número secuencial) → EsNi_Numer_Matr. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_ESTUD_SUBU	EsSu_CodEs_Subu
TBMI_TIPO_CURS	TiCu_Codigo
TBMM_ORDEN_PAGO	OrPa_Estud_Subu OrPa_Numer_Matr

Tabla 1.23. Tablas que se relacionan con TBMI_ESTUD_NIVE

- **TBMI_HORAR_ESTU**

Esta tabla guardará información sobre cada horario generado para cada estudiante. Su clave primaria está formada por el código del horario del grupo → HoEs_Horar_Grup, el código del grupo → HoEs_Grupo y el código del estudiante en la subunidad → HoEs_Estud_Subu. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_ESTUD_SUBU	EsSu_CodEs_Subu
TBMI_HORAR_GRUP	HoGr_Codigo HoGr_Grupo

Tabla 1.24. Tablas que se relacionan con TBMI_HORAR_ESTU

- **TBMI_ESTUD_PENS**

Esta tabla guardará información sobre los pénsum tomados por cada estudiante. Su clave primaria está formada por el código del estudiante en la subunidad → EsPe_Estud_Subu y el código del pénsum que está tomando (actual) → EsPe_Pensu_Actu. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_MOTCA_PENS	MoPe_Codigo
TBMI_ESTUD_SUBU	EsSu_CodEs_Subu
TBMA_PÉNSUM	Pens_Codigo

Tabla 1.25. Tablas que se relacionan con TBMI_ESTUD_PENS

- **TBMM_CATEGORÍA**

Esta tabla guardará información sobre cada categoría. Su clave primaria es el código de la categoría → Cate_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMM_COLEGIO	Cole_Categoria
TBMM_MATRI_ASIG	MaAs_Categoria
TBMM_MATRI_GLOB	MaGl_Categoria

Tabla 1.26. Tablas que se relacionan con TBMM_CATEGORÍA

- **TBMM_COLEGIO**

Esta tabla guardará información sobre cada colegio. Su clave primaria es el código del colegio → Cole_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBME_ESTUDIANTE	Estu_Colegio
TBMM_CATEGORÍA	Cate_Codigo

Tabla 1.27. Tablas que se relacionan con TBMM_COLEGIO

- **TBMM_MATRI_GLOB**

Esta tabla guardará los valores de la matrícula global por categorías. Su clave primaria está formada por el código de la categoría → MaGl_Categoría y por el código de la subunidad → MaGl_Subunidad. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMM_CATEGORIA	Cate_Codigo

Tabla 1.28. Tablas que se relacionan con TBMM_MATRI_GLOB

- **TBMM_MATRI_ASIG**

Esta tabla guardará los valores de la matrícula por cada asignatura clasificada por categorías. Su clave primaria está formada el código de la categoría → MaAs_Categoría y por el código de la subunidad → MaAs_Subunidad. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMM_CATEGORIA	Cate_Codigo

Tabla 1.29. Tablas que se relacionan con TBMM_MATRI_ASIG

- **TBMM_INCRE_MATR**

Esta tabla guardará información sobre los porcentajes de incremento de matrícula. Su clave primaria está formada por el código del tipo de matrícula → InMa_Tipo_Matr y por el código de la

subunidad → InMa_Subunidad. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_SUBUN_ACAD	SuAc_Codigo

Tabla 1.30. Tablas que se relacionan con TBMM_INCRE_MATR

- **TBMM_DESCU_ARAN**

Esta tabla guardará información sobre todos los descuentos y aranceles que se aplican a las órdenes de pago. Su clave primaria es el código del descuento y/o arancel → DeAr_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_DETAL_MATR	DeMa_Motivo

Tabla 1.31. Tablas que se relacionan con TBMM_DESCU_ARAN

- **TBMM_CONCE_COBR**

Esta tabla guardará información sobre los conceptos de cobro. Su clave primaria es el código del concepto de cobro → CoCo_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMM_ORDEN_PAGO	OrPa_Conce_Cobr

Tabla 1.32. Tablas que se relacionan con TBMM_CONCE_COBR

- **TBMM_BANCO**

Esta tabla guardará información sobre cada banco. Su clave primaria es el código del banco → Banc_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMM_ORDEN_PAGO	OrPa_Banco

Tabla 1.33. Tablas que se relacionan con TBMM_BANCO

- **TBMM_ORDEN_PAGO**

Esta tabla guardará información sobre cada orden de pago generada. Su clave primaria es el código del estudiante en la subunidad → OrPa_Estud_Subu y el número de matrícula → OrPa_Numer_Matr. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_ESTUD_NIVE	EsNi_Estud_Subu EsNi_Numer_Matr
TBMM_DETAL_MATR	DeMa_Estud_Subu DeMa_Numer_Matr
TBMM_CONCE_COBR	CoCo_Codigo
TBMM_BANCO	Banc_Codigo
TBMM_CUOTA	Cuot_Estud_Subu Cuot_Numer_Matr

Tabla 1.34. Tablas que se relacionan con TBMM_ORDEN_PAGO

- **TBMM_DETAL_MATR**

Esta tabla guardará los detalles de cada matrícula generada. Su clave primaria está formada por el código del estudiante en la subunidad → DeMa_Estud_Subu, el número de matrícula → DeMa_Numer_Matr y el motivo (descuento o arancel) → DeMa_Motivo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMM_DESCU_ARAN	DeAr_Codigo
TBMM_ORDEN_PAGO	OrPa_Estud_Subu OrPa_Numer_Matr

Tabla 1.35. Tablas que se relacionan con TBMM_DETAL_MATR

- **TBMM_CUOTA**

Esta tabla guardará información sobre cada cuota en que se ha dividido una orden de pago. Su clave primaria está formada por el código del estudiante en la subunidad → Cuot_Estud_Subu, el número de matrícula → Cuot_Numer_Matr y el número de la cuota → Cuot_Número. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMM_ORDEN_PAGO	OrPa_Estud_Subu OrPa_Numer_Matr

Tabla 1.36. Tablas que se relacionan con TBMM_CUOTA

- **TBMM_AUTORIDAD**

Esta tabla guardará información sobre cada autoridad de una subunidad y su correspondiente firma digitalizada. Su clave primaria

está formada por el código de la autoridad → Auto_Codigo y el código de la subunidad → Auto_Subunidad. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMA_SUBUN_ACAD	SuAc_Codigo

Tabla 1.37. Tablas que se relacionan con TBMM_AUTORIDAD

- **TBMM_PERIO_LECT**

Esta tabla guardará información sobre cada periodo lectivo que se crea dentro del sistema. Su clave primaria está formada por el año lectivo → PeLe_Anio_Lect, el tipo de curso → PeLe_Tipo_Curs y el ciclo → PeLe_Ciclo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_TIPO_CURS	TiCu_Codigo
TBMC_ENTRE_CALI	EnCa_Anio_Lect EnCa_Tipo_Curs EnCa_Ciclo

Tabla 1.38. Tablas que se relacionan con TBMM_PERIO_LECT

- **TBMC_TIPO_CALI**

Esta tabla guardará información sobre los tipos de calificaciones que hay en una subunidad. Su clave primaria es el código del tipo de calificación → TiCa_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMA_SUBUN_ACAD	SuAc_Codigo
TBMC_CALIF_PARC	CaPa_Tipo_Cali

Tabla 1.39. Tablas que se relacionan con TBMC_TIPO_CALI

- **TBMC_CALIF_PARC**

Esta tabla guardará las calificaciones parciales de cada estudiante. Su clave primaria está formada por el código del estudiante en la subunidad → CaPa_Estud_Subu, el código de la asignatura → CaPa_Asignatura, el número de crédito → CaPa_Numer_Cred y el tipo de calificación → CaPa_Tipo_Cali. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMC_TIPO_CALI	TiCa_Codigo
TBMI_ESTUD_ASIG	EsAs_Estud_Subu EsAs_Asignatura EsAs_Numer_Cred

Tabla 1.40. Tablas que se relacionan con TBMC_CALIF_PARC

- **TBMC_CALIF_PROM**

Esta tabla guardará los promedios de los estudiantes en las diversas materias que se generen en el sistema. Su clave primaria es el código del estudiante en la subunidad → CaPr_Estud_Subu, el código de la asignatura → CaPr_Asignatura y el número de crédito → CaPr_Numer_Cred. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMI_ESTUD_ASIG	EsAs_Estud_Subu EsAs_Asignatura EsAs_Numer_Cred

Tabla 1.41. Tablas que se relacionan con TBMC_CALIF_PROM

- **TBMC_ENTRE_CALI**

Esta tabla guardará las fechas que los docentes tienen como máximo para entregar las calificaciones de cada parcial. Su clave primaria está formada por el año lectivo → EnCa_Anio_Lect, el código del tipo de curso → EnCa_Tipo_Curs, el ciclo → EnCa_Ciclo y el tipo de calificación → EnCa_Tipo_Cali. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMM_PERIO_LECT	PeLe_Anio_Lect PeLe_Tipo_Curs PeLe_Ciclo

Tabla 1.42. Tablas que se relacionan con TBMC_ENTRE_CALI

- **TBMC_TIPO_SOLI**

Esta tabla guardará tipos de solicitudes. Su clave primaria es el código del país → TiSo_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMC_SOLIC_ESPE	SoEs_Tipo_Soli

Tabla 1.43. Tablas que se relacionan con TBMC_TIPO_SOLI

- **TBMC_SOLIC_ESPE**

Esta tabla guardará información sobre las solicitudes especiales. Su clave primaria está formada por el código del estudiante en la subunidad → SoEs_Estud_Subu, el código de la asignatura → SoEs_Asignatura, el número de crédito → SoEs_Numer_Cred y el número de solicitud → SoEs_Numer_Soli. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	**CAMPO**
TBMC_TIPO_SOLI	TiSo_Codigo
TBMI_ESTUD_ASIG	EsAs_Estud_Subu EsAs_Asignatura EsAs_Numer_Cred

Tabla 1.44. Tablas que se relacionan con TBMC_SOLIC_ESPE

- **TBMN_TIPO_PROY**

Esta tabla guardará información sobre los tipos de proyectos. Su clave primaria es el código del tipo de proyecto → TiPr_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMN_PROYE_GRAD	PrGr_Tipo

Tabla 1.45. Tablas que se relacionan con TBMN_TIPO_PROY

- **TBMN_PROYE_GRAD**

Esta tabla guardará información sobre los proyectos que realizan los estudiantes de una subunidad para culminar sus estudios universitarios. Su clave primaria está formada por el código del proyecto de grado→ PrGr_Codigo y el código de la subunidad → PrGr_Subunidad. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMN_TIPO_PROY	TiPr_Codigo
TBMA_SUBUN_ACAD	SuAc_Codigo

Tabla 1.46. Tablas que se relacionan con TBMN_PROYE_GRAD

- **TBMS_PANTALLA**

Esta tabla guardará información sobre cada pantalla creada en el sistema. Su clave primaria es el código de la pantalla → Pant_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMS_PERMISO	Perm_Pantalla

Tabla 1.47. Tablas que se relacionan con TBMS_PANTALLA

- **TBMS_PERFIL**

Esta tabla guardará información sobre cada perfil del usuario. Su clave primaria es el código del perfil → Perf_Codigo. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMS_PERMISO	Perm_Perfil
TBMS_USUARIO	Usua_Perfil

Tabla 1.48. Tablas que se relacionan con TBMS_PERFIL

- **TBMS_PERMISO**

Esta tabla guardará información sobre los permisos que se otorgan a los usuarios sobre las pantallas. Su clave primaria está formada por el código de la pantalla → Perm_Pantalla y el código del perfil → Perm_Perfil. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMS_PERFIL	Perf_Codigo
TBMS_PANTALLA	Pant_Codigo

Tabla 1.49. Tablas que se relacionan con TBMS_PERMISO

- **TBMS_USUARIO**

Esta tabla guardará información sobre cada usuario. Su clave primaria es el código del usuario → Usua_Usuario. La relación dentro del diagrama que lleva con cada tabla es la siguiente:

TABLA	CAMPO
TBMS_PERFIL	Perf_Codigo

Tabla 1.50. Tablas que se relacionan con TBMS_USUARIO

1.3. Descripción Técnica De Las Pantallas Del Sistema Académico

1.3.1. Pantalla de Acceso

Título	Nombre	Archivos de consulta
Acceso al sistema	FrmAcceso.vb	PrjConexion.dll Base_Acad.ini

Tabla 1.51. Descripción Técnica de la Pantalla de Acceso

1.3.2. Pantallas de Mantenimiento

Título	Nombre	Código	Tablas que modifica
Países	FrmPais.vb	FRM001	TBMG_PAIS
Provincias	FrmProvincia.vb	FRM002	TBMG_PROVINCIA
Ciudades	FrmCiudad.vb	FRM003	TBMG_CIUDAD
Categorías de Estudiantes	FrmCategoria.vb	FRM004	TBMM_CATEGORIA
Colegios	FrmColegio.vb	FRM005	TBMM_COLEGIO
Especializaciones de bachillerato	FrmEspecializ.vb	FRM006	TBME_ESPECIALIZ
Universidades	FrmUniversida.vb	FRM007	TBMG_UNIVERSIDA
Títulos Universitarios	FrmTitul_Univ.vb	FRM008	TBMD_TITUL_UNIV
Tipos de Subunidad	FrmTipo_Subu.vb	FRM009	TBMA_TIPO_SUBU
Tipos de Solicitudes	FrmTipo_Soli.vb	FRM010	TBMC_TIPO_SOLI
Descuentos y Aranceles	FrmDescu_Aran.vb	FRM011	TBMM_DESCU_ARAN
Motivos de Cambio de Pénsum	FrmMotCa_Pens.vb	FRM012	TBMA_MOTCA_PENS
Tipos de Proyectos	FrmTipo_Proy.vb	FRM013	TBMN_TIPO_PROY
Bancos	FrmBanco.vb	FRM014	TBMM_BANCO
Conc de Cobros	FrmConce_Cobr.vb	FRM015	TBMM_CONCE_COBR

Tabla 1.52. Descripción Técnica de las Pantallas de Mantenimiento (1)

Ficha Docente	FrmDocente.vb	FRM016	TBMD_DOCENTE
Ficha Estudiantil	FrmEstudiante.vb	FRM017	TBME_ESTUDIANTE
Facultades	FrmFacultad.vb	FRM018	TBMA_FACULTAD
Subunidades Académicas	FrmSubun_Acad.vb	FRM019	TBMA_SUBUN_ACAD
Areas Académicas	FrmArea_Acad.vb	FRM020	TBMA_AREA_ACAD
Valor de Matrícula Global	FrmMatri_Glob.vb	FRM021	TBMM_MATRI_GLOB
Valor de Matrícula por Asignatura	FrmMatri_Asig.vb	FRM022	TBMM_MATRI_ASIG
Incremento de Matrículas	FrmIncre_Matr.vb	FRM023	TBMM_INCRE_MATR
Tipos de Cursos	FrmTipo_Curs.vb	FRM024	TBMI_TIPO_CURS
Tipos de Calificaciones	FrmTipo_Cali.vb	FRM025	TBMC_TIPO_CALI

Tabla 1.53. Descripción Técnica de las Pantallas de Mantenimiento (2)

1.3.3. Pantallas de Control Académico

Título	Nombre	Código	Tablas que modifica
Mantenimiento de Asignatura	FrmAsignatura.vb	FRM026	TBMA_ASIGNATURA
Mantenimiento de Pénsum	FrmPensum.vb	FRM027	TBMA_PÉNSUM TBMA_PREREQUISI
Prerrequisitos	FrmPrerequisi.vb	FRM028	TBMA_PREREQUISI

Tabla 1.54. Descripción Técnica de las Pantallas de Control Académico

1.3.4. Pantallas de Períodos Lectivos

Título	Nombre	Código	Tablas que modifica
Creación de Periodos Lectivos	FrmPerio_Lect.vb	FRM029	TBMM_PERIO_LECT TBMA_PENSUM TBMC_CALIF_PARC TBMI_ESTUD_ASIG
Cronograma de Entrega de Notas	FrmEntre_Cali.vb	FRM030	TBMC_ENTRE_CALI

Tabla 1.55. Descripción Técnica de las Pantallas de Períodos Lectivos

1.3.5. Pantallas de Control de Horarios

Título	Nombre	Código	Tablas que modifica
Distributivo de Asignaturas a Docentes	FrmDocen_Asig.vb	FRM031	TBMI_DOCEN_ASIG
Horarios por Grupo	FrmHorar_Grup.vb	FRM032	TBMI_HORAR_GRUP

Tabla 1.56. Descripción Técnica de las Pantallas de Control de Horarios

1.3.6. Pantallas de Inscripción y Matriculación

Título	Nombre	Código	Tablas que modifica
Reg de Estudiantes a una Subunidad	FrmEstud_Subu.vb	FRM033	TBMI_ESTUD_SUBU

Tabla 1.57. Descripción Técnica de las Pantallas de Inscripción y Matriculación (1)

Inscripción de Asignaturas por Estudiante	FrmEstud_Asig.vb	FRM034	TBMI_ESTUD_ASIG TBMI_ESTUD_NIVE TBMI_HORAR_ESTU
Orden de Pago	FrmOrden_Pago.vb	FRM035	TBMM_ORDEN_PAGO
Cuotas	FrmCuota.vb	FRM036	TBMM_CUOTA
Cuotas Cancel.	FrmRegOr_Canc.vb	FRM037	TBMM_CUOTA
Cambio de Categoría	FrmCambi_Cate.vb	FRM038	TBME_ESTUDIANTE
Cambio de Pénsum	FrmCambi_Pens.vb	FRM039	TBMI_ESTUD_SUBU TBMI_ESTUD_PENS

Tabla 1.58. Descripción Técnica de las Pantallas de Inscripción y Matriculación (2)

1.3.7. Pantallas de Registro de Calificaciones

Título	Nombre	Código	Tablas que modifica
Ingreso de Calificación Parcial	FrmCalif_Parc.vb	FRM040	TBMC_CALIF_PARC
Cálculo de Promedios	FrmCalif_Prom.vb	FRM041	TBMC_CALIF_PROM
Modificación de una nota parcial	FrmMoCalifica.vb	FRM063	TBMC_CALIF_PARC

Tabla 1.59. Descripción Técnica de las Pantallas de Registro de Calificaciones

1.3.8. Pantallas de Registro de Solicitudes

Título	Nombre	Código	Tablas que modifica
Solicitudes Especiales	FrmSolic_Espe.vb	FRM042	TBMC_SOLIC_ESPE
Recalificación	FrmRecalifica.vb	FRM043	TBMC_CALIF_PARC

Tabla 1.60. Descripción Técnica de las Pantallas de Solicitudes

1.3.9. Pantallas de Reportes

Título	Nombre	Código
Informe de Estudiantes por Grupo	FrmLista_Estu.vb	FRM044
Informe de Inscripciones por Estudiante	FrmLista_Insc.vb	FRM045
Consulta de Docentes por Subunidad	FrmLista_Doce.vb	FRM046
Resumen de Descuentos y Aranceles por Periodo	FrmLista_Desc.vb	FRM047
Reporte de Horarios por Grupo	FrmRepHo_Grup.vb	FRM048
Reporte de Horarios por Docentes	FrmRepHo_Doce.v	FRM049
Informe de Calificaciones por Asignatura	FrmRepCa_Asig.vb	FRM050
Resumen de Calificaciones por Estudiante	FrmRepCa_Estu.vb	FRM051
Comparación de Últimos Periodos Lectivos	FrmRepCo_Lect.vb	FRM052
Informe de Entrega de Calificaciones	FrmLista_Entr_Cali	FRM053
Consulta de los Mejores Promedios	FrmRepEs_Prom	FRM054
Reporte de Estudiantes Matriculados	FrmRepEs_Matr	FRM055
Estadística de Inscritos	FrmRepRe_Insc	FRM056

Tabla 1.61. Descripción Técnica de las Pantallas de Reportes

1.3.10. Pantallas de Seguridad y Administración

Título	Nombre	Código	Tablas que modifica
Cambio de Contraseñas	FrmCambi_Pass.vb	FRM058	Base de datos
Creación de Usuarios	FrmUsuario.vb	FRM059	TBMS_USUARIO
Creación de Perfiles	FrmPerfil.vb	FRM060	TBMS_PERFIL
Creación de Pantallas	FrmPantalla.vb	FRM061	TBMS_PANTALLA
Creación de Permisos	FrmPermiso.vb	FRM062	TBMS_PERMISO

Tabla 1.61. Descripción Técnica de las Pantallas de Reportes

CAPITULO 2

2. DICCIONARIO DE DATOS

2.1. Tablas de la Base de Datos

- **TBMA_FACULTAD**

CAMPO	TIPO	DEFINICIÓN
Facu_Codigo	varchar(2)	Código de Facultad **(PK)**
Facu_Nombre	varchar(60)	Nombre de Facultad
Facu_Decano	varchar(5)	Código del docente (decano)
Facu_Direccion	varchar(60)	Dirección
Facu_Ubicacion	varchar(60)	Ubicación de la facultad
Facu_Observacio	varchar(120)	Observaciones
Facu_Usuario	varchar(15)	Código del usuario que modifica
Facu_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.1. Campos de la Tabla TBMA_FACULTAD

☞ **TBMA_SUBUN_ACAD**

CAMPO	**TIPO**	**DEFINICIÓN**
SuAc_Codigo	varchar(4)	Código de la subunidad **(PK)**
SuAc_Facultad	varchar(2)	Código de la facultad
SuAc_Nombre	varchar(60)	Nombre de la subunidad
SuAc_Director	varchar(5)	Código del docente (director)
SuAc_Telefono1	varchar(10)	Teléfono 1
SuAc_Telefono2	varchar(10)	Teléfono 2
SuAc_Tipo	varchar(2)	Código del tipo de subunidad
SuAc_Nivel	varchar(2)	Número de niveles que tiene la subunidad para egresar
SuAc_Periodicid	varchar(1)	Periodicidad: (A) anual, (S) semestral, (T) trimestral
SuAc_Usuario	varchar(15)	Código del usuario que modifica
SuAc_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.2. Campos de la Tabla TBMA_SUBUN_ACAD

☞ **TBMA_TIPO_SUBU**

CAMPO	**TIPO**	**DEFINICIÓN**
TiSu_Codigo	varchar(2)	Código del tipo de subunidad **(PK)**
TiSu_Descripcio	varchar(60)	Descripción del tipo de subunidad
TiSu_Usuario	varchar(15)	Código del usuario que modifica
TiSu_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.3. Campos de la Tabla TBMA_TIPO_SUBU

☞ **TBMA_AREA_ACAD**

CAMPO	**TIPO**	**DEFINICIÓN**
ArAc_Codigo	varchar(3)	Código del área académica **(PK)**
ArAc_Subunidad	varchar(4)	Código de la subunidad **(PK)**
ArAc_Coordinado	varchar(5)	Código del docente-coordinador
ArAc_Descripcio	varchar(60)	Descripción del área académica
ArAc_Usuario	varchar(15)	Código del usuario que modifica
ArAc_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.4. Campos de la Tabla TBMA_AREA_ACAD

- **TBMA_PREREQUISI**

CAMPO	TIPO	DEFINICION
Prer_Asignatura	varchar(7)	Código de la Asignatura
Prer_Pensum	varchar(6)	Código del Pénsum
Prer_Requisito	varchar(7)	Código de la asignatura-requisito
Prer_Nivel	varchar(2)	Nivel-requisito
Prer_Usuario	varchar(15)	Código del usuario que modifica
Prer_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.5. Campos de la Tabla TBMA_PREREQUISI

- **TBMA_MOTCA_PENS**

CAMPO	TIPO	DEFINICIÓN
MoPe_Codigo	varchar(2)	Código del motivo de cambio de pénsum **(PK)**
MoPe_Descripcio	varchar(30)	Descripción del motivo de cambio de pénsum
MoPe_Usuario	varchar(15)	Código del usuario que modifica
MoPe_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.6. Campos de la Tabla TBMA_MOTCA_PENS

☞ **TBMA_ASIGNATURA**

CAMPO	**TIPO**	**ASIGNATURA**
Asig_Codigo	varchar(7)	Código de la asignatura **(PK)**
Asig_Subunidad	varchar(4)	Código de la subunidad
Asig_Nombre	varchar(30)	Nombre de la asignatura
Asig_Area_Acad	varchar(3)	Código del área académica a la que pertenece dicha asignatura
Asig_Nivel	varchar(2)	Nivel en el cual se dicta la asignatura
Asig_Horas_Sema	varchar(2)	Número de horas semanales que se da la asignatura
Asig_Tipo	varchar(1)	Tipo de asignatura: (O) optativa o (S) selectiva
Asig_Numer_Punt	varchar(2)	Número de puntos acreditados por las materias aprobadas
Asig_Usuario	varchar(15)	Código del usuario que modifica
Asig_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.7. Campos de la Tabla TBMA_ASIGNATURA

☞ **TBMA_PENSUM**

CAMPO	**TIPO**	**DEFINICIÓN**
Pens_Codigo	varchar(6)	Código del pénsum **(PK)**
Pens_Subunidad	varchar(4)	Código de la subunidad
Pens_Descripcio	varchar(60)	Descripción del pénsum que esta vigente
Pens_Fecha_Inic	datetime	Fecha de inicio del pénsum
Pens_Fecha_Cadu	datetime	Fecha en que caduca el pénsum
Pens_Estado	varchar(1)	Guarda solamente un carácter: activo(a), inactivo(i)
Pens_Usuario	varchar(15)	Código del ultimo usuario que modifico el registro del pénsum
Pens_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.8. Campos de la Tabla TBMA_ PENSUM

☞ **TBMC_TIPO_CALI**

CAMPO	**TIPO**	**DEFINICIÓN**
TiCa_Codigo	varchar(6)	Código del tipo de calificación **(PK)**
TiCa_Subunidad	varchar(4)	Código de la Subunidad
TiCa_Descripcio	varchar(60)	Descripción del tipo de calificación: PRIMER PARCIAL, SEGUNDO PARCIAL, SUSPENSO.
TiCa_Reemp_Nota	Varchar(1)	Indica si este tipo de nota reemplaza a la más baja de los parciales: (S) sí, (N) no
TiCa_Suspenso	Varchar(1)	Indica si este tipo de nota se refiere a un examen suspenso
TiCa_Usuario	varchar(15)	Código del usuario que modifica
TiCa_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.9. Campos de la Tabla TBMC_TIPO_CALI

☞ **TBMC_CALIF_PROM**

CAMPO	**TIPO**	**DEFINICIÓN**
CaPr_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
CaPr_Asignatura	varchar(7)	Código de la asignatura **(PK)**
CaPr_Numer_Cred	numeric(1)	Número de vez que toma la materia **(PK)**
CaPr_Prome_Cali	varchar(9)	Promedio de calificaciones
CaPr_Prome_Asis	varchar(9)	Promedio de asistencia
CaPr_Observacio	varchar(120)	Observaciones sobre las calificaciones promedios
CaPr_Usuario	varchar(15)	Código del usuario que modifica
CaPr_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.10. Campos de la Tabla TBMC_CALIF_PROM

☞ **TBMC_CALIF_PARC**

CAMPO	**TIPO**	**DEFINICIÓN**
CaPa_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
CaPa_Asignatura	varchar(7)	Código de la Asignatura **(PK)**
CaPa_Numer_Cred	numeric(1)	Número de vez que toma la materia **(PK)**
CaPa_Calif_Trab	varchar(9)	Calificación de trabajos, deberes
CaPa_Calif_Exam	varchar(9)	Nota que obtiene en el examen
CaPa_Calif_Tota	varchar(9)	Total: examen + trabajo
CaPa_Tipo_Cali	varchar(6)	Código del tipo de calificación **(PK)**
CaPa_Porce_Asis	varchar(9)	Porcentaje de asistencia
CaPa_Estad_Regi	varchar(1)	Estado del registro: (D) digitado o (B) bloqueado
CaPa_Observacio	varchar(120)	Observaciones

Tabla 2.11. Campos de la tabla TBMC_CALIF_PARC (1)

CaPa_Usuario	varchar(15)	Código del usuario que modifica
CaPa_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.12. Campos de la tabla TBMC_CALIF_PARC (2)

- **TBMC_TIPO_SOLI**

CAMPO	**TIPO**	**DEFINICIÓN**
TiSo_Codigo	varchar(2)	Código del tipo de solicitud **(PK)**
TiSo_Descripcio	varchar(60)	Descripción del tipo de la solicitud
TiSo_Usuario	varchar(15)	Código del usuario que modifica
TiSo_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.13. Campos de la Tabla TBMC_TIPO_SOLI

☞ **TBMC_SOLIC_ESPE**

CAMPO	**TIPO**	**DEFINICION**
SoEs_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
SoEs_Asignatura	varchar(7)	Código de la asignatura **(PK)**
SoEs_Numer_Vez	numeric(1)	Número de vez que toma la asignatura **(PK)**
SoEs_Tipo_Soli	varchar(2)	Código del tipo de solicitud
SoEs_Numer_Soli	varchar(10)	Número de la Solicitud **(PK)**
SoEs_Fecha_Apro	Datetime	Fecha de aprobación de solicitud
SoEs_Observacio	varchar(120)	Observaciones
SoEs_Usuario	varchar(15)	Código del usuario que modifica
SoEs_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.14. Campos de la Tabla TBMC_SOLIC_ESPE

☞ **TBMD_DOCENTE**

CAMPO	**TIPO**	**DEFINICIÓN**
Doce_Codigo	varchar(5)	Código del Docente **(PK)**
Doce_Cedula	varchar(10)	Cedula o pasaporte del docente
Doce_Apellido	varchar(30)	Apellidos del Docente
Doce_Nombre	varchar(30)	Nombres del Docente
Doce_Nacionalid	varchar(3)	Código de Nacionalidad
Doce_Provi_Naci	varchar(3)	Código de la Provincia de nacimiento
Doce_Ciuda_Naci	varchar(3)	Código de la Ciudad de nacimiento
Doce_Fecha_Naci	Datetime	Fecha de nacimiento
Doce_Cedul_Mili	varchar(10)	Cedula militar
Doce_Recor_Poli	varchar(10)	Record Policial
Doce_Provi_Resi	varchar(3)	Código de la Provincia donde reside
Doce_Ciuda_Resi	varchar(3)	Código de la ciudad donde reside
Doce_Domicilio	varchar(120)	Dirección domiciliaria
Doce_Telefono1	varchar(10)	Teléfono 1
Doce_Telefono2	varchar(10)	Teléfono 2

Tabla 2.15. Campos de la Tabla TBMD_DOCENTE (1)

Doce_Empresa	Varchar(60)	Nombre de la empresa
Doce_Direc_Trab	varchar(120)	Dirección del trabajo
Doce_Telef_Trab	varchar(10)	Teléfono del trabajo
Doce_Carne_Afil	varchar(10)	Carnet de afiliación al seguro
Doce_Universida	varchar(3)	Código de la universidad (graduó)
Doce_Titul_Univ	varchar(3)	Código del titulo universitario
Doce_Logro_Obte	varchar(120)	Logros obtenidos
Doce_Usuario	varchar(15)	Código del usuario que modifica
Doce_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.16. Campos de la Tabla TBMD_DOCENTE (2)

☞ **TBMD_TITUL_UNI**

CAMPO	**TIPO**	**DEFINICIÓN**
TiUn_Codigo	varchar(3)	Código del titulo universitario **(PK)**
TiUn_Descripcio	varchar(60)	Descripción de título universitario
TiUn_Abreviatur	varchar(5)	Abreviatura del título
TiUn_Usuario	varchar(15)	Código del usuario que modifica
TiUn_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.17. Campos de la Tabla TBMD_TITUL_UNIV

☞ **TBME_ESTUDIANTE**

CAMPO	**TIPO**	**DEFINICIÓN**
Estu_Codigo	varchar(6)	Código del estudiante **(PK)**
Estu_Cedula	varchar(10)	Cedula de identidad o Número de pasaporte si es extranjero
Estu_Apellido	varchar(30)	Apellidos del estudiante
Estu_Nombre	varchar(30)	Nombres del estudiante
Estu_Sexo	varchar(1)	Sexo: (1)masculino y (2)femenino
Estu_Nacionalid	varchar(3)	Código de la nacionalidad
Estu_Provi_Naci	varchar(3)	Código de la Provincia de nacimiento
Estu_Ciuda_Naci	varchar(3)	Código de la Ciudad de nacimiento
Estu_Fecha_Naci	datetime	Fecha de nacimiento
Estu_Estad_Civi	varchar(1)	Estado civil: (1)casado, (2)soltero, (3)viudo, (4)divorciado
Estu_Recor_Poli	varchar(10)	Record Policial
Estu_Provi_Domi	varchar(3)	Código de la Provincia donde reside
Estu_Ciuda_Domi	varchar(3)	Código de la Ciudad donde reside

Tabla 2.18. Campos de la Tabla TBME_ESTUDIANTE (1)

Estu_Direc_Domi	varchar(120)	Dirección domiciliaria
Estu_Telef_Domi	varchar(10)	Teléfono del domicilio
Estu_Nombr_Padr	varchar(60)	Nombres y apellidos del padre
Estu_Nombr_Madr	varchar(60)	Nombres y apellidos de la madre
Estu_Colegio	varchar(4)	Código del colegio donde se graduó
Estu_Categoria	varchar(3)	Código de la categoría del colegio
Estu_Especializ	varchar(2)	Código de la especialización
Estu_Anio_Grad	varchar(4)	Año de graduación del estudiante
Estu_Nota_Grad	varchar(2)	Nota de graduación en el colegio
Estu_Universida	varchar(3)	Código de la universidad, si el estudiante proviene de otra universidad
Estu_Nombr_Empr	varchar(60)	Nombre de la empresa donde trabaja
Estu_Direc_Empr	varchar(120)	Dirección de la empresa donde trabaja
Estu_Telef_Trab	varchar(10)	Teléfono del trabajo
Estu_CarneT_IESS	varchar(10)	Carnet de afiliación al IESS
Estu_Email	varchar(40)	Correo electrónico

Tabla 2.19. Campos de la Tabla TBME_ESTUDIANTE (2)

Estu_Grupo_Sang	varchar(2)	Tipo de sangre
Estu_Cedul_Mili	varchar(10)	Cédula militar
Estu_Exame_Grac	Numeric(2)	Número de exámenes de gracia realizados por el estudiante
Estu_Observacio	varchar(120)	Observaciones
Estu_Usuario	varchar(15)	Código del usuario que modifica
Estu_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.20. Campos de la Tabla TBME_ESTUDIANTE (3)

- **TBME_ESPECIALIZ**

CAMPO	**TIPO**	**DEFINICIÓN**
Espe_Codigo	varchar(2)	Código de la especialización **(PK)**
Espe_Descripcio	varchar(30)	Descripción de la especialización
Espe_Usuario	varchar(15)	Código del usuario que modifica
Espe_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.21. Campos de la Tabla TBME_ESPECIALIZ

☞ **TBME_FOTO_ESTU**

CAMPO	TIPO	DEFINICIÓN
FoEs_Estudiante	varchar(6)	Código del estudiante **(PK)**
FoEs_Imagen	image	Foto del estudiante
FoEs_Usuario	varchar(15)	Código del usuario que modifica
FoEs_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.22. Campos de la Tabla TBME_FOTO_ESTU

☞ **TBMG_UNIVERSIDA**

CAMPO	TIPO	DEFINICIÓN
Univ_Codigo	varchar(3)	Código de la Universidad **(PK)**
Univ_Descripcio	varchar(60)	Nombre de la Universidad
Univ_Pais	varchar(3)	Código del País donde queda la Universidad
Univ_Provincia	varchar(3)	Código de la Provincia donde queda la Universidad
Univ_Ciudad	varchar(3)	Código de la Ciudad donde queda la Universidad
Univ_Dirección	varchar(120)	Dirección de la Universidad
Univ_Telefono	varchar(10)	Teléfono de la Universidad

Tabla 2.23. Campos de la Tabla TBMG_UNIVERSIDA (1)

Univ_Usuario	varchar(15)	Código del usuario que modifica
Univ_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.24. Campos de la Tabla TBMG_UNIVERSIDA (2)

- **TBMG_PAIS**

CAMPO	**TIPO**	**DEFINICIÓN**
Pais_Codigo	varchar(3)	Código del País **(PK)**
Pais_Descripcio	varchar(60)	Nombre del país
Pais_Nacionalid	varchar(30)	Nacionalidad del país
Pais_Usuario	varchar(15)	Código del usuario que modifica
Pais_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.25. Campos de la Tabla TBMG_PAIS

- **TBMG_CIUDAD**

CAMPO	TIPO	DEFINICIÓN
Ciud_Codigo	varchar(3)	Código de la ciudad **(PK)**
Ciud_Provincia	varchar(3)	Código de la provincia **(PK)**
Ciud_Pais	varchar(3)	Código del país **(PK)**
Ciud_Descripcio	varchar(60)	Nombre de la ciudad
Ciud_Usuario	varchar(15)	Código del usuario que modifica
Ciud_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.26. Campos de la Tabla TBMG_CIUDAD

- **TBMG_PROVINCIA**

CAMPO	TIPO	DEFINICIÓN
Prov_Codigo	varchar(3)	Código de la provincia **(PK)**
Prov_Pais	varchar(3)	Código del país **(PK)**
Provi_Descripcio	varchar(60)	Nombre de la provincia
Prov_Usuario	varchar(15)	Código del usuario que modifica
Prov_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.27. Campos de la Tabla TBMG_PROVINCIA

☞ **TBMI_DOCEN_ASIG**

CAMPO	TIPO	DEFINICIÓN
DoAs_Docente	varchar(5)	Código del Docente
DoAs_Asignatura	varchar(7)	Código de la Asignatura
DoAs_Tipo_Curs	varchar(6)	Código del tipo de curso
DoAs_Grupo	varchar(4)	Código del grupo asignado para impartir la asignatura
DoAs_Anio_Lect	varchar(4)	Año lectivo
DoAs_Ciclo	varchar(2)	Ciclo
DoAs_Cupo_Estu	varchar(2)	Máximo número de estudiantes
DoAs_Usuario	varchar(15)	Código del usuario que modifica
DoAs_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.28. Campos de la Tabla TBMI_DOCEN_ASIG

☞ **TBMI_TIPO_CURSO**

CAMPO	**TIPO**	**DEFINICIÓN**
TiCu_Codigo	varchar(6)	Código del tipo de curso **(PK)**
TiCu_Subunidad	varchar(4)	Código de la subunidad
TiCu_Descripcio	varchar(60)	Descripción del tipo de curso: seminarios de arrastre, cursos de invierno , cursos normales, preuniversitarios, actividades extracurriculares, etc.
TiCu_Num_Parc	numeric(2)	Número de parciales
TiCu_NumEx_Adic	numeric(2)	Cantidad de exámenes adicionales
TiCu_MinAp_Asig	Numeric(2)	Mínima nota necesaria para aprobar la asignatura
TiCu_MinPa_Susp	numeric(2)	Mínima nota necesaria para pasar al suspenso
TiCu_MinAp_Susp	numeric(2)	Mínima nota que necesaria para aprobar el suspenso

Tabla 2.29. Campos de la Tabla TBMI_TIPO_CURS (1)

TiCu_MinPo_Asis	numeric(2)	Porcentaje mínimo de asistencia que el alumno debe tener
TiCu_Usuario	varchar(15)	Código del usuario que modifica
TiCu_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.30. Campos de la Tabla TBMI_TIPO_CURS (2)

☞ **TBMI_ESTUD_PENS**

CAMPO	**TIPO**	**DEFINICIÓN**
EsPe_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
EsPe_Pensu_Actu	varchar(6)	Código del pénsum actual **(PK)**
EsPe_Pensu_Ante	varchar(6)	Código del pénsum anterior
EsPe_MotCa_Pens	varchar(2)	Código del motivo de cambio de pénsum
EsPe_Usuario	varchar(15)	Código del usuario que modifica
EsPe_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.31. Campos de la Tabla TBMI_ESTUD_PENS

☞ **TBMI_HORAR_GRUP**

CAMPO	**TIPO**	**DEFINICIÓN**
HoGr_Codigo	varchar(2)	Código del horario del grupo **(PK)**
HoGr_Grupo	varchar(8)	Grupo al que pertenece el horario **(PK)**
HoGr_Seccion	varchar(1)	Sección: matutino(M), vespertino(V), nocturno(N)
HoGr_Dia_Sema	varchar(1)	Día de la semana :Lunes(1), Martes(2), Miércoles(3), Jueves(4), Viernes(5), Sábado(6), Domingo(7)
HoGr_Hora_Clas	varchar(1)	Hora clase del grupo: 1, 2, 3, etc.
HoGr_Asignatura	varchar(7)	Código de la asignatura
HoGr_Usuario	varchar(15)	Código del usuario que modifica
HoGr_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.32. Campos de la Tabla TBMI_HORAR_GRUP

☞ **TBMI_ESTUD_ASIG**

CAMPO	TIPO	DEFINICIÓN
EsAs_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
EsAs_Asignatura	varchar(7)	Código de la asignatura **(PK)**
EsAs_Numer_Cred	numeric(1)	Número de vez que el estudiante toma la materia **(PK)**
EsAs_Anio_Lect	varchar(4)	Año lectivo
EsAs_Ciclo	varchar(2)	Ciclo
EsAs_Tipo_Curs	varchar(6)	Código del tipo de curso
EsAs_Grupo	varchar(4)	Código del grupo donde se inscribió el estudiante
EsAs_Subgrupo	varchar(1)	Subgrupo: A, B
EsAs_Anulada	varchar(1)	Si anula la asignatura (S) o no anula la asignatura(N)
EsAs_Promedio	varchar(9)	Promedio del estudiante en esa asignatura
EsAs_Aprobado	varchar(1)	Si el estudiante aprobó (S) o reprobo(N)

Tabla 2.33. Campos de la Tabla TBMI_ESTUD_ASIG (1)

EsAs_Observacio	varchar(120)	Observaciones
EsAs_Usuario	varchar(15)	Código del usuario que modifica
EsAs_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.34. Campos de la Tabla TBMI_ESTUD_ASIG (2)

☞ **TBMI_ESTUD_SUBU**

CAMPO	**TIPO**	**DEFINICIÓN**
EsSu_CodEs_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
EsSu_Subunidad	varchar(4)	Código de la subunidad
EsSu_Estudiante	varchar(6)	Código del estudiante
EsSu_UltNi_Apro	varchar(2)	Ultimo nivel aprobado Ej.: 1, 2, etc.
EsSu_Pensu_Actu	varchar(6)	Pénsum vigente para el estudiante
EsSu_Cruce_Perm	tinyint	Número de cruces de horarios que se le permita al estudiante
EsSu_Fecha_Ingr	datetime	Fecha de ingreso a la subunidad
EsSu_Fecha_Egre	datetime	Fecha de egreso a la subunidad

Tabla 2.35. Campos de la Tabla TBMI_ESTUD_SUBU (1)

EsSu_Condi_Acad	varchar(1)	Guarda un carácter: activo(A), inactivo(I) , egresado(E), graduado(G)
EsSu_Prome_Egre	varchar(9)	Promedio con el que egreso
EsSu_Prome_Grad	varchar(9)	Promedio de tesis o seminario
EsSu_Prome_Fina	varchar(9)	Promedio con el que finaliza la carrera
EsSu_Proye_Grad	varchar(4)	Código del proyecto de grado
EsSu_NumPr_Grad	tinyint	Número de promoción
EsSu_Observacio	varchar(120)	Observaciones
EsSu_Usuario	varchar(15)	Código del usuario que modifica
EsSu_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.36. Campos de la Tabla TBMI_ESTUD_SUBU (2)

☞ **TBMI_ESTUD_NIVE**

CAMPO	**TIPO**	**DEFINICIÓN**
EsNi_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
EsNi_Numer_Matr	varchar(10)	Número de matrícula **(PK)**
EsNi_Tipo_Curs	varchar(6)	Código del tipo de curso
EsNi_Anio_Lect	varchar(4)	Año lectivo
EsNi_Ciclo	varchar(2)	Ciclo
EsNi_Nivel	varchar(2)	Nivel en el cual el estudiante tomo las asignaturas
EsNi_Numer_vez	varchar(1)	Número de vez que repite el nivel
EsNi_Mater_Toma	tinyint	Número de materias que ha tomado
EsNi_promedio	varchar(9)	Promedio que tuvo en ese nivel
EsNi_Numer_Susp	tinyint	Número de asignaturas que se quedo suspenso
EsNi_Mater_Apro	tinyint	Número de asignaturas aprobadas
EsNi_Mater_Repr	tinyint	Número de asignaturas reprobadas

Tabla 2.37. Campos de la Tabla TBMI_ESTUD_NIVE (1)

EsNi_Premiado	varchar(1)	Si recibió algún premio de la carrera si (S) o no (N)
EsNi_Sancionado	varchar(1)	Si ha recibido alguna sanción (S) o no (N)
EsNi_Observacio	varchar(120)	Observaciones
EsNi_Usuario	varchar(15)	Código del usuario que modifica
EsNi_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.38. Campos de la Tabla TBMI_ESTUD_NIVE (2)

☞ **TBMI_HORAR_ESTU**

CAMPO	**TIPO**	**DEFINICIÓN**
HoEs_Horar_Grup	varchar(3)	Código del horario del grupo **(PK)**
HoEs_Grupo	varchar(4)	Código del grupo **(PK)**
HoEs_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
HoEs_Usuario	varchar(15)	Código del usuario que modifica
HoEs_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.39. Campos de la Tabla TBMI_HORAR_ESTU

☞ **TBMM_MATRI_ASIG**

CAMPO	**TIPO**	**DEFINICION**
MaAs_Categoría	varchar(3)	Código de la categoría **(PK)**
MaAs_Subunidad	varchar(4)	Código de la subunidad en la que se determina dicho valor **(PK)**
MaAs_Valor	numeric(8,2)	Valor de la Matrícula por asignatura para esa determinada categoría
MaAs_Limite	numeric(1)	Número máximo de asignaturas para la que se aplica dicho valor
MaAs_Usuario	varchar(15)	Código del usuario que modifica
MaAs_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.40. Campos de la Tabla TBMM_MATRI_ASIG

☞ **TBMM_AUTORIDAD**

CAMPO	**TIPO**	**DEFINICIÓN**
Auto_Codigo	varchar(2)	Código de la autoridad **(PK)**
Auto_Subunidad	varchar(4)	Código de la subunidad **(PK)**
Auto_Cargo	varchar(30)	Cargo que desempeña
Auto_Apellido	varchar(30)	Apellido de dicha autoridad
Auto_Nombre	varchar(30)	Nombre de dicha autoridad
Auto_Firma	Image	Firma digitalizada
Auto_Usuario	varchar(15)	Código del usuario que modifica
Auto_FecUl_Modi	Datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.41. Campos de la Tabla TBMM_AUTORIDAD

☞ **TBMM_PERIO_LECT**

CAMPO	**TIPO**	**DEFINICION**
PeLe_Anio_Lect	Varchar(4)	Año lectivo **(PK)**
PeLe_Tipo_Curs	varchar(6)	Código del tipo de curso **(PK)**
PeLe_Ciclo	varchar(2)	Ciclo **(PK)**

Tabla 2.42. Campos de la tabla TBMM_PERIO_LECT (1)

PeLe_Fecha_Inic	Datetime	Fecha en que se cierran las matrículas (inicio del primer parcial)
PeLe_Fecha_Fin	Datetime	Fecha de cierre de periodo (finaliza)
PeLe_Fecha_Matr	Datetime	Fecha de apertura de la matriculación
PeLe_Estado	Varchar(1)	Estado actual del periodo: (E) espera, (M) inicio de matriculación, (I) cierre de matriculación, (F) cierre de periodo
PeLe_MaxNu_Cred	Numeric(1)	Máximo número de créditos que se aceptan en este periodo
PeLe_UltNu_Reci	Varchar(4)	Ultimo número de recibo de una orden de pago emitida. Es secuencial
PeLe_Usuario	varchar(15)	Código del usuario que modifica
PeLe_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.43. Campos de la tabla TBMM_PERIO_LECT (2)

- **TBMM_MATRI_GLOB**

CAMPO	TIPO	DEFINICIÓN
MaGl_Categoria	varchar(3)	Código de la categoría **(PK)**
MaGl_Subunidad	varchar(4)	Código de la subunidad en la que se determina dicho valor **(PK)**
MaGl_Valor	numeric(8,2)	Valor de la matrícula para esa determinada categoría
MaGl_Usuario	varchar(15)	Código del usuario que modifica
MaGl_FecUl_Modi	Datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.44. Campos de la tabla TBMM_MATRI_GLOB

- **TBMM_CATEGORÍA**

CAMPO	TIPO	DEFINICIÓN
Cate_Codigo	varchar(3)	Código de la categoría **(PK)**
Cate_Observacio	varchar(30)	Observaciones
Cate_Usuario	varchar(15)	Código del usuario que modifica
Cate_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.45. Campos de la tabla TBMM_CATEGORIA

☞ **TBMM_DETAL_MATR**

CAMPO	**TIPO**	**DEFINICION**
DeMa_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
DeMa_Numer_Matr	varchar(10)	Número de la matrícula **(PK)**
DeMa_Motivo	varchar(3)	Código de arancel o descuento **(PK)**
DeMa_Tipo	varchar(1)	Tipo: Arancel o descuento
DeMa_Monto	numeric(8,2)	Monto es el valor base para el calculo del interés o descuento
DeMa_Valor_Calc	numeric(8,2)	Valor calculado, generado al considerar el monto base, la forma de calculo y el valor del interés o descuento
DeMa_Usuario	varchar(15)	Código del usuario que modifica
DeMa_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.46. Campos de la tabla TBMM_DETAL_MATR

☞ **TBMM_DESCU_ARAN**

CAMPO	**TIPO**	**DEFINICION**
DeAr_Codigo	varchar(3)	Código del descuento/arancel **(PK)**
DeAr_Descripcio	varchar(60)	Descripción del descuento o arancel
DeAr_Tipo	varchar(1)	Tipo: Descuento(d) o Arancel(a)
DeAr_Valor	numeric(8,2)	Valor del descuento o arancel
DeAr_Usuario	varchar(15)	Código del usuario que modifica
DeAr_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.47. Campos de la tabla TBMM_DESCU_ARAN

☞ **TBMM_COLEGIO**

CAMPO	**TIPO**	**DEFINICIÓN**
Cole_Codigo	varchar(4)	Código del colegio **(PK)**
Cole_Categoría	varchar(3)	Código de la categoría a la que pertenece el colegio
Cole_Nombre	varchar(60)	Nombre del colegio

Tabla 2.48. Campos de la tabla TBMM_COLEGIO (1)

Cole_Tipo	varchar(1)	Tpo de colegio: Nacional (1), Particular (2)
Cole_Usuario	varchar(15)	Código del usuario que modifica
Cole_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.49. Campos de la tabla TBMM_COLEGIO (2)

☞ **TBMM_ORDEN_PAGO**

CAMPO	**TIPO**	**DEFINICIÓN**
OrPa_Estud_Subu	varchar(10)	Código del estudiante en la subunidad **(PK)**
OrPa_Numer_Matr	varchar(10)	Número de matrícula **(PK)**
OrPa_Conce_Cobr	Varchar(3)	Código del concepto de cobro
OrPa_Banco	Varchar(3)	Código del banco
OrPa_Numer_Reci	Varchar(4)	Número de Recibo
OrPa_Fecha_Emis	Datetime	Fecha de Emisión
OrPa_Valor_Tota	Numeric(8,2)	Valor total a pagar
OrPa_Observacio	Varchar(120)	Observaciones
OrPa_Usuario	varchar(15)	Código del usuario que modifica
OrPa_FecUl_Modi	Datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.50. Campos de la tabla TBMM_ORDEN_PAGO

☞ **TBMM_CUOTA**

CAMPO	TIPO	DEFINICIÓN
Cuot_Estud_Subu	varchar(10)	Código del estudiante dentro de la subunidad
Cuot_Numer_Matr	varchar(10)	Número de matrícula del estudiante en ese periodo lectivo
Cuot_Número	varchar(2)	El Número de la cuota
Cuot_Valor	numeric(8,2)	Valor que corresponde a ese cuota
Cuot_Estado	varchar(1)	Estado de la cuota: Pendiente, por pagar o cancelado
Cuot_Fecha_Venc	datetime	Fecha de vencimiento de la cuota
Cuot_Porce_Cuot	Numeric(8,2)	Porcentaje de la cuota
Cuot_Inter_Mens	Numeric(8,2)	Interés Mensual de la cuota
Cuot_Usuario	varchar(15)	Código del ultimo usuario que modifico el registro de las cuotas del alumno
Cuot_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.51. Campos de la tabla TBMM_CUOTA

- **TBMM_INCRE_MATR**

CAMPO	TIPO	DEFINICIÓN
InMa_Tipo_Matr	varchar(2)	Código del tipo de matrícula **(PK)**
InMa_Subunidad	varchar(4)	Código de la Subunidad **(PK)**
InMa_Descripcio	varchar(30)	Descripción del tipo de Matrícula: primera, segunda o tercera vez
InMa_Porce_incr	numeric(3,2)	Porcentaje de incremento
InMa_Usuario	varchar (15)	Código del usuario que modifica
InMa_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.52. Campos de la tabla TBMM_INCRE_MATR

- **TBMS_PERFIL**

CAMPO	TIPO	DEFINICIÓN
Perf_Codigo	varchar(2)	Código del perfil **(PK)**
Perf_Descripcio	varchar(30)	Descripción del perfil

Tabla 2.53. Campos de la tabla TBMS_PERFIL

☞ **TBMS_PERMISO**

CAMPO	**TIPO**	**DEFINICIÓN**
Perm_Pantalla	varchar(3)	Código de pantalla **(PK)**
Perm_Perfil	varchar(2)	Código del perfil **(PK)**
Perm_Tipo_Acce	varchar(3)	Código de tipo de acceso: (L) lectura, (I) impresión, (T) total

Tabla 2.54. Campos de la tabla TBMS_PERMISO

☞ **TBMS_USUARIO**

CAMPO	**TIPO**	**DEFINICIÓN**
Usua_Usuario	varchar(15)	Código de usuario **(PK)**
Usua_Perfil	varchar(2)	Código del perfil
Usua_Apellido	varchar(30)	Apellidos del usuario
Usua_Nombre	varchar(30)	Nombres del usuario
Usua_Estado	varchar(1)	Guarda un solo caracter: activo(A) o eliminado(E)
Usua_Caducidad	int	Tiempo en el que caduca la contraseña Ej.: 20, 30 días, etc
Usua_Subunidad	varchar(4)	Código de la subunidad a la que pertenece

Tabla 2.55. Campos de la tabla TBMS_USUARIO

- **TBMS_PANTALLA**

CAMPO	TIPO	DEFINICIÓN
Pant_Codigo	varchar(3)	Código de la pantalla **(PK)**
Pant_Descripcio	varchar(30)	Descripción de cada pantallas que esta en el sistema

Tabla 2.56. Campos de la tabla TBMS_PANTALLA

- **TBMN_TIPO_PROY**

CAMPO	TIPO	DEFINICIÓN
TiPr_Codigo	varchar(1)	Código del tipo de proyecto **(PK)**
TiPr_Descripcio	varchar(30)	Descripción del tipo de proyecto
TiPr_Usuario	Varchar(15)	Código del usuario que modifica
TiPr_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.57. Campos de la tabla TBMN_TIPO_PROY

☞ **TBMN_ PROYE_GRAD**

CAMPO	**TIPO**	**DEFINICIÓN**
PrGr_Codigo	varchar(4)	Código del proyecto de grado **(PK)**
PrGr_Subunidad	Varchar(4)	Código de la subunidad **(PK)**
PrGr_Tipo	Varchar(1)	Código del tipo de proyecto
PrGr_Descripcio	varchar(120)	Descripción del proyecto
PrGr_Estado	Varchar(1)	Estado o avance del proyecto
PrGr_Fecha_Inic	datetime	Fecha en que se inició el proyecto
PrGr_Fecha_Fin	datetime	Fecha que concluye
PrGr_Usuario	Varchar(15)	Código del usuario que modifica
PrGr_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.58. Campos de la tabla TBMN_ PROYE_GRAD

☞ **TBMM_CONCE_COBR**

CAMPO	**TIPO**	**DEFINICIÓN**
CoCo_Codigo	varchar(3)	Código de concepto de cobro **(PK)**
CoCo_Descripcio	Varchar(60)	Descripción del concepto de cobro
CoCo_Usuario	Varchar(15)	Código del usuario que modifica
PrGr_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.59. Campos de la tabla TBMM_CONCE_COBR

- **TBMM_BANCO**

CAMPO	TIPO	DEFINICIÓN
Banc_Codigo	varchar(3)	Código del banco **(PK)**
Banc_Descripcio	Varchar(60)	Descripción del banco
Banc_Numer_Cuen	Varchar(10)	Número de cuenta
Banc_Tipo_Cuen	varchar(1)	Tipo de cuenta:(A)horro,(C)orriente
Banc_Usuario	Varchar(15)	Código del usuario que modifica
Banc_FecUl_Modi	datetime	Fecha que se realizo la última modificación en el registro

Tabla 2.60. Campos de la tabla TBMM_BANCO

- **TBMC_ENTRE_CALI**

CAMPO	TIPO	DEFINICIÓN
EnCa_Anio_Lect	varchar(4)	Año lectivo **(PK)**
EnCa_Tipo_Curs	Varchar(6)	Código de tipo de curso **(PK)**
EnCa_Ciclo	Varchar(1)	Ciclo **(PK)**
EnCa_Tipo_Cali	varchar(6)	Código del tipo de calificación **(PK)**
EnCa_FecMa_Entr	datetime	Fecha Máxima de entrega de notas
Banc_Usuario	Varchar(15)	Código del usuario que modifica
Banc_FecUl_Modi	datetime	Fecha que se realizó la última modificación en el registro

Tabla 2.61. Campos de la tabla TBMM_ENTRE_CALI

2.2. Scripts de Creación de las Tablas

A continuación se detalla el contenido del archivo scripts.sql que contiene todas las sentencias para la creación de las tablas del sistema académico:

```
CREATE TABLE TBMM_Conce_Cobr (
	CoCo_Codigo		varchar(3) NOT NULL,
	CoCo_Descripcio		varchar(60) NOT NULL,
	CoCo_Usuario		varchar(15) NOT NULL,
	CoCo_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (CoCo_Codigo)
)
go


CREATE TABLE TBMN_Tipo_Proy (
	TiPr_Codigo		varchar(1) NOT NULL,
	TiPr_Descripcio		varchar(30) NOT NULL,
	TiPr_Usuario		varchar(15) NOT NULL,
	TiPr_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (TiPr_Codigo)
)
go


CREATE TABLE TBMG_Pais (
	Pais_Codigo		varchar(3) NOT NULL,
	Pais_Descripcio		varchar(60) NOT NULL,
	Pais_Nacionalid		varchar(30) NOT NULL,
	Pais_Usuario		varchar(15) NOT NULL,
	Pais_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (Pais_Codigo)
)
go


CREATE TABLE TBMG_Provincia (
	Prov_Codigo		varchar(3) NOT NULL,
```

```
	Prov_Pais		varchar(3) NOT NULL,
	Prov_Descripcio		varchar(60) NULL,
	Prov_Usuario		varchar(15) NULL,
	Prov_FecUl_Modi		datetime NULL,
	PRIMARY KEY (Prov_Codigo, Prov_Pais),
	FOREIGN KEY (Prov_Pais)
			REFERENCES TBMG_Pais
)
go


CREATE TABLE TBMG_Ciudad (
	Ciud_Codigo		varchar(3) NOT NULL,
	Ciud_Provincia		varchar(3) NOT NULL,
	Ciud_Pais		varchar(3) NOT NULL,
	Ciud_Descripcio		varchar(60) NOT NULL,
	Ciud_Usuario		varchar(15) NOT NULL,
	Ciud_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (Ciud_Codigo, Ciud_Provincia, Ciud_Pais),
	FOREIGN KEY (Ciud_Provincia, Ciud_Pais)
			REFERENCES TBMG_Provincia
)
go


CREATE TABLE TBMG_Universida (
	Univ_Codigo		varchar(3) NOT NULL,
	Univ_Descripcio		varchar(60) NOT NULL,
	Univ_Pais		varchar(3) NOT NULL,
	Univ_Provincia		varchar(3) NOT NULL,
	Univ_Ciudad		varchar(3) NOT NULL,
	Univ_Direccion		varchar(120) NULL,
	Univ_Telefono		varchar(10) NOT NULL,
	Univ_Usuario		varchar(15) NOT NULL,
	Univ_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (Univ_Codigo)
)
go


CREATE TABLE TBMD_Titul_Univ (
	TiUn_Codigo		varchar(3) NOT NULL,
	TiUn_Descripcio		varchar(60) NOT NULL,
```

```
    TiUn_Abreviatur     varchar(15) NOT NULL,
    TiUn_Usuario        varchar(15) NOT NULL,
    TiUn_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (TiUn_Codigo)
)
go


CREATE TABLE TBMD_Docente (
    Doce_Codigo         varchar(5) NOT NULL,
    Doce_Cedula         varchar(10) NOT NULL,
    Doce_Nombre         varchar(30) NOT NULL,
    Doce_Apellido       varchar(30) NOT NULL,
    Doce_Provi_Resi     varchar(3) NOT NULL,
    Doce_Ciuda_Resi     varchar(3) NOT NULL,
    Doce_Domicilio      varchar(120) NOT NULL,
    Doce_Telefono1      varchar(10) NULL,
    Doce_Telefono2      varchar(10) NULL,
    Doce_Direc_Trab     varchar(120) NULL,
    Doce_Telef_Trab     varchar(10) NULL,
    Doce_Nacionalid     varchar(3) NOT NULL,
    Doce_Provi_Naci     varchar(3) NOT NULL,
    Doce_Ciuda_Naci     varchar(3) NOT NULL,
    Doce_Fecha_Naci     datetime NOT NULL,
    Doce_Carne_Afil     varchar(10) NULL,
    Doce_Empresa        varchar(60) NULL,
    Doce_Universida     varchar(3) NOT NULL,
    Doce_Titul_Univ     varchar(3) NOT NULL,
    Doce_Logro_Obte     varchar(120) NULL,
    Doce_Cedul_Mili     varchar(10) NULL,
    Doce_Recor_Poli     varchar(10) NULL,
    Doce_Usuario        varchar(15) NOT NULL,
    Doce_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (Doce_Codigo),
    FOREIGN KEY (Doce_Universida)
                REFERENCES TBMG_Universida,
    FOREIGN KEY (Doce_Titul_Univ)
                REFERENCES TBMD_Titul_Univ
)
go


CREATE TABLE TBMA_Tipo_Subu (
```

```
    TiSu_Codigo         varchar(2) NOT NULL,
    TiSu_Descripcio     varchar(60) NOT NULL,
    TiSu_Usuario        varchar(15) NOT NULL,
    TiSu_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (TiSu_Codigo)
)
go


CREATE TABLE TBMA_Facultad (
    Facu_Codigo         varchar(2) NOT NULL,
    Facu_Nombre         varchar(60) NOT NULL,
    Facu_Decano         varchar(5) NOT NULL,
    Facu_Usuario        varchar(15) NOT NULL,
    Facu_Direccion      varchar(60) NOT NULL,
    Facu_FecUl_Modi     datetime NOT NULL,
    Facu_Ubicacion      varchar(60) NOT NULL,
    Facu_Observacio     varchar(120) NULL,
    PRIMARY KEY (Facu_Codigo),
    FOREIGN KEY (Facu_Decano)
                REFERENCES TBMD_Docente
)
go


CREATE TABLE TBMA_Subun_Acad (
    SuAc_Codigo         varchar(4) NOT NULL,
    SuAc_Tipo           varchar(2) NOT NULL,
    SuAc_Facultad       varchar(2) NOT NULL,
    SuAc_Nombre         varchar(60) NOT NULL,
    SuAc_Director       varchar(5) NOT NULL,
    SuAc_Nivel          varchar(2) NOT NULL,
    SuAc_Telefono1      varchar(10) NULL,
    SuAc_Telefono2      varchar(10) NULL,
    SuAc_Usuario        varchar(15) NOT NULL,
    SuAc_FecUl_Modi     datetime NOT NULL,
    SuAc_Periodicid     varchar(1) NOT NULL,
    PRIMARY KEY (SuAc_Codigo),
    FOREIGN KEY (SuAc_Director)
                REFERENCES TBMD_Docente,
    FOREIGN KEY (SuAc_Tipo)
                REFERENCES TBMA_Tipo_Subu,
    FOREIGN KEY (SuAc_Facultad)
```

```
                    REFERENCES TBMA_Facultad
)
go


CREATE TABLE TBMA_Area_Acad (
      ArAc_Codigo         varchar(3) NOT NULL,
      ArAc_Subunidad      varchar(4) NOT NULL,
      ArAc_Coordinado     varchar(5) NOT NULL,
      ArAc_Descripcio     varchar(60) NULL,
      ArAc_Usuario        varchar(15) NOT NULL,
      ArAc_FecUl_Modi     datetime NOT NULL,
      PRIMARY KEY (ArAc_Codigo, ArAc_Subunidad),
      FOREIGN KEY (ArAc_Subunidad)
                    REFERENCES TBMA_Subun_Acad,
      FOREIGN KEY (ArAc_Coordinado)
                    REFERENCES TBMD_Docente
)
go


CREATE TABLE TBMI_Tipo_Curs (
      TiCu_Codigo         varchar(6) NOT NULL,
      TiCu_Descripcio     varchar(60) NOT NULL,
      TiCu_Subunidad      varchar(4) NOT NULL,
      TiCu_Numer_Parc     numeric(2) NOT NULL,
      TiCu_NumEx_Adic     numeric(2) NOT NULL,
      TiCu_MinPa_Susp     numeric(2) NOT NULL,
      TiCu_MinAp_Asig     numeric(2) NOT NULL,
      TiCu_MinAp_Susp     numeric(2) NOT NULL,
      TiCu_MinPo_Asis     numeric(2) NOT NULL,
      TiCu_Usuario        varchar(15) NOT NULL,
      TiCu_FecUl_Modi     datetime NOT NULL,
      PRIMARY KEY (TiCu_Codigo),
      FOREIGN KEY (TiCu_Subunidad)
                    REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMM_Perio_Lect (
      PeLe_Fecha_Inic     datetime NULL,
      PeLe_Tipo_Curs      varchar(6) NOT NULL,
```

```
        PeLe_Estado         varchar(1) NOT NULL,
        PeLe_Ciclo          varchar(2) NOT NULL,
        PeLe_Fecha_Fin      datetime NULL,
        PeLe_MaxNu_Cred     numeric(1) NULL,
        PeLe_Fecha_Matr     datetime NULL,
        PeLe_Anio_Lect      varchar(4) NOT NULL,
        PeLe_Usuario        varchar(15) NOT NULL,
        PeLe_UltNu_Reci     varchar(4) NOT NULL,
        PeLe_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (PeLe_Anio_Lect, PeLe_Tipo_Curs, PeLe_Ciclo),
        FOREIGN KEY (PeLe_Tipo_Curs)
                        REFERENCES TBMI_Tipo_Curs
)
go


CREATE TABLE TBMA_MotCa_Pens (
        MoPe_Codigo         varchar(2) NOT NULL,
        MoPe_Descripcio     varchar(30) NOT NULL,
        MoPe_Usuario        varchar(15) NOT NULL,
        MoPe_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (MoPe_Codigo)
)
go


CREATE TABLE TBME_Especializ (
        Espe_Codigo         varchar(2) NOT NULL,
        Espe_Descripcio     varchar(30) NOT NULL,
        Espe_Usuario        varchar(15) NOT NULL,
        Espe_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (Espe_Codigo)
)
go


CREATE TABLE TBMM_Categoria (
        Cate_Codigo         varchar(3) NOT NULL,
        Cate_Observacio     varchar(30) NULL,
        Cate_Usuario        varchar(15) NOT NULL,
        Cate_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (Cate_Codigo)
)
```

```
go


CREATE TABLE TBMM_Colegio (
      Cole_Codigo          varchar(4) NOT NULL,
      Cole_Categoria       varchar(3) NOT NULL,
      Cole_Nombre          varchar(60) NOT NULL,
      Cole_Tipo           varchar(1) NOT NULL,
      Cole_Usuario         varchar(15) NOT NULL,
      Cole_FecUl_Modi      datetime NOT NULL,
      PRIMARY KEY (Cole_Codigo),
      FOREIGN KEY (Cole_Categoria)
                  REFERENCES TBMM_Categoria
)
go


CREATE TABLE TBME_Estudiante (
      Estu_Codigo          varchar(6) NOT NULL,
      Estu_Colegio         varchar(4) NOT NULL,
      Estu_Cedula          varchar(10) NOT NULL,
      Estu_Apellido        varchar(30) NOT NULL,
      Estu_Nombre          varchar(30) NOT NULL,
      Estu_Nacionalid      varchar(3) NOT NULL,
      Estu_Sexo           varchar(1) NOT NULL,
      Estu_Provi_Naci      varchar(3) NOT NULL,
      Estu_Ciuda_Naci      varchar(3) NOT NULL,
      Estu_Fecha_Naci      datetime NOT NULL,
      Estu_Estad_Civi      varchar(1) NOT NULL,
      Estu_Provi_Domi      varchar(3) NOT NULL,
      Estu_Ciuda_Domi      varchar(3) NOT NULL,
      Estu_Direc_Domi      varchar(120) NOT NULL,
      Estu_Telef_Domi      varchar(10) NULL,
      Estu_Telef_Trab      varchar(10) NULL,
      Estu_Nombr_Padr      varchar(60) NOT NULL,
      Estu_Categoria       varchar(3) NOT NULL,
      Estu_Nombr_Madr      varchar(60) NOT NULL,
      Estu_Especializ      varchar(2) NOT NULL,
      Estu_Anio_Grad       varchar(4) NOT NULL,
      Estu_Nota_Grad       varchar(2) NOT NULL,
      Estu_Nombr_Empr      varchar(60) NULL,
      Estu_Universida      varchar(3) NULL,
      Estu_Direc_Empr      varchar(120) NULL,
```

```
    Estu_Carnet_IESS     varchar(10) NULL,
    Estu_Email           varchar(40) NULL,
    Estu_Grupo_Sang      varchar(2) NOT NULL,
    Estu_Cedul_Mili      varchar(10) NULL,
    Estu_Recor_Poli      varchar(10) NULL,
    Estu_Observacio      varchar(120) NULL,
    Estu_Exame_Grac      numeric(2) NOT NULL,
    Estu_Usuario         varchar(15) NOT NULL,
    Estu_FecUl_Modi      datetime NOT NULL,
    PRIMARY KEY (Estu_Codigo),
    FOREIGN KEY (Estu_Universida)
                 REFERENCES TBMG_Universida,
    FOREIGN KEY (Estu_Especializ)
                 REFERENCES TBME_Especializ,
    FOREIGN KEY (Estu_Colegio)
                 REFERENCES TBMM_Colegio
)
go


CREATE TABLE TBMI_Estud_Subu (
    EsSu_CodEs_Subu      varchar(10) NOT NULL,
    EsSu_Subunidad       varchar(4) NOT NULL,
    EsSu_Estudiante      varchar(6) NOT NULL,
    EsSu_UltNi_Apro      varchar(2) NOT NULL,
    EsSu_Usuario         varchar(15) NOT NULL,
    EsSu_Pensu_Actu      varchar(6) NOT NULL,
    EsSu_Cruce_Perm      tinyint NOT NULL,
    EsSu_Fecha_Ingr      datetime NOT NULL,
    EsSu_FecUl_Modi      datetime NOT NULL,
    EsSu_Fecha_Egre      datetime NULL,
    EsSu_Condi_Acad      varchar(1) NOT NULL,
    EsSu_Prome_Egre      varchar(9) NULL,
    EsSu_Prome_Grad      varchar(9) NULL,
    EsSu_Prome_Fina      varchar(9) NULL,
    EsSu_Proye_Grad      varchar(4) NULL,
    EsSu_NumPr_Grad      tinyint NULL,
    EsSu_Observacio      varchar(120) NULL,
    PRIMARY KEY (EsSu_CodEs_Subu),
    FOREIGN KEY (EsSu_Subunidad)
                 REFERENCES TBMA_Subun_Acad,
    FOREIGN KEY (EsSu_Estudiante)
                 REFERENCES TBME_Estudiante
```

```
)
go


CREATE TABLE TBMA_Pensum (
      Pens_Codigo          varchar(6) NOT NULL,
      Pens_Subunidad       varchar(4) NOT NULL,
      Pens_Descripcio      varchar(60) NOT NULL,
      Pens_Usuario         varchar(15) NOT NULL,
      Pens_Fecha_Inic      datetime NOT NULL,
      Pens_FecUl_Modi      datetime NOT NULL,
      Pens_Fecha_Cadu      datetime NOT NULL,
      Pens_Estado          varchar(1) NOT NULL,
      PRIMARY KEY (Pens_Codigo),
      FOREIGN KEY (Pens_Subunidad)
                  REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMI_Estud_Pens (
      EsPe_Estud_Subu      varchar(10) NOT NULL,
      EsPe_Pensu_Actu      varchar(6) NOT NULL,
      EsPe_Pensu_Ante      varchar(6) NOT NULL,
      EsPe_MotCa_Pens      varchar(2) NOT NULL,
      EsPe_Usuario         varchar(15) NOT NULL,
      EsPe_FecUl_Modi      datetime NOT NULL,
      PRIMARY KEY (EsPe_Estud_Subu, EsPe_Pensu_Actu),
      FOREIGN KEY (EsPe_MotCa_Pens)
                  REFERENCES TBMA_MotCa_Pens,
      FOREIGN KEY (EsPe_Estud_Subu)
                  REFERENCES TBMI_Estud_Subu,
      FOREIGN KEY (EsPe_Pensu_Actu)
                  REFERENCES TBMA_Pensum
)
go


CREATE TABLE TBMM_Incre_Matr (
      InMa_Tipo_Matr       varchar(2) NOT NULL,
      InMa_Porce_Incr      numeric(8,2) NOT NULL,
      InMa_Subunidad       varchar(4) NOT NULL,
      InMa_Descripcio      varchar(30) NOT NULL,
```

```
        InMa_Usuario        varchar(15) NOT NULL,
        InMa_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (InMa_Tipo_Matr, InMa_Subunidad),
        FOREIGN KEY (InMa_Subunidad)
                        REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMM_Descu_Aran (
        DeAr_Codigo         varchar(3) NOT NULL,
        DeAr_Descripcio     varchar(60) NOT NULL,
        DeAr_Tipo           varchar(1) NOT NULL,
        DeAr_Valor          numeric(8,2) NOT NULL,
        DeAr_Usuario        varchar(15) NOT NULL,
        DeAr_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (DeAr_Codigo)
)
go


CREATE TABLE TBMM_Banco (
        Banc_Codigo         varchar(3) NOT NULL,
        Banc_Descripcio     varchar(60) NOT NULL,
        Banc_Numer_Cuen     varchar(10) NOT NULL,
        Banc_Tipo_Cuen      varchar(1) NOT NULL,
        Banc_Usuario        varchar(15) NOT NULL,
        Banc_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (Banc_Codigo)
)
go


CREATE TABLE TBMI_Estud_Nive (
        EsNi_Estud_Subu     varchar(10) NOT NULL,
        EsNi_Numer_Matr     varchar(10) NOT NULL,
        EsNi_Tipo_Curs      varchar(6) NULL,
        EsNi_Anio_Lect      varchar(4) NOT NULL,
        EsNi_Ciclo          varchar(2) NOT NULL,
        EsNi_Nivel          varchar(2) NOT NULL,
        EsNi_Numer_Vez      varchar(1) NOT NULL,
        EsNi_Asign_Toma     tinyint NOT NULL,
        EsNi_Promedio       varchar(9) NULL,
```

```
    EsNi_Numer_Susp      tinyint NULL,
    EsNi_Asign_Apro      tinyint NULL,
    EsNi_Asign_Repr      tinyint NULL,
    EsNi_Premiado        varchar(1) NOT NULL,
    EsNi_Sancionado      varchar(1) NOT NULL,
    EsNi_Observacio      varchar(120) NOT NULL,
    EsNi_Usuario         varchar(15) NOT NULL,
    EsNi_FecUl_Modi      datetime NOT NULL,
    PRIMARY KEY (EsNi_Estud_Subu, EsNi_Numer_Matr),
    FOREIGN KEY (EsNi_Tipo_Curs)
                REFERENCES TBMI_Tipo_Curs,
    FOREIGN KEY (EsNi_Estud_Subu)
                REFERENCES TBMI_Estud_Subu
)
go


CREATE TABLE TBMM_Orden_Pago (
    OrPa_Estud_Subu      varchar(10) NOT NULL,
    OrPa_Numer_Matr      varchar(10) NOT NULL,
    OrPa_Fecha_Emis      datetime NOT NULL,
    OrPa_Conce_Cobr      varchar(3) NOT NULL,
    OrPa_Banco           varchar(3) NOT NULL,
    OrPa_Valor_Total     numeric(8,2) NOT NULL,
    OrPa_Numer_Reci      varchar(4) NOT NULL,
    OrPa_Observacio      varchar(120) NULL,
    OrPa_Usuario         varchar(15) NOT NULL,
    OrPa_FecUl_Modi      datetime NOT NULL,
    PRIMARY KEY (OrPa_Estud_Subu, OrPa_Numer_Matr),
    FOREIGN KEY (OrPa_Conce_Cobr)
                REFERENCES TBMM_Conce_Cobr,
    FOREIGN KEY (OrPa_Banco)
                REFERENCES TBMM_Banco,
    FOREIGN KEY (OrPa_Estud_Subu, OrPa_Numer_Matr)
                REFERENCES TBMI_Estud_Nive
)
go


CREATE TABLE TBMM_Cuota (
    Cuot_Estud_Subu      varchar(10) NOT NULL,
    Cuot_Numer_Matr      varchar(10) NOT NULL,
    Cuot_Número          varchar(2) NOT NULL,
```

```
	Cuot_Valor		numeric(8,2) NOT NULL,
	Cuot_Fecha_Venc		datetime NOT NULL,
	Cuot_Estado		varchar(1) NOT NULL,
	Cuot_Usuario		varchar(15) NOT NULL,
	Cuot_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (Cuot_Estud_Subu, Cuot_Numer_Matr,
Cuot_Número),
	FOREIGN KEY (Cuot_Estud_Subu, Cuot_Numer_Matr)
			REFERENCES TBMM_Orden_Pago
)
go


CREATE TABLE TBMM_Detal_Matr (
	DeMa_Estud_Subu		varchar(10) NOT NULL,
	DeMa_Numer_Matr		varchar(10) NOT NULL,
	DeMa_Tipo		varchar(1) NOT NULL,
	DeMa_Motivo		varchar(3) NOT NULL,
	DeMa_Monto		numeric(8,2) NOT NULL,
	DeMa_Valor_Calc		numeric(8,2) NOT NULL,
	DeMa_Usuario		varchar(15) NOT NULL,
	DeMa_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (DeMa_Estud_Subu, DeMa_Numer_Matr,
DeMa_Motivo),
	FOREIGN KEY (DeMa_Motivo)
			REFERENCES TBMM_Descu_Aran,
	FOREIGN KEY (DeMa_Estud_Subu, DeMa_Numer_Matr)
			REFERENCES TBMM_Orden_Pago
)
go


CREATE TABLE TBMM_Matri_Asig (
	MaAs_Categoria		varchar(3) NOT NULL,
	MaAs_Valor		numeric(8,2) NOT NULL,
	MaAs_Subunidad		varchar(4) NOT NULL,
	MaAs_Limite		numeric(1) NOT NULL,
	MaAs_Usuario		varchar(15) NOT NULL,
	MaAs_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (MaAs_Categoria, MaAs_Subunidad),
	FOREIGN KEY (MaAs_Categoria)
			REFERENCES TBMM_Categoria,
	FOREIGN KEY (MaAs_Subunidad)
```

```
                    REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMM_Matri_Glob (
    MaGl_Categoria      varchar(3) NOT NULL,
    MaGl_Valor          numeric(8,2) NOT NULL,
    MaGl_Subunidad      varchar(4) NOT NULL,
    MaGl_Usuario        varchar(15) NOT NULL,
    MaGl_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (MaGl_Categoria, MaGl_Subunidad),
    FOREIGN KEY (MaGl_Categoria)
                    REFERENCES TBMM_Categoria,
    FOREIGN KEY (MaGl_Subunidad)
                    REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMA_Asignatura (
    Asig_Codigo         varchar(7) NOT NULL,
    Asig_Nombre         varchar(30) NOT NULL,
    Asig_Numer_Punt     varchar(2) NOT NULL,
    Asig_Nivel          varchar(2) NOT NULL,
    Asig_Area_Acad      varchar(3) NULL,
    Asig_Subunidad      varchar(4) NOT NULL,
    Asig_Horas_Sema     varchar(2) NOT NULL,
    Asig_Tipo           varchar(1) NULL,
    Asig_Usuario        varchar(15) NOT NULL,
    Asig_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (Asig_Codigo),
    FOREIGN KEY (Asig_Subunidad)
                    REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMI_Estud_Asig (
    EsAs_Asignatura     varchar(7) NOT NULL,
    EsAs_Estud_Subu     varchar(10) NOT NULL,
    EsAs_Numer_Cred     numeric(1) NOT NULL,
    EsAs_Grupo          varchar(8) NOT NULL,
```

```
        EsAs_Tipo_Curs      varchar(6) NOT NULL,
        EsAs_Subgrupo       varchar(1) NULL,
        EsAs_Aprobado       varchar(1) NOT NULL,
        EsAs_Promedio       varchar(9) NULL,
        EsAs_Anulada        varchar(1) NOT NULL,
        EsAs_Anio_Lect      varchar(4) NOT NULL,
        EsAs_Ciclo          varchar(2) NOT NULL,
        EsAs_Observacio     varchar(120) NULL,
        EsAs_Usuario        varchar(15) NOT NULL,
        EsAs_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (EsAs_Estud_Subu, EsAs_Asignatura,
EsAs_Numer_Cred),
        FOREIGN KEY (EsAs_Asignatura)
                        REFERENCES TBMA_Asignatura,
        FOREIGN KEY (EsAs_Estud_Subu)
                        REFERENCES TBMI_Estud_Subu
)
go


CREATE TABLE TBMC_Calif_Prom (
        CaPr_Estud_Subu     varchar(10) NOT NULL,
        CaPr_Asignatura     varchar(7) NOT NULL,
        CaPr_Numer_Cred     numeric(1) NOT NULL,
        CaPr_Prome_Cali     varchar(9) NOT NULL,
        CaPr_Prome_Asis     varchar(9) NOT NULL,
        CaPr_Observacio     varchar(120) NULL,
        CaPr_Usuario        varchar(15) NOT NULL,
        CaPr_FecUl_Modi     datetime NOT NULL,
        PRIMARY KEY (CaPr_Estud_Subu, CaPr_Asignatura,
CaPr_Numer_Cred),
        FOREIGN KEY (CaPr_Estud_Subu, CaPr_Asignatura,
CaPr_Numer_Cred)
                        REFERENCES TBMI_Estud_Asig
)
go


CREATE TABLE TBMC_Tipo_Soli (
        TiSo_Codigo         varchar(2) NOT NULL,
        TiSo_Descripcio     varchar(60) NOT NULL,
        TiSo_Usuario        varchar(15) NOT NULL,
        TiSo_FecUl_Modi     datetime NOT NULL,
```

```
        PRIMARY KEY (TiSo_Codigo)
)
go


CREATE TABLE TBMC_Solic_Espe (
        SoEs_Estud_Subu      varchar(10) NOT NULL,
        SoEs_Numer_Cred      numeric(1) NOT NULL,
        SoEs_Asignatura      varchar(7) NOT NULL,
        SoEs_Tipo_Soli       varchar(2) NOT NULL,
        SoEs_Numer_Soli      varchar(10) NOT NULL,
        SoEs_Fecha_Apro      datetime NOT NULL,
        SoEs_Observacio      varchar(120) NOT NULL,
        SoEs_Usuario         varchar(15) NOT NULL,
        SoEs_FecUl_Modi      datetime NOT NULL,
        PRIMARY KEY (SoEs_Estud_Subu, SoEs_Asignatura,
SoEs_Numer_Cred,
                SoEs_Numer_Soli),
        FOREIGN KEY (SoEs_Estud_Subu, SoEs_Asignatura,
SoEs_Numer_Cred)
                        REFERENCES TBMI_Estud_Asig,
        FOREIGN KEY (SoEs_Tipo_Soli)
                        REFERENCES TBMC_Tipo_Soli
)
go


CREATE TABLE TBMC_Tipo_Cali (
        TiCa_Codigo          varchar(6) NOT NULL,
        TiCa_Descripcio      varchar(60) NOT NULL,
        TiCa_Subunidad       varchar(4) NOT NULL,
        TiCa_Usuario         varchar(15) NOT NULL,
        TiCa_Reemp_Nota      varchar(1) NOT NULL,
        TiCa_FecUl_Modi      datetime NOT NULL,
        TiCa_Suspenso        varchar(1) NOT NULL,
        PRIMARY KEY (TiCa_Codigo),
        FOREIGN KEY (TiCa_Subunidad)
                        REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMC_Calif_Parc (
```

```
	CaPa_Estud_Subu	varchar(10) NOT NULL,
	CaPa_Numer_Cred	numeric(1) NOT NULL,
	CaPa_Asignatura	varchar(7) NOT NULL,
	CaPa_Tipo_Cali	varchar(6) NOT NULL,
	CaPa_Calif_Trab	varchar(9) NULL,
	CaPa_Fecha_Entr	datetime NOT NULL,
	CaPa_Calif_Exam	varchar(9) NULL,
	CaPa_Calif_Tota	varchar(9) NOT NULL,
	CaPa_Estad_Regi	varchar(1) NOT NULL,
	CaPa_Porce_Asis	varchar(9) NOT NULL,
	CaPa_Observacio	varchar(120) NULL,
	CaPa_Usuario	varchar(15) NOT NULL,
	CaPa_FecUl_Modi	datetime NOT NULL,
	PRIMARY KEY (CaPa_Estud_Subu, CaPa_Asignatura,
CaPa_Numer_Cred,
		CaPa_Tipo_Cali),
	FOREIGN KEY (CaPa_Estud_Subu, CaPa_Asignatura,
CaPa_Numer_Cred)
			REFERENCES TBMI_Estud_Asig,
	FOREIGN KEY (CaPa_Tipo_Cali)
			REFERENCES TBMC_Tipo_Cali
)
go


CREATE TABLE TBMI_Docen_Asig (
	DoAs_Asignatura	varchar(7) NOT NULL,
	DoAs_Docente	varchar(5) NOT NULL,
	DoAs_Tipo_Curs	varchar(6) NOT NULL,
	DoAs_Grupo	varchar(8) NOT NULL,
	DoAs_Cupo_Estu	varchar(2) NOT NULL,
	DoAs_Anio_Lect	varchar(4) NOT NULL,
	DoAs_Ciclo	varchar(2) NOT NULL,
	DoAs_Usuario	varchar(15) NOT NULL,
	DoAs_FecUl_Modi	datetime NOT NULL,
	FOREIGN KEY (DoAs_Docente)
			REFERENCES TBMD_Docente,
	FOREIGN KEY (DoAs_Asignatura)
			REFERENCES TBMA_Asignatura
)
go
```

```
CREATE TABLE TBMI_Horar_Grup (
      HoGr_Grupo          varchar(8) NOT NULL,
      HoGr_Codigo         varchar(2) NOT NULL,
      HoGr_Asignatura     varchar(7) NOT NULL,
      HoGr_Seccion        varchar(1) NOT NULL,
      HoGr_Dia_Sema       varchar(1) NOT NULL,
      HoGr_Hora_Clas      varchar(1) NOT NULL,
      HoGr_Descr_Hora     varchar(15) NULL,
      HoGr_Usuario        varchar(15) NOT NULL,
      HoGr_FecUl_Modi     datetime NOT NULL,
      PRIMARY KEY (HoGr_Codigo, HoGr_Grupo),
      FOREIGN KEY (HoGr_Asignatura)
                  REFERENCES TBMA_Asignatura
)
go


CREATE TABLE TBMI_Horar_Estu (
      HoEs_Estud_Subu     varchar(10) NOT NULL,
      HoEs_Horar_Grup     varchar(2) NOT NULL,
      HoEs_Grupo          varchar(8) NOT NULL,
      HoEs_Usuario        varchar(15) NOT NULL,
      HoEs_FecUl_Modi     datetime NOT NULL,
      PRIMARY KEY (HoEs_Horar_Grup, HoEs_Grupo,
HoEs_Estud_Subu),
      FOREIGN KEY (HoEs_Horar_Grup, HoEs_Grupo)
                  REFERENCES TBMI_Horar_Grup,
      FOREIGN KEY (HoEs_Estud_Subu)
                  REFERENCES TBMI_Estud_Subu
)
go


CREATE TABLE TBMA_Prerequisi (
      Prer_Asignatura     varchar(7) NOT NULL,
      Prer_Pensum         varchar(6) NOT NULL,
      Prer_Requisito      varchar(7) NULL,
      Prer_Usuario        varchar(15) NOT NULL,
      Prer_Nivel          varchar(2) NULL,
      Prer_FecUl_Modi     datetime NOT NULL,
      FOREIGN KEY (Prer_Pensum)
                  REFERENCES TBMA_Pensum,
      FOREIGN KEY (Prer_Asignatura)
```

```
                    REFERENCES TBMA_Asignatura
)
go


CREATE TABLE TBMS_Perfil (
      Perf_Codigo         varchar(2) NOT NULL,
      Perf_Descripcio     varchar(30) NOT NULL,
      PRIMARY KEY (Perf_Codigo)
)
go


CREATE TABLE TBMS_Usuario (
      Usua_Usuario        varchar(15) NOT NULL,
      Usua_Perfil         varchar(2) NOT NULL,
      Usua_Apellido       varchar(30) NOT NULL,
      Usua_Nombre         varchar(30) NOT NULL,
      Usua_Estado         varchar(1) NOT NULL,
      Usua_Caducidad      int NOT NULL,
      Usua_Fecha_Ingr     datetime NOT NULL,
      Usua_Subunidad      varchar(4) NULL,
      PRIMARY KEY (Usua_Usuario),
      FOREIGN KEY (Usua_Perfil)
                    REFERENCES TBMS_Perfil
)
go


CREATE TABLE TBMS_Pantalla (
      Pant_Codigo         varchar(6) NOT NULL,
      Pant_Descripcio     varchar(60) NOT NULL,
      PRIMARY KEY (Pant_Codigo)
)
go


CREATE TABLE TBMS_Permiso (
      Perm_Pantalla       varchar(6) NOT NULL,
      Perm_Perfil         varchar(2) NOT NULL,
      Perm_Tipo_Acce      varchar(3) NOT NULL,
      PRIMARY KEY (Perm_Pantalla, Perm_Perfil),
      FOREIGN KEY (Perm_Pantalla)
```

```
                    REFERENCES TBMS_Pantalla,
    FOREIGN KEY (Perm_Perfil)
                    REFERENCES TBMS_Perfil
)
go


CREATE TABLE TBME_Foto_Estu (
    FoEs_Estudiante     varchar(6) NOT NULL,
    FoEs_Imagen         image NULL,
    FoEs_Usuario        varchar(15) NULL,
    FoEs_FecUl_Modi     datetime NULL,
    PRIMARY KEY (FoEs_Estudiante),
    FOREIGN KEY (FoEs_Estudiante)
                    REFERENCES TBME_Estudiante
)
go


CREATE TABLE TBMM_Autoridad (
    Auto_Subunidad      varchar(4) NOT NULL,
    Auto_Codigo         varchar(2) NOT NULL,
    Auto_Cargo          varchar(30) NOT NULL,
    Auto_Apellido       varchar(30) NOT NULL,
    Auto_Nombre         varchar(30) NOT NULL,
    Auto_Firma          image NULL,
    Auto_Usuario        varchar(15) NOT NULL,
    Auto_FecUl_Modi     datetime NOT NULL,
    PRIMARY KEY (Auto_Codigo, Auto_Subunidad),
    FOREIGN KEY (Auto_Subunidad)
                    REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMN_Proye_Grad (
    PrGr_Codigo         varchar(4) NOT NULL,
    PrGr_Subunidad      varchar(4) NOT NULL,
    PrGr_Tipo           varchar(1) NOT NULL,
    PrGr_Descripcio     varchar(120) NOT NULL,
    PrGr_Estado         varchar(1) NULL,
    PrGr_Fecha_Inic     datetime NOT NULL,
    PrGr_Fecha_Fin      datetime NULL,
```

```
	PrGr_Usuario		varchar(15) NOT NULL,
	PrGr_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (PrGr_Codigo, PrGr_Subunidad),
	FOREIGN KEY (PrGr_Tipo)
			REFERENCES TBMN_Tipo_Proy,
	FOREIGN KEY (PrGr_Subunidad)
			REFERENCES TBMA_Subun_Acad
)
go


CREATE TABLE TBMC_Entre_Cali (
	EnCa_Anio_Lect		varchar(4) NOT NULL,
	EnCa_Tipo_Curs		varchar(6) NOT NULL,
	EnCa_Ciclo		varchar(2) NOT NULL,
	EnCa_Tipo_Cali		varchar(6) NOT NULL,
	EnCa_FecMa_Entr		datetime NOT NULL,
	EnCa_Usuario		varchar(15) NOT NULL,
	EnCa_FecUl_Modi		datetime NOT NULL,
	PRIMARY KEY (EnCa_Anio_Lect, EnCa_Tipo_Curs, EnCa_Ciclo,
	    EnCa_Tipo_Cali),
	FOREIGN KEY (EnCa_Anio_Lect, EnCa_Tipo_Curs, EnCa_Ciclo)
			REFERENCES TBMM_Perio_Lect
)
go
```

2.3. Stored Procedures

A continuación se detallan los procedimientos almacenados que son empleados para insertar, modificar o eliminar registros en cada una de las tablas del sistema académico:

```
CREATE  PROCEDURE SPMA_APERTURA_PENSUM
@INAnio_Lect		VARCHAR(4)	,
@INCiclo		VARCHAR(2)	,
@INTipo_Curs		VARCHAR(6)	,
@OUTSql			VARCHAR(2) OUTPUT
```

```
AS

DECLARE @error_var int, @rowcount_var int, @fecha_matr datetime

    Select @fecha_matr=PeLe_Fecha_Matr
    From TBMM_Perio_Lect
    Where PeLe_Anio_Lect=@InAnio_Lect
    And PeLe_Ciclo=@InCiclo
    And PeLe_Tipo_Curs=@INTipo_Curs

    Update TBMA_Pensum
    set Pens_estado='V'
    where Pens_Fecha_Inic<=@fecha_matr

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       if (@rowcount_var = 1)
          Set @OutSql = 'OK'
       else
          Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_AREA_ACAD_INSERTA
@INSubunidad          VARCHAR(4)        ,
@INDescripcio         VARCHAR(60)       ,
@INCoordinado         VARCHAR(5)        ,
@INUsuario            VARCHAR(15)       ,
@OUTCodigo            VARCHAR(3) OUTPUT,
@OUTSql               VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(ArAc_Codigo),0)+1 as
VarChar(3)))
From TBMA_Area_Acad
```

```
set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMA_Area_Acad
(Arac_Codigo, Arac_subunidad, Arac_Coordinado, Arac_descripcio,
Arac_Usuario, Arac_FecUl_Modi)
Values
(@OutCodigo, @InSubunidad, @InCoordinado, @InDescripcio,
@InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE PROCEDURE SPMA_AREA_ACAD_MODIFICA
@INCodigo           VARCHAR(3)      ,
@INSubunidad        VARCHAR(4)      ,
@INDescripcio       VARCHAR(60)     ,
@INCoordinado       VARCHAR(5)      ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_Area_Acad
Set
   Arac_Descripcio     = @indescripcio,
   Arac_Subunidad      = @insubunidad,
   Arac_Coordinado     = @incoordinado,
   Arac_Usuario = @InUsuario,
   Arac_FecUl_Modi     = getdate()
Where  Arac_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
```

```
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_ASIGNATURA_INSERTA
@INNombre              VARCHAR(30)       ,
@INArea_Acad           VARCHAR(3)        ,
@INNivel          VARCHAR(2)        ,
@INHoras_Sema                VARCHAR(2)        ,
@INTipo                VARCHAR(1)        ,
@INNumer_Punt                VARCHAR(2)        ,
@INSubUnidad           VARCHAR(4)        ,
@INUsuario             VARCHAR(15)       ,
@OUTCodigo             VARCHAR(7) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Asig_Codigo),0)+1 as
VarChar(7)))
From TBMA_Asignatura
Where Asig_Subunidad = @InSubunidad

set @OutCodigo = replicate('0', 7 - len(@OutCodigo) ) + @OutCodigo

if @OutCodigo = '0000001'
  begin
    set @OutCodigo = @InSubUnidad + '001'
  end

Insert into TBMA_Asignatura
(Asig_Codigo, Asig_Nombre, Asig_Area_Acad, Asig_Nivel,
Asig_Horas_Sema, Asig_Tipo, Asig_Numer_Punt, Asig_Subunidad,
Asig_Usuario, Asig_FecUl_Modi)
Values
(@OutCodigo, @InNombre, @InArea_Acad, @InNivel, @InHoras_Sema,
@InTipo, @InNumer_Punt,  @InSubunidad, @InUsuario, getdate())

SELECT @error_var = @@ERROR
```

```
If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMA_ASIGNATURA_MODIFICA
@INCodigo               VARCHAR(7)     ,
@INNombre               VARCHAR(30)       ,
@INArea_Acad            VARCHAR(3)        ,
@INNivel          VARCHAR(2)        ,
@INHoras_Sema                VARCHAR(2)        ,
@INTipo           VARCHAR(1)        ,
@INNumer_Punt                VARCHAR(2)        ,
@INSubUnidad            VARCHAR(4)        ,
@INUsuario              VARCHAR(15)       ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_Asignatura
Set
   Asig_Nombre          = @InNombre,
   Asig_Area_Acad       = @InArea_Acad,
   Asig_Nivel           = @InNivel,
   Asig_Horas_Sema   = @InHoras_Sema,
   Asig_Tipo            = @InTipo,
   Asig_Numer_Punt   = @InNumer_Punt,
   Asig_Usuario = @InUsuario,
   Asig_FecUl_Modi   = getdate()
Where Asig_SubUnidad       = @InSubunidad
and   Asig_Codigo       = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
```

```
    else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_CIERRE_PENSUM
@INAnio_Lect          VARCHAR(4)        ,
@INCiclo         VARCHAR(2)        ,
@INTipo_Curs          VARCHAR(6)        ,
@OUTSql               VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @fecha_fin datetime

    Select @fecha_fin=PeLe_Fecha_Fin
    From TBMM_Perio_Lect
    Where PeLe_Anio_Lect=@InAnio_Lect
    And PeLe_Ciclo=@InCiclo
    And PeLe_Tipo_Curs=@INTipo_Curs

    Update TBMA_Pensum
    set Pens_estado='I'
    where Pens_Fecha_Cadu<=@fecha_fin

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       if (@rowcount_var = 1)
          Set @OutSql = 'OK'
       else
          Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_FACULTAD_INSERTA
@INNombre             VARCHAR(60),
@INDecano             VARCHAR(5),
@INDireccion          VARCHAR(60),
@INUbicacion          VARCHAR(60),
@INObservacio         VARCHAR(120),
@INUsuario            VARCHAR(15),
```

```
@OUTCodigo          VARCHAR(2) OUTPUT,
@OUTSql                  VARCHAR(2) OUTPUT
AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Facu_Codigo),0)+1 as
VarChar(2)))
From TBMA_Facultad


set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMA_Facultad
(Facu_Codigo, Facu_Nombre, Facu_Decano, Facu_Direccion,
Facu_Ubicacion, Facu_Observacio, Facu_Usuario, Facu_FecUl_Modi)
Values
(@OutCodigo, @InNombre, @InDecano, @InDireccion, @InUbicacion,
@InObservacio, @InUsuario, getdate())

select @error_var=@@ERROR

if (@error_var <> 0)
 set @OUTSql = 'NO'
ELSE
 set @OUTSql = 'OK'


CREATE  PROCEDURE SPMA_FACULTAD_MODIFICA
@INCodigo           VARCHAR(2)     ,
@INNombre           VARCHAR(60)      ,
@INDecano           VARCHAR(5)       ,
@INDireccion        VARCHAR(60)      ,
@INUbicacion        VARCHAR(60)      ,
@INObservacio       VARCHAR(120)     ,
@INUsuario          VARCHAR(15)      ,
@OUTSql             VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_Facultad
```

```
Set
   Facu_Nombre                = @InNombre,
   Facu_Decano        = @InDecano,
   Facu_Direccion     = @InDireccion,
   Facu_Ubicacion     = @InUbicacion,
   Facu_Observacio    = @InObservacio,
   Facu_Usuario= @InUsuario,
   Facu_FecUl_Modi    = getdate()
Where Facu_Codigo    = @InCodigo


SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_MOTCA_PENS_INSERTA
@INDescripcio          VARCHAR(30)      ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(2) OUTPUT,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(MoPe_Codigo),0)+1 as
VarChar(2)))
From TBMA_MotCa_Pens

set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMA_MotCa_Pens
(MoPe_Codigo, MoPe_Descripcio,MoPe_Usuario, MoPe_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio,@InUsuario, getdate())
```

```
SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMA_MOTCA_PENS_MODIFICA
@INCodigo              VARCHAR(2)        ,
@INDescripcio          VARCHAR(30)       ,
@INUsuario             VARCHAR(15)       ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_MotCa_Pens
Set
   MoPe_Descripcio    = @InDescripcio,
   MoPe_Usuario       = @InUsuario,
   MoPe_FecUl_Modi  = getdate()
Where  MoPe_Codigo  = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE    PROCEDURE SPMA_PENSUM_INSERTA
@INSubunidad           VARCHAR(4)        ,
@INDescripcio          VARCHAR(60)       ,
@INFecha_Inic          datetime      ,
@INFecha_Cadu                datetime      ,
@INEstado              VARCHAR(1)        ,
@INUsuario             VARCHAR(15)       ,
```

```
@OUTCodigo          VARCHAR(6) OUTPUT,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @primero varchar(1), @AntCodigo varchar(6)

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Pens_Codigo),0)+1 as
VarChar(6)))
From TBMA_Pensum
Where Pens_Subunidad = @InSubunidad

set @OutCodigo = replicate('0', 6 - len(@OutCodigo) ) + @OutCodigo

Select @AntCodigo = RTrim(Cast(Coalesce(Max(Pens_Codigo),0) as
VarChar(6)))
From TBMA_Pensum
Where Pens_Subunidad = @InSubunidad

set @AntCodigo = replicate('0', 6 - len(@AntCodigo) ) + @AntCodigo

set @primero = 'n'

if @OutCodigo = '000001'
  begin
    set @OutCodigo = @InSubUnidad + '01'
          set @primero = 's'
  end
Insert into TBMA_Pensum
(Pens_Codigo, Pens_Subunidad, Pens_Descripcio, Pens_Fecha_Inic,
Pens_Fecha_Cadu, Pens_Estado, Pens_Usuario, Pens_FecUl_Modi)
Values
(@OutCodigo, @INSubunidad, @InDescripcio, @InFecha_Inic,
@InFecha_Cadu, @InEstado, @InUsuario, getdate())

if (@primero <> 's')
 begin
    Insert into TBMA_Prerequisi (Prer_Asignatura, Prer_Pensum,
Prer_Requisito, Prer_Nivel, Prer_Usuario, Prer_FecUl_Modi)
    Select Prer_Asignatura, @OutCodigo, Prer_Requisito, Prer_Nivel,
@InUsuario, getdate()
    From TBMA_Prerequisi
    Where Prer_Pensum = @AntCodigo
```

```
 end

SELECT @error_var = @@ERROR
If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMA_PENSUM_MODIFICA
@INCodigo              VARCHAR(6)       ,
@INSubunidad           VARCHAR(4)       ,
@INDescripcio          VARCHAR(60)      ,
@INFecha_Inic          datetime      ,
@INFecha_Cadu                datetime      ,
@INEstado              VARCHAR(1)       ,
@INUsuario             VARCHAR(15)      ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_Pensum
Set
   Pens_Descripcio     = @inDescripcio,
   Pens_Fecha_Inic     = @inFecha_Inic,
   Pens_Fecha_Cadu  = @InFecha_Cadu,
   Pens_Estado         = @InEstado,
   Pens_Usuario        = @InUsuario,
   Pens_FecUl_Modi   = getdate()
Where  Pens_SubUnidad= @InSubunidad
and Pens_Codigo       = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'
```

```
CREATE  PROCEDURE SPMA_PREREQUISI_ELIMINA
@INAsignatura          VARCHAR(7)        ,
@INPensum              VARCHAR(6)        ,
@INRequisito           VARCHAR(7)        ,
@INNivel         VARCHAR(2)        ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Delete from TBMA_Prerequisi
Where Prer_Asignatura        =       @InAsignatura
And Prer_Pensum      =       @InPensum
And Prer_Requisito   =       @InRequisito
And Prer_Nivel       =       @InNivel

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_PREREQUISI_INSERTA
@INAsignatura          VARCHAR(7)        ,
@INPensum              VARCHAR(6)        ,
@INRequisito           VARCHAR(7)        ,
@INNivel         VARCHAR(2)        ,
@INUsuario             VARCHAR(15)       ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @count int

--Previo a la inserción, reviso sino existe el registro
```

```
Select @count = count(*)
From TBMA_Prerequisi
Where Prer_Asignatura      =      @InAsignatura
And Prer_Pensum      =      @InPensum
And Prer_Requisito     =      @InRequisito
And Prer_Nivel          =      @InNivel

IF @count <> 0
   Set @OutSql = 'DU'

--Ahora reviso si la inserción es por nivel (valido que no exista requisitos
por asignatura)
IF RTRIM(@InNivel)<>'' and @count=0
  Begin
    Select @count = count(*)
    From TBMA_Prerequisi
    Where Prer_Asignatura     =      @InAsignatura
    And Prer_Pensum           =      @InPensum
    And Prer_Requisito        <>     ''
    And Prer_Nivel            =      ''

    IF @count <> 0
      Set @OutSql = 'NI'

  End


IF RTRIM(@InRequisito)<>'' and @count=0
  Begin
    Select @count = count(*)
    From TBMA_Prerequisi
    Where Prer_Asignatura     =      @InAsignatura
    And Prer_Pensum           =      @InPensum
    And Prer_Requisito        =      ''
    And Prer_Nivel            <>     ''

    IF @count <> 0
      Set @OutSql = 'AS'

  End

IF @count = 0
  Begin
```

```
   Insert into TBMA_Prerequisi
   (Prer_Asignatura, Prer_Pensum, Prer_Requisito, Prer_Nivel,
Prer_Usuario, Prer_FecUl_Modi)
   Values
   (@InAsignatura, @InPensum, @InRequisito, @InNivel, @InUsuario,
getdate())
   SELECT @error_var = @@ERROR

   If (@error_var <> 0)
     Set @OutSql = 'NO'
   else
     Set @OutSql = 'OK'
 End


CREATE  PROCEDURE SPMA_SUBUN_ACAD_INSERTA
@INTipo          VARCHAR(2)        ,
@INFacultad              VARCHAR(2)        ,
@INNombre                VARCHAR(60)       ,
@INDirector              VARCHAR(5)        ,
@INNivel         VARCHAR(2)        ,
@INTelefono1             VARCHAR(10)       ,
@INTelefono2             VARCHAR(10)       ,
@INPeriodicid            VARCHAR(1)        ,
@INUsuario               VARCHAR(15)       ,
@OUTCodigo               VARCHAR(4) OUTPUT,
@OUTSql                          VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(SuAc_Codigo),0)+1 as
VarChar(4)))
From TBMA_Subun_Acad Where SuAc_Facultad = @INFacultad

set @OutCodigo = replicate('0', 4 - len(@OutCodigo) ) + @OutCodigo
if @OutCodigo='0001'
--set @OutCodigo = @INFacultad + substring(@OutCodigo,3,2)
    begin
       set @OutCodigo = @INFacultad + '01'
    end
```

```
Insert Into TBMA_Subun_Acad
(SuAc_Codigo, SuAc_Tipo, SuAc_Facultad, SuAc_Nombre,
SuAc_Director, SuAc_Nivel, SuAc_Telefono1, SuAc_Telefono2,
SuAc_Periodicid, SuAc_Usuario, SuAc_FecUl_Modi)
Values
(@OutCodigo, @InTipo, @InFacultad, @InNombre, @InDirector,
@InNivel, @InTelefono1, @InTelefono2, @InPeriodicid, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMA_SUBUN_ACAD_MODIFICA
@INCodigo            VARCHAR(4)     ,
@INTipo         VARCHAR(2)        ,
@INFacultad          VARCHAR(2)        ,
@INNombre            VARCHAR(60)       ,
@INDirector          VARCHAR(5)        ,
@INNivel        VARCHAR(2)        ,
@INTelefono1         VARCHAR(10)       ,
@INTelefono2         VARCHAR(10)       ,
@INPeriodicid        VARCHAR(1)        ,
@INUsuario           VARCHAR(15)       ,
@OUTSql                   VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int
Update TBMA_Subun_Acad
Set
   SuAc_Tipo         = @InTipo,
   SuAc_Facultad     = @InFacultad,
   SuAc_Nombre             = @InNombre,
   SuAc_Director     = @InDirector,
   SuAc_Nivel        = @InNivel,
   SuAc_Telefono1    = @InTelefono1,
   SuAc_Telefono2    = @InTelefono2,
   SuAc_Periodicid   = @InPeriodicid,
```

```
    SuAc_Usuario        = @InUsuario,
    SuAc_FecUl_Modi     = getdate()
Where   SuAc_Codigo = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
    Set @OutSql = 'NO'
else
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMA_TIPO_SUBU_INSERTA
@INDescripcio           VARCHAR(60)       ,
@INUsuario              VARCHAR(15)       ,
@OUTCodigo              VARCHAR(2) OUTPUT,
@OUTSql                         VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiSu_Codigo),0)+1 as
VarChar(2)))
From TBMA_Tipo_Subu

set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMA_Tipo_Subu
(TiSu_Codigo, TiSu_Descripcio,  TiSu_Usuario, TiSu_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
    Set @OutSql = 'NO'
else
    Set @OutSql = 'OK'
```

```
CREATE  PROCEDURE SPMA_TIPO_SUBU_MODIFICA
@INCodigo               VARCHAR(2)      ,
@INDescripcio           VARCHAR(60)       ,
@INUsuario              VARCHAR(15)       ,
@OUTSql                         VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMA_Tipo_Subu
Set
   TiSu_Descripcio      = @InDescripcio,
   TiSu_Usuario = @InUsuario,
   TiSu_FecUl_Modi    = getdate()
Where Tisu_Codigo     = @InCodigo


SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMC_BLOQUEO_NOTAS
@INAnio_Lect            VARCHAR(4)        ,
@INCiclo          VARCHAR(2)        ,
@INTipo_Curs            VARCHAR(6)        ,
@OUTSql                 VARCHAR(2) OUTPUT
AS

DECLARE @error_var int, @rowcount_var int, @Estud_Subu
varchar(10), @Asignatura varchar(7), @Numer_Cred numeric(1),
@Promedio varchar(9), @MinAp_Asig numeric(2)
    SELECT @MinAp_Asig =TiCu_MinAp_Asig
    from TBMI_Tipo_Curs
    where TiCu_Codigo=@INTipo_Curs
```

```
      --actualizo la tabla tbmc_calif_parc
    DECLARE Estud_Asig_Cursor CURSOR FOR
    SELECT EsAs_Estud_Subu, EsAs_Asignatura, EsAs_Numer_Cred,
EsAs_Promedio
    FROM TBMI_Estud_Asig
    WHERE EsAs_Anio_Lect = @InAnio_Lect
    and EsAs_Ciclo = @InCiclo
    and EsAs_Tipo_Curs=@InTipo_Curs
    and EsAs_Anulada='N'
    and EsAs_Aprobado='N'

    OPEN Estud_Asig_Cursor

    FETCH NEXT FROM Estud_Asig_Cursor INTO @Estud_Subu,
@Asignatura, @Numer_Cred, @Promedio
    WHILE @@FETCH_STATUS = 0
    BEGIN
       Update TBMC_Calif_Parc
       set CaPa_Estad_Regi = 'B'
       where CaPa_Estud_Subu=@Estud_Subu
       and CaPa_Asignatura=@Asignatura
       and CaPa_Numer_Cred=@Numer_Cred
       and CaPa_Estad_Regi = 'D'

      if cast(@promedio as integer) >= @minap_asig
            begin
                  UPDATE TBMI_ESTUD_ASIG
                  SET EsAs_Aprobado='S'
                  WHERE EsAs_Anio_Lect = @InAnio_Lect
                  and EsAs_Ciclo = @InCiclo
                  and EsAs_Tipo_Curs=@InTipo_Curs
                  and EsAs_Anulada='N'
                  and EsAs_Aprobado='N'
            end
       FETCH NEXT FROM Estud_Asig_Cursor INTO @Estud_Subu,
@Asignatura, @Numer_Cred, @Promedio
    END

    CLOSE Estud_Asig_Cursor
    DEALLOCATE Estud_Asig_Cursor

    SELECT @error_var = @@ERROR
```

```
    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'


CREATE  PROCEDURE SPMC_CALIF_PARC_INSERTA
@INEstud_Subu               VARCHAR(10)      ,
@INAsignatura               VARCHAR(7)       ,
@INNumer_Cred               VARCHAR(1)       ,
@INTipo_Cali                VARCHAR(6)       ,
@INCalif_Trab               VARCHAR(9)       ,
@INCalif_Exam               VARCHAR(9)       ,
@INCalif_Tota               VARCHAR(9)       ,
@INPorce_Asis               VARCHAR(9)       ,
@INObservacio                     VARCHAR(120)     ,
@INUsuario                  VARCHAR(15)      ,
@OUTSql                     VARCHAR(2) OUTPUT
AS

DECLARE @error_var int

Insert into TBMC_Calif_Parc
(CaPa_Estud_Subu, CaPa_Asignatura, CaPa_Numer_Cred,
CaPa_Tipo_Cali, CaPa_Calif_Trab, CaPa_Calif_Exam, CaPa_Calif_Tota,
CaPa_Porce_Asis, CaPa_Estad_Regi, CaPa_Observacio, CaPa_Usuario,
CaPa_FecUl_Modi)
Values
(@InEstud_Subu, @InAsignatura, @InNumer_Cred, @InTipo_Cali,
@InCalif_Trab, @InCalif_Exam, @InCalif_Tota, @InPorce_Asis, 'D',
@InObservacio, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE    PROCEDURE SPMC_ENTRE_CALI_INSERTA
@INAnio_Lect            VARCHAR(4)       ,
@INCiclo          VARCHAR(2)       ,
```

```
@INTipo_Curs        VARCHAR(6)      ,
@INTipo_Cali        VARCHAR(6)      ,
@INFecMa_Entr           DATETIME    ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Insert Into TBMC_Entre_Cali
(EnCa_Anio_Lect, EnCa_Ciclo, EnCa_Tipo_Curs, EnCa_Tipo_Cali,
EnCa_FecMa_Entr, EnCa_Usuario, EnCa_FecUl_Modi)
Values
(@InAnio_Lect, @InCiclo, @InTipo_Curs, @InTipo_Cali,
@InFecMa_Entr, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE   PROCEDURE SPMC_ENTRE_CALI_MODIFICA
@INAnio_Lect        VARCHAR(4)      ,
@INCiclo        VARCHAR(2)      ,
@INTipo_Curs        VARCHAR(6)      ,
@INTipo_Cali        VARCHAR(6)      ,
@INFecMa_Entr           DATETIME    ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int
Update TBMC_Entre_Cali
Set
   EnCa_FecMa_entr    = @inFecMa_Entr,
   EnCa_Usuario       = @InUsuario,
   EnCa_FecUl_Modi    = getdate()
Where  EnCa_Anio_Lect      = @InAnio_Lect
```

```
And    EnCa_Ciclo      = @InCiclo
And    EnCa_Tipo_Curs   = @InTipo_Curs
And    EnCa_Tipo_Cali   = @InTipo_Cali

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMC_RECALIFICA_MODIFICA
@INCodig_estu          VARCHAR(10)      ,
@INCodig_asig          VARCHAR(7)       ,
@INnumer_cred          NUMERIC (1)      ,
@INtipo_cali           VARCHAR(6)       ,
@INcalif_trab          VARCHAR(9)       ,
@INcalif_exam          VARCHAR(9)       ,
@INcalif_suma          VARCHAR(9)       ,
@INUsuario             VARCHAR(15)      ,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMC_Calif_Parc
Set
   Capa_Calif_trab      = @InCalif_trab,
   Capa_Calif_exam      = @InCalif_exam,
   Capa_Calif_tota      = @InCalif_suma,
   Capa_Usuario                 = @InUsuario,
   Capa_FecUl_Modi   = getdate()
Where  Capa_estud_subu= @Incodig_estu
and    Capa_asignatura         = @Incodig_asig
and    Capa_numer_cred         = @Innumer_cred
and    Capa_tipo_cali   = @Intipo_cali
SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
```

```
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'


CREATE    PROCEDURE SPMC_SOLIC_ESPE_INSERTA
@INEstud_Subu        VARCHAR(10)       ,
@INAsignatura        VARCHAR(7)        ,
@INNumer_Cred        NUMERIC(1),
@INTipo_Soli         VARCHAR(2)        ,
@INNumer_Soli        VARCHAR(10)       ,
@INFecha_Apro        DATETIME   ,
@INObservacio        VARCHAR(120)      ,
@INUsuario           VARCHAR(15)       ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Insert Into TBMC_Solic_Espe
(SoEs_Estud_Subu, SoEs_Asignatura, SoEs_Numer_Cred,
SoEs_Tipo_Soli, SoEs_Numer_Soli, SoEs_Fecha_Apro,
SoEs_Observacio, SoEs_Usuario, SoEs_FecUl_Modi)
Values
(@InEstud_Subu, @InAsignatura, @InNumer_Cred, @InTipo_Soli,
@InNumer_Soli, @InFecha_Apro, @InObservacio,  @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'



CREATE    PROCEDURE SPMC_SOLIC_ESPE_MODIFICA
@INEstud_Subu              VARCHAR(10)       ,
@INAsignatura        VARCHAR(7)        ,
@INNumer_Cred              numeric(1,0) ,
```

```
@INTipo_Soli        VARCHAR(2)        ,
@INNumer_Soli             VARCHAR(10)       ,
@INFecha_Apro             datetime       ,
@INObservacio       VARCHAR(120)    ,
@INUsuario          VARCHAR(15)       ,
@OUTSql                   VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMC_Solic_Espe
Set
    SoEs_Tipo_Soli     = @InTipo_Soli,
    SoEs_Observacio   = @InObservacio,
      SoEs_Fecha_Apro        = @InFecha_Apro,
    SoEs_Usuario       = @InUsuario,
    SoEs_FecUl_Modi  = getdate()
Where   SoEs_Estud_Subu = @InEstud_Subu and
    SoEs_Asignatura = @InAsignatura and
    SoEs_Numer_Cred = @InNumer_Cred and
    SoEs_Numer_Soli = @InNumer_Soli

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMC_TIPO_CALI_INSERTA
@INSubunidad        VARCHAR(4)        ,
@INDescripcio       VARCHAR(60)       ,
@INReemp_Nota       VARCHAR(1)        ,
@INSuspenso         VARCHAR(1)        ,
@INUsuario          VARCHAR(15)       ,
@OUTCodigo          VARCHAR(6) OUTPUT,
@OUTSql             VARCHAR(2) OUTPUT
```

```
AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiCa_Codigo),0)+1 as
VarChar(6)))
From TBMC_Tipo_Cali
Where TiCa_Subunidad = @InSubunidad

set @OutCodigo = replicate('0', 6 - len(@OutCodigo) ) + @OutCodigo

if @OutCodigo = '000001'
  begin
    set @OutCodigo = @InSubUnidad + '01'
  end

Insert into TBMC_Tipo_Cali
(TiCa_Codigo, TiCa_Descripcio, TiCa_Reemp_Nota, TiCa_Suspenso,
TiCa_Subunidad, TiCa_Usuario, TiCa_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InReemp_Nota, @InSuspenso,
@InSubunidad, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMC_TIPO_CALI_MODIFICA
@INCodigo            VARCHAR(6)     ,
@INDescripcio        VARCHAR(60)      ,
@INSubunidad         VARCHAR(4)       ,
@InReemp_Nota        VARCHAR(1)       ,
@InSuspenso          VARCHAR(1)       ,
@INUsuario           VARCHAR(15)      ,
@OUTSql              VARCHAR(2) OUTPUT

AS
DECLARE @error_var int, @rowcount_var int
```

```
Update TBMC_Tipo_Cali
Set
   TiCa_Descripcio      = @InDescripcio,
   TiCa_Reemp_Nota  = @InReemp_Nota,
   TiCa_Suspenso      = @InSuspenso,
   TiCa_Usuario         = @InUsuario,
   TiCa_FecUl_Modi    = getdate()
Where TiCa_SubUnidad       = @InSubunidad
and   TiCa_Codigo      = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMC_TIPO_SOLI_INSERTA
@INDescripcio          VARCHAR(60)       ,
@INUsuario             VARCHAR(15)       ,
@OUTCodigo             VARCHAR(2) OUTPUT,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiSo_Codigo),0)+1 as
VarChar(2)))
From TBMC_Tipo_Soli

set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMC_Tipo_soli
(TiSo_Codigo, TiSo_descripcio, TiSo_Usuario, TiSo_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio,@InUsuario, getdate())

SELECT @error_var = @@ERROR
```

```
If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMC_TIPO_SOLI_MODIFICA
@INCodigo           VARCHAR(2)      ,
@INDescripcio       VARCHAR(60)     ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMC_Tipo_Soli
Set
   TiSo_Descripcio     = @InDescripcio,
   TiSo_Usuario = @InUsuario,
   TiSo_FecUl_Modi     = getdate()
Where  TiSo_Codigo    = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE   PROCEDURE SPMD_DOCENTE_INSERTA
@INCedula           VARCHAR(10)     ,
@INNombre           VARCHAR(30)     ,
@INApellido         VARCHAR(30)     ,
@INProvi_Resi       VARCHAR(3)      ,
@INCiuda_Resi       VARCHAR(3)      ,
@INDomicilio        VARCHAR(120)    ,
@INTelefono1        VARCHAR(10)     ,
```

```
@INTelefono2        VARCHAR(10)      ,
@INEmpresa          VARCHAR(60)      ,
@INDirec_Trab       VARCHAR(120)     ,
@INTelef_Trab       VARCHAR(10)      ,
@INNacionalid       VARCHAR(3)       ,
@INProvi_Naci       VARCHAR(3)       ,
@INCiuda_Naci       VARCHAR(3)       ,
@INFecha_Naci             DATETIME   ,
@INCarne_Afil       VARCHAR(10)      ,
@INUniversida       VARCHAR(3)       ,
@INTitul_Univ       VARCHAR(3)       ,
@INLogro_Obte       VARCHAR(120)     ,
@INCedul_Mili       VARCHAR(10)      ,
@INRecor_Poli       VARCHAR(10)      ,
@INUsuario          VARCHAR(15)      ,
@OUTCodigo          VARCHAR(5) OUTPUT,
@OUTSql             VARCHAR(2) OUTPUT
AS

Declare @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Doce_Codigo),0)+1 as
VarChar(5)))
From TBMD_Docente

set @OutCodigo = replicate('0', 5 - len(@OutCodigo) ) + @OutCodigo

Insert Into TBMD_Docente
(Doce_Codigo, Doce_Cedula, Doce_Nombre, Doce_Apellido,
Doce_Provi_Resi, Doce_Ciuda_Resi, Doce_Domicilio, Doce_Telefono1,
Doce_Telefono2, Doce_Empresa, Doce_Direc_Trab, Doce_Telef_Trab,
Doce_Nacionalid, Doce_Provi_Naci, Doce_Ciuda_Naci,
Doce_Fecha_Naci, Doce_Carne_Afil, Doce_Universida, Doce_Titul_Univ,
Doce_Logro_Obte, Doce_Cedul_Mili, Doce_Recor_Poli, Doce_Usuario,
Doce_FecUl_Modi)
Values
(@OutCodigo, @INCedula, @InNombre, @InApellido, @InProvi_Resi,
@InCiuda_Resi, @InDomicilio, @InTelefono1, @InTelefono2,
@InEmpresa, @InDirec_Trab, @InTelef_Trab, @InNacionalid,
@InProvi_Naci, @InCiuda_Naci, @InFecha_Naci, @InCarne_Afil,
@InUniversida, @InTitul_Univ, @InLogro_Obte, @InCedul_Mili,
@InRecor_Poli, @InUsuario, getdate())
```

```
SELECT @error_var = @@ERROR
if (@error_var <> 0)
    set @OutSql='NO'
else
    set @OutSql='OK'


CREATE   PROCEDURE SPMD_DOCENTE_MODIFICA
@INCodigo           VARCHAR(5)      ,
@INCedula           VARCHAR(10)     ,
@INNombre           VARCHAR(30)     ,
@INApellido         VARCHAR(30)     ,
@INProvi_Resi       VARCHAR(3)      ,
@INCiuda_Resi       VARCHAR(3)      ,
@INDomicilio        VARCHAR(120)    ,
@INTelefono1        VARCHAR(10)     ,
@INTelefono2        VARCHAR(10)     ,
@INEmpresa          VARCHAR(60)     ,
@INDirec_Trab       VARCHAR(120)    ,
@INTelef_Trab       VARCHAR(10)     ,
@INNacionalid       VARCHAR(3)      ,
@INProvi_Naci       VARCHAR(3)      ,
@INCiuda_Naci       VARCHAR(3)      ,
@INFecha_Naci           DATETIME    ,
@INCarne_Afil       VARCHAR(10)     ,
@INUniversida       VARCHAR(3)      ,
@INTitul_Univ       VARCHAR(3)      ,
@INLogro_Obte       VARCHAR(120)    ,
@INCedul_Mili       VARCHAR(10)     ,
@INRecor_Poli       VARCHAR(10)     ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

Declare @error_var int, @rowcount_var int

Update TBMD_Docente
Set
  Doce_Cedula= @InCedula,
  Doce_Apellido= @InApellido,
  Doce_Nombre=@InNombre,
  Doce_Nacionalid=@InNacionalid,
```

```
   Doce_Provi_Resi=@InProvi_Resi,
   Doce_Ciuda_Resi=@InCiuda_Resi,
   Doce_Domicilio=@InDomicilio,
   Doce_Telefono1=@InTelefono1,
   Doce_Telefono2=@InTelefono2,
   Doce_Empresa=@InEmpresa,
   Doce_Direc_Trab=@InDirec_Trab,
   Doce_Telef_Trab=@InTelef_Trab,
   Doce_Provi_Naci=@InProvi_Naci,
   Doce_Ciuda_Naci=@InCiuda_Naci,
   Doce_Fecha_Naci=@InFecha_Naci,
   Doce_Carne_Afil=@InCarne_Afil,
   Doce_Universida=@InUniversida,
   Doce_Titul_Univ=@InTitul_Univ,
   Doce_Logro_Obte=@InLogro_Obte,
   Doce_Cedul_Mili=@InCedul_Mili,
   Doce_Recor_poli=@InRecor_Poli,
   Doce_Usuario=@InUsuario,
   Doce_FecUl_Modi=getdate()
Where Doce_Codigo=@InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMD_TITUL_UNIV_INSERTA
@INDescripcio          VARCHAR(60)      ,
@INAbreviatur          VARCHAR(15)      ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int
```

```
Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiUn_Codigo),0)+1 as
VarChar(3)))
From TBMD_Titul_Univ
set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMD_Titul_Univ
(TiUn_Codigo, TiUn_Descripcio, TiUn_Abreviatur, TiUn_Usuario,
TiUn_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InAbreviatur, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMD_TITUL_UNIV_MODIFICA
@INCodigo              VARCHAR(3)       ,
@INDescripcio          VARCHAR(60)      ,
@INAbreviatur          VARCHAR(15)      ,
@INUsuario             VARCHAR(15)      ,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMD_Titul_Univ
Set
   TiUn_Descripcio     = @InDescripcio,
   TiUn_Abreviatur     = @InAbreviatur,
   TiUn_Usuario= @InUsuario,
   TiUn_FecUl_Modi     = getdate()
Where  TiUn_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
```

```
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'
CREATE  PROCEDURE SPME_ESPECIALIZ_INSERTA
@INDescripcio          VARCHAR(30)       ,
@INUsuario             VARCHAR(15)       ,
@OUTCodigo             VARCHAR(2) OUTPUT,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Espe_Codigo),0)+1 as
VarChar(2)))
From TBME_Especializ

set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo

Insert into TBME_Especializ
(Espe_Codigo, Espe_Descripcio,  Espe_Usuario, Espe_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio,  @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPME_ESPECIALIZ_MODIFICA
@INCodigo              VARCHAR(2)        ,
@INDescripcio          VARCHAR(30)       ,
@INUsuario             VARCHAR(15)       ,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int
```

```
Update TBME_Especializ
Set
   Espe_Descripcio     = @InDescripcio,
   Espe_Usuario        = @InUsuario,
   Espe_FecUl_Modi     = getdate()
Where  Espe_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE PROCEDURE SPME_ESTUDIANTE_CAMBIA_CATEGORIA
@INEstudiante                VARCHAR(6)       ,
@INCategoria                 VARCHAR(3)       ,
@INObservacio                VARCHAR(60) ,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @ancho_observacio int

Select @ancho_observacio = len(Estu_Observacio)
From TBME_Estudiante
Where Estu_Codigo = @INEstudiante

SELECT @error_var = @@ERROR

If (@error_var = 0)

  Begin
    if @ancho_observacio <= 60
          Update TBME_Estudiante
          Set
          Estu_Categoria = @INCategoria,
          Estu_Observacio = Estu_Observacio + @InObservacio
          Where Estu_Codigo = @INEstudiante
```

```
    else
            Update TBME_Estudiante
            Set
            Estu_Categoria = @INCategoria
            Where Estu_Codigo = @INEstudiante;

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

    If (@error_var <> 0)
      Set @OutSql = 'NO'
    else
       if (@rowcount_var = 1)
              Set @OutSql = 'OK'
             else
              Set @OutSql = 'NO'
  End
else
  Set @OutSql = 'NO'


CREATE  PROCEDURE SPME_ESTUDIANTE_INSERTA
@INColegio          VARCHAR(4)      ,
@INCedula           VARCHAR(10)     ,
@INApellido         VARCHAR(30)     ,
@INNombre           VARCHAR(30)     ,
@INNacionalid       VARCHAR(3)      ,
@INSexo             VARCHAR(1)      ,
@INProvi_Naci       VARCHAR(3)      ,
@INCiuda_Naci       VARCHAR(3)      ,
@INFecha_Naci           DATETIME    ,
@INEstad_Civi       VARCHAR(1)      ,
@INProvi_Domi       VARCHAR(3)      ,
@INCiuda_Domi           VARCHAR(3)      ,
@INDirec_Domi       VARCHAR(120)    ,
@INTelef_Domi       VARCHAR(10)     ,
@INTelef_Trab       VARCHAR(10)     ,
@INNombr_Padr           VARCHAR(60)     ,
@INNombr_Madr           VARCHAR(60)     ,
@INEspecializ       VARCHAR(2)      ,
@INAnio_Grad        VARCHAR(4)      ,
@INNota_Grad        VARCHAR(2)      ,
@INUniversida       VARCHAR(3)      ,
```

```
@INNombr_Empr           VARCHAR(60)      ,
@INDirec_Empr       VARCHAR(120)     ,
@INCarnet_IESS          VARCHAR(10)      ,
@INEmail            VARCHAR(40)      ,
@INGrupo_Sang           VARCHAR(2)       ,
@INCedul_Mili       VARCHAR(10)      ,
@INRecor_Poli       VARCHAR(10)      ,
@INObservacio       VARCHAR(120)     ,
@INUsuario          VARCHAR(15)      ,
@OUTCodigo          VARCHAR(6) OUTPUT,
@OUTSql                 VARCHAR(2) OUTPUT
AS

Declare @Categoria varchar(3), @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Estu_Codigo),0)+1 as
VarChar(6)))
From TBME_Estudiante

set @OutCodigo = replicate('0', 6 - len(@OutCodigo) ) + @OutCodigo

select @Categoria = Cole_Categoria
from TBMM_Colegio
where Cole_Codigo = @INColegio

Insert into TBME_Estudiante
(Estu_Codigo, Estu_Categoria, Estu_Colegio, Estu_Cedula,
Estu_Apellido, Estu_Nombre, Estu_Nacionalid, Estu_Sexo,
Estu_Provi_Naci, Estu_Ciuda_Naci, Estu_Fecha_Naci, Estu_Estad_Civi,
Estu_Provi_Domi, Estu_Ciuda_Domi, Estu_Direc_Domi,
Estu_Telef_Domi, Estu_Telef_Trab, Estu_Nombr_Padr,
Estu_Nombr_Madr, Estu_Especializ, Estu_Anio_Grad, Estu_Nota_Grad,
Estu_Universida, Estu_Nombr_Empr, Estu_Direc_Empr,
Estu_Carnet_IESS, Estu_Email, Estu_Grupo_Sang, Estu_Cedul_Mili,
Estu_Recor_Poli, Estu_Exame_Grac, Estu_Observacio, Estu_Usuario,
Estu_FecUl_Modi)
Values
(@OutCodigo, @Categoria, @InColegio, @InCedula, @InApellido,
@InNombre, @InNacionalid, @InSexo, @InProvi_Naci, @InCiuda_Naci,
@InFecha_Naci, @InEstad_Civi, @InProvi_Domi, @InCiuda_Domi,
@InDirec_Domi, @InTelef_Domi, @InTelef_Trab, @InNombr_Padr,
@InNombr_Madr, @InEspecializ, @InAnio_Grad, @InNota_Grad,
@InUniversida,  @InNombr_Empr, @InDirec_Empr, @InCarnet_IESS,
```

```
@InEmail, @InGrupo_Sang, @InCedul_Mili, @InRecor_Poli, 0,
@InObservacio, @InUsuario, getdate())

SELECT @error_var = @@ERROR

if (@error_var <> 0)
    set @OutSql='NO'
else
    set @OutSql='OK'


CREATE  PROCEDURE SPME_ESTUDIANTE_MODIFICA
@INCodigo           VARCHAR(6)      ,
@INColegio          VARCHAR(4)      ,
@INCedula           VARCHAR(10)     ,
@INApellido         VARCHAR(30)     ,
@INNombre           VARCHAR(30)     ,
@INNacionalid       VARCHAR(3)      ,
@INSexo             VARCHAR(1)      ,
@INProvi_Naci       VARCHAR(3)      ,
@INCiuda_Naci       VARCHAR(3)      ,
@INFecha_Naci           DATETIME    ,
@INEstad_Civi       VARCHAR(1)      ,
@INProvi_Domi       VARCHAR(3)      ,
@INCiuda_Domi           VARCHAR(3)      ,
@INDirec_Domi       VARCHAR(120)    ,
@INTelef_Domi       VARCHAR(10)     ,
@INTelef_Trab       VARCHAR(10)     ,
@INNombr_Padr           VARCHAR(60)     ,
@INNombr_Madr           VARCHAR(60)     ,
@INEspecializ       VARCHAR(2)      ,
@INAnio_Grad        VARCHAR(4)      ,
@INNota_Grad        VARCHAR(2)      ,
@INUniversida       VARCHAR(3)      ,
@INNombr_Empr           VARCHAR(60)     ,
@INDirec_Empr       VARCHAR(120)    ,
@INCarnet_IESS          VARCHAR(10)     ,
@INEmail            VARCHAR(40)     ,
@INGrupo_Sang           VARCHAR(2)      ,
@INCedul_Mili       VARCHAR(10)     ,
@INRecor_Poli       VARCHAR(10)     ,
@INObservacio       VARCHAR(120)    ,
@INUsuario          VARCHAR(15)     ,
```

```
@OUTSql                    VARCHAR(2) OUTPUT

AS

Declare @Categoria varchar(3), @error_var int, @rowcount_var int

--para evitar cambios innecesarios de la categoría del estudiante
if  (select estu_colegio from TBME_Estudiante where
estu_codigo=@INCodigo) <> @INColegio
  begin
     select @Categoria=Cole_Categoria
     from TBMM_Colegio
     where Cole_Codigo=@INColegio

     Update TBME_Estudiante
     Set
       Estu_Categoria=@Categoria,
       Estu_Colegio= @InColegio,
       Estu_Cedula= @InCedula,
       Estu_Apellido= @InApellido,
       Estu_Nombre=@InNombre,
       Estu_Nacionalid=@InNacionalid,
       Estu_Sexo=@InSexo,
       Estu_Provi_Naci=@InProvi_Naci,
       Estu_Ciuda_Naci=@InCiuda_Naci,
       Estu_Fecha_Naci=@InFecha_Naci,
       Estu_Estad_Civi=@InEstad_Civi,
       Estu_Provi_Domi=@InProvi_Domi,
       Estu_Ciuda_Domi=@InCiuda_Domi,
       Estu_Direc_Domi=@InDirec_Domi,
       Estu_Telef_Domi=@InTelef_Domi,
       Estu_Telef_Trab=@InTelef_Trab,
       Estu_Nombr_Padr=@InNombr_Padr,
       Estu_Nombr_Madr=@InNombr_Madr,
       Estu_Especializ=@InEspecializ,
       Estu_Anio_Grad=@InAnio_Grad,
       Estu_Nota_Grad=@InNota_Grad,
       Estu_Universida=@InUniversida,
       Estu_Nombr_Empr=@InNombr_Empr,
       Estu_Direc_Empr=@InDirec_Empr,
       Estu_Carnet_IESS=@InCarnet_IESS,
       Estu_Email=@InEmail,
       Estu_Grupo_Sang=@InGrupo_Sang,
```

```
        Estu_Cedul_Mili=@InCedul_Mili,
        Estu_Recor_Poli=@InRecor_Poli,
        Estu_Observacio=@InObservacio,
        Estu_Usuario=@InUsuario,
        Estu_FecUl_Modi=getdate()
      Where Estu_Codigo=@InCodigo
    end
  else
    begin
      Update TBME_Estudiante
      Set
        Estu_Cedula= @InCedula,
        Estu_Apellido= @InApellido,
        Estu_Nombre=@InNombre,
        Estu_Nacionalid=@InNacionalid,
        Estu_Sexo=@InSexo,
        Estu_Provi_Naci=@InProvi_Naci,
        Estu_Ciuda_Naci=@InCiuda_Naci,
        Estu_Fecha_Naci=@InFecha_Naci,
        Estu_Estad_Civi=@InEstad_Civi,
        Estu_Provi_Domi=@InProvi_Domi,
        Estu_Ciuda_Domi=@InCiuda_Domi,
        Estu_Direc_Domi=@InDirec_Domi,
        Estu_Telef_Domi=@InTelef_Domi,
        Estu_Telef_Trab=@InTelef_Trab,
        Estu_Nombr_Padr=@InNombr_Padr,
        Estu_Nombr_Madr=@InNombr_Madr,
        Estu_Especializ=@InEspecializ,
        Estu_Anio_Grad=@InAnio_Grad,
        Estu_Nota_Grad=@InNota_Grad,
        Estu_Universida=@InUniversida,
        Estu_Nombr_Empr=@InNombr_Empr,
        Estu_Direc_Empr=@InDirec_Empr,
        Estu_Carnet_IESS=@InCarnet_IESS,
        Estu_Email=@InEmail,
        Estu_Grupo_Sang=@InGrupo_Sang,
        Estu_Cedul_Mili=@InCedul_Mili,
        Estu_Recor_Poli=@InRecor_Poli,
        Estu_Observacio=@InObservacio,
        Estu_Usuario=@InUsuario,
        Estu_FecUl_Modi=getdate()
      Where Estu_Codigo=@InCodigo
    end
```

```
SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'
CREATE  PROCEDURE SPMG_CIUDAD_INSERTA
@INDescripcio          VARCHAR(60)      ,
@INPais                VARCHAR(3)       ,
@INProvincia           VARCHAR(3)       ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Ciud_Codigo),0)+1 as
VarChar(3)))
From TBMG_Ciudad

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMG_Ciudad
(Ciud_Codigo, Ciud_Descripcio, Ciud_Pais, Ciud_Provincia,
Ciud_Usuario, Ciud_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InPais, @InProvincia, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'
```

```
CREATE  PROCEDURE SPMG_CIUDAD_MODIFICA
@INCodigo           VARCHAR(3)      ,
@INDescripcio       VARCHAR(60)     ,
@INPais             VARCHAR(3)      ,
@INProvincia        VARCHAR(3)      ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int
Update TBMG_Ciudad
Set
   Ciud_Descripcio     = @InDescripcio,
   Ciud_Pais           = @InPais,
   Ciud_Provincia      = @InProvincia,
   Ciud_Usuario = @InUsuario,
   Ciud_FecUl_Modi     = getdate()
Where  Ciud_Codigo   = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE  PROCEDURE SPMG_PAIS_INSERTA
@INDescripcio       VARCHAR(60)     ,
@INNacionalid       VARCHAR(30)     ,
@INUsuario          VARCHAR(15)     ,
@OUTCodigo          VARCHAR(3) OUTPUT,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int
```

```
Select @OutCodigo = RTrim(Cast(Coalesce(Max(Pais_Codigo),0)+1 as
VarChar(3)))
From TBMG_Pais

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMG_Pais
(Pais_Codigo, Pais_Descripcio, Pais_Nacionalid, Pais_Usuario,
Pais_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InNacionalid, @InUsuario, getdate())

SELECT @error_var = @@ERROR
If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMG_PAIS_MODIFICA
@INCodigo              VARCHAR(3)       ,
@INDescripcio          VARCHAR(60)      ,
@INnacionalid          VARCHAR(30)      ,
@INUsuario             VARCHAR(15)      ,
@OUTSql                        VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMG_Pais
Set
   Pais_Descripcio     = @InDescripcio,
   Pais_Nacionalid     = @InNacionalid,
   Pais_Usuario = @InUsuario,
   Pais_FecUl_Modi     = getdate()
Where  pais_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
```

```
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMG_PROVINCIA_INSERTA
@INDescripcio          VARCHAR(60)      ,
@INPais                VARCHAR(3)       ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                      VARCHAR(2) OUTPUT

AS
DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Prov_Codigo),0)+1 as
VarChar(3)))
From TBMG_Provincia

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMG_Provincia
(Prov_Codigo, Prov_Descripcio, Prov_Pais, Prov_Usuario,
Prov_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InPais, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMG_PROVINCIA_MODIFICA
@INCodigo              VARCHAR(3)       ,
@INDescripcio          VARCHAR(60)      ,
@INpais                VARCHAR(3)       ,
@INUsuario             VARCHAR(15)      ,
@OUTSql                      VARCHAR(2) OUTPUT
```

```
AS

DECLARE @error_var int, @rowcount_var int

Update TBMG_Provincia
Set
   Prov_Descripcio      = @InDescripcio,
   Prov_Pais            = @InPais,
   Prov_Usuario = @InUsuario,
   Prov_FecUl_Modi    = getdate()
Where  prov_Codigo    = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMG_UNIVERSIDA_INSERTA
@INDescripcio          VARCHAR(60)      ,
@INPais          VARCHAR(3)       ,
@INProvincia           VARCHAR(3)       ,
@INCiudad              VARCHAR(3)       ,
@INDireccion           VARCHAR(120)     ,
@INTelefono            VARCHAR(10)      ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Univ_Codigo),0)+1 as
VarChar(3)))
From TBMG_Universida

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo
```

```
Insert into TBMG_Universida
(Univ_Codigo, Univ_Descripcio, Univ_Pais, Univ_Provincia, Univ_Ciudad,
Univ_Direccion, Univ_Telefono, Univ_Usuario, Univ_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InPais, @InProvincia, @InCiudad,
@InDireccion, @InTelefono, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMG_UNIVERSIDA_MODIFICA
@INCodigo              VARCHAR(3)        ,
@INDescripcio          VARCHAR(60)       ,
@INPais          VARCHAR(3)        ,
@INProvincia           VARCHAR(3)        ,
@INCiudad              VARCHAR(3)        ,
@INDireccion           VARCHAR(120)      ,
@INTelefono            VARCHAR(10)       ,
@INUsuario             VARCHAR(15)       ,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMG_Universida
Set
   Univ_Descripcio      = @InDescripcio,
   Univ_Pais      = @InPais,
   Univ_Provincia = @InProvincia,
   Univ_Ciudad = @InCiudad,
   Univ_Direccion = @InDireccion,
   Univ_Telefono = @InTelefono,
   Univ_Usuario = @InUsuario,
   Univ_FecUl_Modi     = getdate()
Where  Univ_Codigo    = @InCodigo
```

```
SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE PROCEDURE SPMI_CIERRE_NIVEL
@INAnio_Lect          VARCHAR(4)        ,
@INCiclo         VARCHAR(2)        ,
@INTipo_Curs          VARCHAR(6)        ,
@OUTSql               VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @Estud_Subu
varchar(10), @Suma_Prom numeric(3,2), @Numer_Apro integer,
@Numer_Repr integer, @Promedio varchar(9), @asignatura varchar(7),
@aprobado varchar(1), @estudiante varchar(10)
    --cursor que contiene todos los estudiante y todas las materias que
ellos vieron en este periodo lectivo
    DECLARE Estud_Asig_Cursor CURSOR FOR
    SELECT distinct EsAs_Estud_Subu, EsAs_Asignatura,
EsAs_Promedio, EsAs_Aprobado
    FROM TBMI_Estud_Asig
    WHERE EsAs_Anio_Lect = @InAnio_Lect
    and EsAs_Ciclo = @InCiclo
    and EsAs_Tipo_Curs=@InTipo_Curs
    and EsAs_Anulada='N'
    order by EsAs_Estud_Subu

    OPEN Estud_Asig_Cursor
    FETCH NEXT FROM Estud_Asig_Cursor INTO @Estud_Subu,
@Asignatura, @Promedio, @Aprobado
    --encero las variables mas importantes
    set @estudiante=@estud_subu
    set @suma_prom=0
    set @numer_apro=0
    set @numer_repr=0
```

```
    WHILE @@FETCH_STATUS = 0
    BEGIN
            if @estudiante=@estud_subu
            BEGIN
                    set @suma_prom=@suma_prom + cast (@promedio as
integer)
                    if @aprobado = 'S'
                            set @numer_apro=@numer_apro+1
                    else
                            set @numer_repr=@numer_repr+1
            END
            ELSE
            BEGIN
                    set @suma_prom=@suma_prom / (@numer_apro +
@numer_repr)

                    Update TBMI_Estud_Nive
                    Set EsNi_Promedio = cast(@suma_Prom as varchar (9)),
                        EsNi_Asign_Apro = @numer_apro,
                        EsNi_Asign_Repr=@numer_repr
                    Where EsNi_Estud_Subu=@estudiante
                    And EsNi_Anio_Lect = @InAnio_Lect
                    And EsNi_Ciclo = @InCiclo
                    And EsNi_Tipo_Curs = @InTipo_Curs

                    set @estudiante = @estud_subu
                    set @numer_apro = 0
                    set @numer_repr = 0
            END
            FETCH NEXT FROM Estud_Asig_Cursor INTO @Estud_Subu,
@Asignatura, @Promedio, @Aprobado
    END
    CLOSE Estud_Asig_Cursor
    DEALLOCATE Estud_Asig_Cursor

    SELECT @error_var = @@ERROR
    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'
```

```
CREATE    PROCEDURE SPMI_CIERRE_SUBUNIDAD
@INAnio_Lect            VARCHAR(4)        ,
@INCiclo          VARCHAR(2)        ,
@INTipo_Curs            VARCHAR(6)        ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @Maxim_Nive integer,
@Estud_Subu varchar(10), @Nivel_Apro integer, @Asig_Codigo
varchar(7), @aprobado varchar(1), @Asign_Apro integer, @Numer_Asig
integer, @suma_prom numeric(3,2), @suma_asig integer, @asignatura
varchar(7), @promedio varchar(9)

    --obtengo el maximo nivel de la subunidad
    Select @Maxim_Nive=SuAc_Nivel
    From TBMA_Subun_Acad
    Where SuAc_Codigo=substring(@InTipo_Curs,1,2)

    --cursor que contendra todos los estudiantes que se inscribieron al
inicio del periodo lectivo
    DECLARE Estud_Subu_Cursor CURSOR FOR
    SELECT DISTINCT EsAs_Estud_Subu
    FROM TBMI_Estud_Asig
    WHERE EsAs_Anio_Lect = @InAnio_Lect
    and EsAs_Ciclo = @InCiclo
    and EsAs_Tipo_Curs=@InTipo_Curs

    OPEN Estud_Subu_Cursor

    FETCH NEXT FROM Estud_Subu_Cursor INTO @Estud_Subu
    WHILE @@FETCH_STATUS = 0
    BEGIN
        --se pide el ultimo nivel aprobado para cada estudiante
        Select @Nivel_Apro=cast(EsSu_UltNi_Apro as integer)
        From TBMI_Estud_Subu
        Where EsSu_CodEs_Subu=@Estud_Subu
        --establecemos valores para las variables mas importantes

        set @Nivel_Apro=@Nivel_Apro+1
        set @asign_apro=0
        set @numer_Asig=0
        --este cursor contiene todas las asignaturas que están dentro de
```

```
--un nivel superior al aprobado por este estudiante
DECLARE Asignatura_Cursor CURSOR FOR
SELECT Asig_Codigo
FROM TBMA_Asignatura
WHERE Asig_Nivel=@Nivel_Apro

OPEN Asignatura_Cursor

FETCH NEXT FROM Asignatura_Cursor INTO @Asig_Codigo
WHILE @@FETCH_STATUS = 0
BEGIN
	--se verfica por cada materia si la aprobo o no
	Select @aprobado=EsAs_Aprobado
	From TBMI_Estud_Asig
	Where EsAs_Asignatura=@Asig_Codigo
	and EsAs_Estud_Subu=@Estud_Subu
	and EsAs_Aprobado='S'
	--se va contando cada materia aprobada
	if @aprobado='S'
	begin
		set @asign_apro=@asign_apro+1
	end

	FETCH NEXT FROM Asignatura_Cursor INTO
@Asig_Codigo
END
CLOSE Asignatura_Cursor
DEALLOCATE Asignatura_Cursor
--se consulta cuantas asignaturas deberia aprobar para pasar al
siguiente nivel
Select @Numer_Asig=count(*)
From TBMA_Asignatura
WHERE Asig_Nivel=@Nivel_Apro
--se compara si ha aprobado todas las materias
if @asign_apro = @numer_asig
begin
	--se actualiza el registro indicando que el estudiante ha
aprobado un nivel mas
	update TBMI_Estud_Subu
	Set EsSu_UltNi_Apro=cast(@Nivel_Apro as varchar(2))
	Where EsSu_CodEs_Subu=@Estud_Subu
	--se pregunta si el estudiante ya aprobo el ultimo nivel
	if @nivel_apro=@Maxim_Nive
```

```
begin
        --se actualiza ademas el estado academico a
EGRESADO
        update TBMI_Estud_Subu
        Set EsSu_Condi_Acad='4',
        EsSu_Fecha_Egre=getdate()
        Where EsSu_CodEs_Subu=@Estud_Subu
        --se calcula el promedio egresado
        set @suma_prom=0
        set @suma_asig=0
        --se trae cada asignatura que ha visto con el
promedio que aprobo
        DECLARE Estud_Asig_Cursor CURSOR FOR
        Select EsAs_Asignatura, EsAs_Promedio
        from TBMI_Estud_Asig a
        Where EsAs_Estud_Subu=@Estud_Subu
        And EsAs_Numer_Cred=(Select
max(EsAs_Numer_Cred)
                From TBMI_Estud_Asig
                where
EsAs_Estud_Subu=@Estud_Subu
                and
EsAs_Asignatura=a.EsAs_Asignatura)
        And EsAs_Anulada='N'

        OPEN Estud_Asig_Cursor

        Fetch Next From Estud_Asig_Cursor INTO
@asignatura, @promedio

        While @@FETCH_STATUS = 0
        begin
                --se suman los promedios y cuantas
asignaturas ha visto
                set @suma_prom=@suma_prom +
cast(@promedio as integer)
                set @suma_asig=@suma_asig+1
                Fetch Next From Estud_Asig_Cursor INTO
@asignatura, @promedio
        end
        CLOSE Estud_Asig_Cursor
        DEALLOCATE Estud_Asig_Cursor
        --se obtiene el promedio de los promedios
```

```
                        set @suma_prom=@suma_prom/@suma_asig
                        --se actualiza el registro
                        update TBMI_Estud_Subu
                        Set EsSu_Prome_Egre=cast(@suma_prom as
varchar(9))
                        Where EsSu_CodEs_Subu=@Estud_Subu
                    end
                end
                FETCH NEXT FROM Estud_Subu_Cursor INTO @Estud_Subu
    END
    CLOSE Estud_Subu_Cursor
    DEALLOCATE Estud_Subu_Cursor

    SELECT @error_var = @@ERROR
    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'



CREATE  PROCEDURE SPMI_DOCEN_ASIG_ELIMINA
@INDocente              VARCHAR(5)      ,
@INAsignatura           VARCHAR(7)      ,
@InAnio_Lect            VARCHAR(4)      ,
@INCiclo          VARCHAR(2)        ,
@INTipo_Curs            VARCHAR(6)      ,
@INGrupo                VARCHAR(8)      ,
@OUTSql                         VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Delete From TBMI_Docen_Asig
Where DoAs_Docente = @InDocente
and     DoAs_Asignatura = @InAsignatura
and     DoAs_Anio_Lect=@InAnio_Lect
and     DoAs_Ciclo=@InCiclo
and     DoAs_Tipo_Curs=@InTipo_Curs
and     DoAs_Grupo = @InGrupo
```

```
SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE  PROCEDURE SPMI_DOCEN_ASIG_INSERTA
@INDocente              VARCHAR(5)        ,
@INAsign_Ante           VARCHAR(7)        ,
@INAsign_Actu           VARCHAR(7)        ,
@INAnio_Lect            VARCHAR(4)        ,
@INCiclo          VARCHAR(2)        ,
@INTipo_Curs            VARCHAR(6)        ,
@INGrupo_Ante                   VARCHAR(8)        ,
@INGrupo_Actu                   VARCHAR(8)        ,
@INCupo_Estu            VARCHAR(2)        ,
@INUsuario              VARCHAR(15)       ,
@OUTSql                 VARCHAR(2) OUTPUT
AS

DECLARE @error_var int, @contador int, @docente varchar(5),
@docentes int

Select @contador = count(*)
From       TBMI_Docen_Asig
Where DoAs_Docente = @InDocente
and      DoAs_Asignatura = @InAsign_Ante
and      DoAs_Anio_Lect=@InAnio_Lect
and      DoAs_Ciclo=@InCiclo
and      DoAs_Tipo_Curs=@InTipo_Curs
and      DoAs_Grupo = @InGrupo_Ante

if @contador = 0
   begin
    Select @docentes = count(*)
    From  TBMI_Docen_Asig
    Where DoAs_Asignatura = @InAsign_Actu
    and      DoAs_Anio_Lect=@InAnio_Lect
```

```
    and     DoAs_Ciclo=@InCiclo
    and     DoAs_Tipo_Curs=@InTipo_Curs
    and     DoAs_Grupo = @InGrupo_Actu

    if @docentes=0
       begin
            insert into TBMI_Docen_Asig
            (DoAs_Docente, DoAs_Asignatura, DoAs_Anio_Lect,
DoAs_Ciclo, DoAs_Tipo_Curs, DoAs_Grupo, DoAs_Cupo_Estu,
DoAs_Usuario, DoAs_FecUl_Modi)
            values
            (@InDocente, @InAsign_Actu, @InAnio_Lect, @InCiclo,
@InTipo_Curs, @InGrupo_Actu, @InCupo_Estu, @InUsuario, getdate())

            SELECT @error_var = @@ERROR

            If (@error_var <> 0)
               Set @OutSql = 'NO'
            else
               Set @OutSql = 'OK'
       end
    else
            set @OutSql='DU'
   end
else
   begin
    Select @docentes=count(*)
    From  TBMI_Docen_Asig
    Where DoAs_Asignatura = @InAsign_Actu
    and     DoAs_Anio_Lect=@InAnio_Lect
    and     DoAs_Ciclo=@InCiclo
    and     DoAs_Tipo_Curs=@InTipo_Curs
    and     DoAs_Grupo = @InGrupo_Actu

    if @docentes=0
       begin
            Update TBMI_Docen_Asig
            Set DoAs_Asignatura=@InAsign_Actu,
                DoAs_Grupo=@InGrupo_Actu,
                DoAs_Cupo_Estu=@InCupo_Estu
            Where DoAs_Docente=@InDocente
            and     DoAs_Asignatura=@InAsign_Ante
            and     DoAs_Anio_Lect=@InAnio_Lect
```

```
            and     DoAs_Ciclo=@InCiclo
            and     DoAs_Tipo_Curs=@InTipo_Curs
            and     DoAs_Grupo=@InGrupo_Ante

            SELECT @error_var = @@ERROR

            If (@error_var <> 0)
               Set @OutSql = 'NO'
            else
               Set @OutSql = 'OK'
      end
   else
      begin
            Select @docente=doas_docente
            From  TBMI_Docen_Asig
            Where DoAs_Asignatura = @InAsign_Actu
            and     DoAs_Anio_Lect=@InAnio_Lect
            and     DoAs_Ciclo=@InCiclo
            and     DoAs_Tipo_Curs=@InTipo_Curs
            and     DoAs_Grupo = @InGrupo_Actu

            if @docente=@indocente
               begin
                  Update TBMI_Docen_Asig
                  Set DoAs_Asignatura=@InAsign_Actu,
                        DoAs_Grupo=@InGrupo_Actu,
                        DoAs_Cupo_Estu=@InCupo_Estu
                  Where DoAs_Docente=@InDocente
                  and     DoAs_Asignatura=@InAsign_Ante
                  and     DoAs_Anio_Lect=@InAnio_Lect
                  and     DoAs_Ciclo=@InCiclo
                  and     DoAs_Tipo_Curs=@InTipo_Curs
                  and     DoAs_Grupo=@InGrupo_Ante

                  SELECT @error_var = @@ERROR

                  If (@error_var <> 0)
                     Set @OutSql = 'NO'
                  else
                     Set @OutSql = 'OK'
               end
            else
                  Set @OutSql = 'DU'
```

```
    end
  end


CREATE PROCEDURE SPMI_ESTUD_ASIG_ELIMINA
@INEstud_Subu       VARCHAR(10)      ,
@INAsignatura       VARCHAR(7)       ,
@INNumer_Cred       NUMERIC(1),
@INGrupo_Ante             VARCHAR(8)       ,
@INSeccion_Ante           VARCHAR(1)       ,
@OUTSql             VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

DECLARE @HoGr_Codigo varchar(2)


   --Elimino registros de la tabla tbmi_horar_estu
  DECLARE Horar_Grup_Cursor CURSOR FOR
  SELECT HoGr_Codigo
  FROM TBMI_Horar_Grup
  WHERE HoGr_Grupo          =      @InGrupo_Ante
  AND   HoGr_Seccion        =      @InSeccion_Ante
  AND   HoGr_Asignatura   =      @InAsignatura

  OPEN Horar_Grup_Cursor

  FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
  WHILE @@FETCH_STATUS = 0
  BEGIN
     Delete From TBMI_Horar_Estu
     Where HoEs_Estud_Subu     =      @InEstud_Subu
     And HoEs_Grupo            =      @InGrupo_Ante
     And HoEs_Horar_Grup =      @HoGr_Codigo

     FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
  END

  CLOSE Horar_Grup_Cursor
  DEALLOCATE Horar_Grup_Cursor
```

```
Delete From TBMI_Estud_Asig
Where    EsAs_Estud_Subu  =      @InEstud_Subu
and      EsAs_Asignatura  =      @InAsignatura
and      EsAs_Numer_Cred  =      @InNumer_Cred

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE PROCEDURE SPMI_ESTUD_ASIG_MODIFICA
@INEstud_Subu        VARCHAR(10)      ,
@INAsignatura        VARCHAR(7)       ,
@INNumer_Cred        NUMERIC(1),
@INAnio_Lect         VARCHAR(4)       ,
@INCiclo             VARCHAR(2)       ,
@INTipo_Curs         VARCHAR(6)       ,
@INGrupo             VARCHAR(8)       ,
@INGrupo_Ante              VARCHAR(8)       ,
@INSubgrupo          VARCHAR(1)       ,
@INSeccion           VARCHAR(1)       ,
@INSeccion_Ante            VARCHAR(1)       ,
@INUsuario           VARCHAR(15)      ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @contador int
DECLARE @HoGr_Codigo varchar(2)

--Reviso si la asignatura existe (ya registrada en tbmi_estud_asig)
Select @contador = count(*)
From     TBMI_Estud_Asig
Where    EsAs_Estud_Subu  =      @InEstud_Subu
And      EsAs_Asignatura  =      @InAsignatura
And      EsAs_Numer_Cred        =       @InNumer_Cred
```

```
if @contador = 0
  --Significa que debo insertar, es un nueva materia
  Begin
     --Inserto en la tabla tbmi_estud_asig
    Insert into TBMI_Estud_Asig
    (EsAs_Estud_Subu, EsAs_Asignatura, EsAs_Numer_Cred,
EsAs_Anio_Lect, EsAs_Ciclo, EsAs_Tipo_curs, EsAs_Grupo,
EsAs_Subgrupo, EsAs_Anulada, EsAs_Promedio, EsAs_Aprobado,
EsAs_Observacio, EsAs_Usuario, EsAs_FecUl_Modi)
    Values
    (@InEstud_Subu, @InAsignatura, @InNumer_Cred, @InAnio_Lect,
@InCiclo, @InTipo_Curs, @InGrupo, @InSubgrupo, 'N', '', 'N', '',
@InUsuario, getdate())

     --Inserto en la tabla tbmi_horar_estu
    DECLARE Horar_Grup_Cursor CURSOR FOR
    SELECT HoGr_Codigo
    FROM TBMI_Horar_Grup
    WHERE HoGr_Grupo          =     @InGrupo
    AND   HoGr_Seccion        =     @InSeccion
    AND   HoGr_Asignatura     =     @InAsignatura

    OPEN Horar_Grup_Cursor

    FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    WHILE @@FETCH_STATUS = 0
    BEGIN
       Insert into TBMI_Horar_Estu
       (HoEs_Horar_Grup, HoEs_Grupo, HoEs_Estud_Subu,
HoEs_Usuario, HoEs_FecUl_Modi)
       values
       (@HoGr_Codigo, @InGrupo, @INEstud_Subu, @InUsuario,
getdate())

       FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    END

    CLOSE Horar_Grup_Cursor
    DEALLOCATE Horar_Grup_Cursor

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
```

```
        Set @OutSql = 'NO'
    else
        Set @OutSql = 'OK'

    End

else
    --Significa que la asignatura ya está asignada al estudiante, actualizo
datos

     if (@InGrupo_Ante <> '' )
       Begin
       --Actualizo la tabla tbmi_estud_asig
    Update TBMI_Estud_Asig
    Set     EsAs_Grupo          =       @INGrupo,
            EsAs_Subgrupo               =       @INSubgrupo,
            EsAs_Usuario                =       @INUsuario,
            EsAs_FecUl_Modi     =       getdate()
    WhereEsAs_Estud_Subu        =       @InEstud_Subu
    And     EsAs_Asignatura     =       @InAsignatura
    And     EsAs_Numer_Cred     =       @InNumer_Cred

       --Elimino registros de la tabla tbmi_horar_estu
    DECLARE Horar_Grup_Cursor CURSOR FOR
    SELECT HoGr_Codigo
    FROM TBMI_Horar_Grup
    WHERE HoGr_Grupo            =       @InGrupo_Ante
    AND   HoGr_Seccion          =       @InSeccion_Ante
    AND   HoGr_Asignatura       =       @InAsignatura

    OPEN Horar_Grup_Cursor

    FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    WHILE @@FETCH_STATUS = 0
    BEGIN
       Delete From TBMI_Horar_Estu
       Where HoEs_Estud_Subu    =       @InEstud_Subu
       And HoEs_Grupo           =       @InGrupo_Ante
       And HoEs_Horar_Grup      =       @HoGr_Codigo

       FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    END
```

```
    CLOSE Horar_Grup_Cursor
    DEALLOCATE Horar_Grup_Cursor

      --Inserto en la tabla tbmi_horar_estu
    DECLARE Horar_Grup_Cursor CURSOR FOR
    SELECT HoGr_Codigo
    FROM TBMI_Horar_Grup
    WHERE HoGr_Grupo          =     @InGrupo
    AND   HoGr_Seccion        =     @InSeccion
    AND   HoGr_Asignatura     =     @InAsignatura

    OPEN Horar_Grup_Cursor

    FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    WHILE @@FETCH_STATUS = 0
    BEGIN
       Insert into TBMI_Horar_Estu
       (HoEs_Horar_Grup, HoEs_Grupo, HoEs_Estud_Subu,
HoEs_Usuario, HoEs_FecUl_Modi)
       values
       (@HoGr_Codigo, @InGrupo, @INEstud_Subu, @InUsuario,
getdate())

       FETCH NEXT FROM Horar_Grup_Cursor INTO @HoGr_Codigo
    END

    CLOSE Horar_Grup_Cursor
    DEALLOCATE Horar_Grup_Cursor

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'
       End
   else
       Set @OutSql = 'OK'


CREATE PROCEDURE SPMI_ESTUD_NIVE_MODIFICA
@INEstud_Subu        VARCHAR(10)      ,
@INAnio_Lect         VARCHAR(4)       ,
```

```
@INCiclo            VARCHAR(2)      ,
@INTipo_Curs        VARCHAR(6)      ,
@INNivel        VARCHAR(2)      ,
@INAsign_Toma       NUMERIC     ,
@INNumer_Vez        VARCHAR(1),
@INUsuario          VARCHAR(15)     ,
@OUTNumer_Matr      VARCHAR(10) OUTPUT,
@OUTSql             VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @contador int, @rowcount_var int

If (@InAsign_Toma) = 0
-- Elimino la matrícula

  Begin
    Select      @OutNumer_Matr = EsNi_Numer_Matr
    From TBMI_Estud_Nive
    Where       EsNi_Estud_Subu  =      @InEstud_Subu
    and         EsNi_Anio_Lect          =       @InAnio_Lect
    and         EsNi_Ciclo          =       @InCiclo
    and         EsNi_Tipo_Curs          =       @InTipo_Curs

    Delete From TBMI_Estud_Nive
    Where       EsNi_Estud_Subu  =      @InEstud_Subu
    and         EsNi_Numer_Matr  =      @OutNumer_Matr

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

    If (@error_var <> 0)
      Set @OutSql = 'NO'
    else
      if (@rowcount_var = 1)
        Set @OutSql = 'OK'
      else
        Set @OutSql = 'NO'
  End

else
-- Reviso si existe para actualizar o Inserto un registro nuevo en
tbmi_estud_nive
```

```
Begin

   Select @contador = count(*)
   From TBMI_Estud_Nive
   Where      EsNi_Estud_Subu  =      @InEstud_Subu
   and        EsNi_Anio_Lect          =      @InAnio_Lect
   and        EsNi_Ciclo       =      @InCiclo
   and        EsNi_Tipo_Curs          =      @InTipo_Curs

   SELECT @error_var = @@ERROR

   If (@contador = 0)
     Begin
      --Inserto
      Select @OutNumer_Matr =
rtrim(cast(right(coalesce(Max(EsNi_Numer_Matr),0),4)+1 as varchar(4)))
      From TBMI_Estud_Nive
      Where    EsNi_Estud_Subu >= substring(@InEstud_SubU,1,4) +
'000000'
      and      EsNi_Estud_Subu <= substring(@InEstud_SubU,1,4) +
'999999'
      and      EsNi_Anio_Lect          =      @InAnio_Lect
      and      EsNi_Ciclo       =      @InCiclo
      and      EsNi_Tipo_Curs          =      @InTipo_Curs

      set @OutNumer_Matr = ltrim(str(Year(getdate()))) + replicate('0', 2
- len(Month(getdate()))) + ltrim(str(Month(getdate()))) + replicate('0', 4 -
len(@OutNumer_Matr) ) + @OutNumer_Matr

       Insert into TBMI_Estud_Nive
       (EsNi_Estud_Subu, EsNi_Numer_Matr, EsNi_Tipo_Curs,
EsNi_Anio_Lect, EsNi_Ciclo, EsNi_Nivel, EsNi_Numer_Vez,
EsNi_Asign_Toma, EsNi_Promedio, EsNi_Numer_Susp,
EsNi_Asign_Apro, EsNi_Asign_Repr, EsNi_Premiado, EsNi_Sancionado,
EsNi_Observacio, EsNi_Usuario, EsNi_FecUl_Modi)
       Values
       (@InEstud_Subu, @OutNumer_Matr, @InTipo_Curs,
@InAnio_Lect, @InCiclo, @InNivel, @InNumer_Vez, @InAsign_Toma, '',
0, 0, 0,'N','N','', @InUsuario, getdate())

      set @error_var = 0

      SELECT @error_var = @@ERROR
```

```
        If (@error_var <> 0)
          Set @OutSql = 'NO'
        else
          Set @OutSql = 'OK'
      End

    else
      Begin
        --Actualizo
        Select      @OutNumer_Matr = EsNi_Numer_Matr
        From TBMI_Estud_Nive
        Where    EsNi_Estud_Subu  =      @InEstud_Subu
        and              EsNi_Anio_Lect          =      @InAnio_Lect
        and              EsNi_Ciclo          =      @InCiclo
        and              EsNi_Tipo_Curs          =      @InTipo_Curs

        Update TBMI_Estud_Nive
        Set
          EsNi_Asign_Toma          =      @InAsign_Toma,
    EsNi_Usuario                   =      @InUsuario,
    EsNi_FecUl_Modi          =      getdate()
        Where    EsNi_Estud_Subu  =      @InEstud_Subu
        and      EsNi_Numer_Matr  =      @OutNumer_Matr

        SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT
        If (@error_var <> 0)
          Set @OutSql = 'NO'
        else
          if (@rowcount_var = 1)
            Set @OutSql = 'OK'
          else
            Set @OutSql = 'NO'
      End
  End


CREATE PROCEDURE SPMI_ESTUD_SUBU_CAMBIA_PENSUM
@INEstudiante              VARCHAR(10)     ,
@INPensu_Actu              VARCHAR(6)      ,
@INPensu_Ante              VARCHAR(6)      ,
@INMotCa_Pens              VARCHAR(2) ,
```

```
@INObservacio           VARCHAR(60) ,
@INUsuario              VARCHAR(15) ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @ancho_observacio int

Select @ancho_observacio = len(EsSu_Observacio)
From TBMI_Estud_Subu
Where EsSu_CodEs_Subu = @INEstudiante

SELECT @error_var = @@ERROR

if (@error_var = 0)
Begin
    Insert into TBMI_Estud_Pens
    (EsPe_Estud_Subu, EsPe_Pensu_Actu, EsPe_Pensu_Ante,
EsPe_MotCa_Pens, EsPe_Usuario, EsPe_FecUl_Modi)
    Values
    (@InEstudiante, @InPensu_Actu, @InPensu_Ante, @InMotCa_Pens,
@InUsuario, getdate())

    SELECT @error_var = @@ERROR
End

If (@error_var = 0)
  Begin
    if @ancho_observacio <= 60
            Update TBMI_Estud_Subu
            Set
            EsSu_Pensu_Actu = @INPensu_Actu,
            EsSu_Observacio = EsSu_Observacio + @InObservacio
            Where EsSu_CodEs_Subu = @INEstudiante
    else
            Update TBMI_Estud_Subu
            Set
            EsSu_Pensu_Actu = @INPensu_Actu
            Where EsSu_CodEs_Subu = @INEstudiante;

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT
```

```
    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       if (@rowcount_var = 1)
          Set @OutSql = 'OK'
       else
          Set @OutSql = 'NO'

  End

else
   Set @OutSql = 'NO'


CREATE PROCEDURE SPMI_ESTUD_SUBU_ELIMINA
@INCodEs_Subu              VARCHAR(10)      ,
@OUTSql                    VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Delete
From TBMI_Estud_SubU
where EsSu_CodEs_Subu    =       @InCodEs_Subu

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'
CREATE PROCEDURE SPMI_ESTUD_SUBU_INSERTA
@INSubUnidad               VARCHAR(4)       ,
@INEstudiante              VARCHAR(6)       ,
@INUltNi_Apro              VARCHAR(2)       ,
@INPensu_Actu              VARCHAR(6)       ,
@INFecha_Ingr              DATETIME   ,
@INFecha_Egre              DATETIME   ,
@INCondi_Acad              VARCHAR(1)       ,
```

```
@INProme_Egre           VARCHAR(9)        ,
@INProme_Grad           VARCHAR(9)        ,
@INProme_Fina                 VARCHAR(9)        ,
@INProye_Grad                 VARCHAR(4)        ,
@INNumPr_Grad           TINYINT       ,
@INObservacio           VARCHAR(120) ,
@INCruce_Perm           TINYINT       ,
@INUsuario              VARCHAR(15)       ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @codEs_Subu varchar(10), @error_var int, @count int

Select @count = count(*)
From TBMI_Estud_SubU
where EsSu_Subunidad      =       @InSubUnidad
and EsSu_Estudiante  =       @InEstudiante

IF @count <> 0
   Set @OutSql = 'DU'
else
  Begin
    SET @codEs_Subu=@inSubunidad + @InEstudiante
    Insert into TBMI_Estud_SubU
    (EsSu_CodEs_Subu, EsSu_Subunidad, EsSu_Estudiante,
EsSu_UltNi_Apro, EsSu_Usuario, EsSu_Pensu_Actu, EsSu_Fecha_Ingr,
EsSu_Fecha_Egre, EsSu_Condi_Acad, EsSu_Prome_Egre,
EsSu_Prome_Grad, EsSu_Prome_Fina, EsSu_Proye_Grad,
EsSu_NumPr_Grad, EsSu_Observacio, EsSu_FecUl_Modi,
EsSu_Cruce_Perm)
    Values
    (@codEs_Subu, @INSubUnidad, @INEstudiante, @INUltNi_Apro,
@INUsuario, @INPensu_Actu, @InFecha_Ingr, @InFecha_Egre,
@InCondi_Acad, @InProme_Egre, @InProme_Grad, @InProme_Fina,
@InProye_Grad, @InNumPr_Grad, @InObservacio, getdate(),
@InCruce_Perm)

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
```

```
        Set @OutSql = 'OK'
    End


CREATE PROCEDURE SPMI_ESTUD_SUBU_MODIFICA
@INCodEs_Subu               VARCHAR(10)       ,
@INUltNi_Apro               VARCHAR(2)        ,
@INFecha_Egre               DATETIME   ,
@INCondi_Acad               VARCHAR(1)        ,
@INProye_Grad                       VARCHAR(4)        ,
@INNumPr_Grad               TINYINT      ,
@INObservacio               VARCHAR(120) ,
@INCruce_Perm               TINYINT      ,
@INUsuario                  VARCHAR(15)       ,
@OUTSql                     VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

     Update TBMI_Estud_SubU
     Set
     EsSu_UltNi_Apro = @INUltNi_Apro,
     EsSu_Fecha_Egre = @InFecha_Egre,
     EsSu_Condi_Acad = @InCondi_Acad,
     EsSu_Proye_Grad = @InProye_Grad,
     EsSu_NumPr_Grad = @InNumPr_Grad,
     EsSu_Observacio = @InObservacio,
     EsSu_Cruce_Perm = @InCruce_Perm,
     EsSu_Usuario = @INUsuario,
     EsSu_FecUl_Modi = getdate()
     Where EsSu_CodEs_Subu = @INcodEs_Subu

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
    Set @OutSql = 'NO'
else
    if (@rowcount_var = 1)
       Set @OutSql = 'OK'
    else
       Set @OutSql = 'NO'
```

```
CREATE PROCEDURE SPMI_HORAR_GRUP_ELIMINA
@INCodigo               VARCHAR(2)        ,
@INGrupo                VARCHAR(8)        ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Delete From TBMI_Horar_Grup
Where HoGr_Codigo   = @InCodigo
and   HoGr_Grupo    = @InGrupo


SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'

CREATE PROCEDURE SPMI_HORAR_GRUP_INSERTA
@INGrupo                VARCHAR(8)        ,
@INAsignatura           VARCHAR(7)        ,
@INSeccion              VARCHAR(1)        ,
@INDia_Sema             VARCHAR(1)        ,
@INHora_Clas            VARCHAR(1)        ,
@INUsuario              VARCHAR(15)       ,
@INAnio_Lect            VARCHAR(4)        ,
@INCiclo          VARCHAR(2)        ,
@INCurso                VARCHAR(6)        ,
@INDescr_Hora                VARCHAR(15),
@OUTCodigo              VARCHAR(2) OUTPUT,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @contador int, @Docente varchar(5), @Cruce
int
```

```
--Valido que el registro por asignatura, seccion, dia de la semana, horario
de clase
Select @contador = count(*)
From TBMI_Horar_Grup
Where HoGr_Grupo     = @InGrupo
And     HoGr_Asignatura       = @InAsignatura
And     HoGr_Seccion  = @InSeccion
And     HoGr_Dia_Sema       = @InDia_Sema
And     HoGr_Hora_Clas       = @InHora_Clas

--Reviso si existe otra asignatura con la misma seccion, dia de la semana
y hora clase
if @contador = 0
  Begin

    Select @contador = count(*)
    From TBMI_Horar_Grup
    Where HoGr_Grupo        = @InGrupo
    And     HoGr_Seccion      = @InSeccion
    And     HoGr_Dia_Sema   = @InDia_Sema
    And     HoGr_Hora_Clas   = @InHora_Clas

    if @contador = 0
      -- Reviso si al profesor de esa asignatura se le cruzan las materias
      Begin
            --Obtengo el docente que dicta esa asignatura
            Select @Docente = DoAs_Docente
            From TBMI_Docen_Asig
            Where DoAs_Asignatura   = @InAsignatura
            And  DoAs_Grupo          = @InGrupo
               And  DoAs_Anio_Lect           = @InAnio_Lect
                     And  DoAs_Ciclo                 = @InCiclo
                     And  DoAs_Tipo_Curs           = @InCurso

            if @Docente <> ''
              Begin
                --Reviso si el mismo profesor está dando una materia en el
mismo horario
                  Select @Cruce = count(*)
                  From TBMI_Docen_Asig, TBMI_Horar_Grup
                  Where DoAs_Docente        = @Docente
                  And  DoAs_Grupo             <> @InGrupo
```

```
                And  DoAs_Anio_Lect          = @InAnio_Lect
                And  DoAs_Ciclo              = @InCiclo
                And  DoAs_Tipo_Curs          = @InCurso
                And  HoGr_Grupo              = DoAs_Grupo
                And  HoGr_Asignatura         = DoAs_Asignatura
                And  HoGr_Seccion            = @InSeccion
                  And  HoGr_Dia_Sema         = @InDia_Sema
                       And  HoGr_Hora_Clas          = @InHora_Clas

                if @Cruce > 0
                  Set @OutSql = 'CR'
              End  --Fin de Docente


          if @Docente <> '' and @Cruce = 0
            Begin
              Select @OutCodigo =
RTrim(Cast(Coalesce(Max(HoGr_Codigo),0)+1 as VarChar(2)))
              From TBMI_Horar_Grup
              Where HoGr_Grupo    = @InGrupo

              set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) +
@OutCodigo

             Insert into TBMI_Horar_Grup
             (HoGr_Codigo, HoGr_Grupo, HoGr_Asignatura,
HoGr_Seccion, HoGr_Dia_Sema, HoGr_Hora_Clas, HoGr_Usuario,
HoGr_FecUl_Modi, HoGr_Descr_Hora)
             Values
             (@OutCodigo, @InGrupo, @InAsignatura, @InSeccion,
@InDia_Sema, @InHora_Clas, @InUsuario, getdate(), @InDescr_Hora)

              SELECT @error_var = @@ERROR

             If (@error_var <> 0)
                Set @OutSql = 'NO'
             else
                Set @OutSql = 'OK'
            End
      End
    else
      Set @OutSql = 'AS'
   End
```

```
else
    Set @OutSql = 'DU'


CREATE PROCEDURE SPMI_HORAR_GRUP_MODIFICA
@INCodigo               VARCHAR(2)      ,
@INGrupo                VARCHAR(8)      ,
@INAsign_Ante           VARCHAR(7)      ,
@INAsign_Actu           VARCHAR(7)      ,
@INSeccion              VARCHAR(1)      ,
@INDia_Sema             VARCHAR(1)      ,
@INHora_Clas            VARCHAR(1)      ,
@INUsuario              VARCHAR(15)     ,
@INAnio_Lect            VARCHAR(4)      ,
@INCiclo        VARCHAR(2)      ,
@INCurso                VARCHAR(6)      ,
@INDescr_Hora                   VARCHAR(15),
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int, @Docente varchar(5),
@Cruce int

--Valido que el registro por asignatura, seccion, dia de la semana, horario
de clase no existan
Select @rowcount_var = count(*)
From TBMI_Horar_Grup
Where HoGr_Grupo    = @InGrupo
And     HoGr_Asignatura     = @InAsign_Actu
And     HoGr_Seccion  = @InSeccion
And     HoGr_Dia_Sema       = @InDia_Sema
And     HoGr_Hora_Clas      = @InHora_Clas

--Reviso si existe otra asignatura con la misma seccion, dia de la semana
y hora clase
if @rowcount_var = 0 or @InAsign_Actu = @InAsign_Ante
  Begin
    Select @rowcount_var = count(*)
    From TBMI_Horar_Grup
    Where HoGr_Grupo        = @InGrupo
    And     HoGr_Seccion        = @InSeccion
```

```
And     HoGr_Dia_Sema    = @InDia_Sema
And     HoGr_Hora_Clas   = @InHora_Clas
and     HoGr_Asignatura     <> @InAsign_Ante

if @rowcount_var = 0
  -- Reviso si al profesor de esa asignatura se le cruzan las materias
  Begin
        --Obtengo el docente que dicta esa asignatura
        Select @Docente = DoAs_Docente
        From TBMI_Docen_Asig
        Where DoAs_Asignatura   = @InAsign_Actu
        And  DoAs_Grupo         = @InGrupo
          And  DoAs_Anio_Lect           = @InAnio_Lect
               And  DoAs_Ciclo                  = @InCiclo
               And  DoAs_Tipo_Curs              = @InCurso

        if @Docente <> ''
          Begin
            --Reviso si el mismo profesor está dando una materia en el
mismo horario
            Select @Cruce = count(*)
            From TBMI_Docen_Asig, TBMI_Horar_Grup
            Where DoAs_Docente          = @Docente
            And  DoAs_Grupo             <> @InGrupo
            And  DoAs_Anio_Lect         = @InAnio_Lect
            And  DoAs_Ciclo             = @InCiclo
            And  DoAs_Tipo_Curs         = @InCurso
            And  HoGr_Grupo             = DoAs_Grupo
            And  HoGr_Asignatura        = DoAs_Asignatura
            And  HoGr_Seccion           = @InSeccion
              And  HoGr_Dia_Sema        = @InDia_Sema
                   And  HoGr_Hora_Clas          = @InHora_Clas

            if @Cruce > 0
              Set @OutSql = 'CR'
          End  --Fin de Docente


        if @Docente <> '' and @Cruce = 0
          Begin
            Update TBMI_Horar_Grup
              Set
                HoGr_Asignatura   = @InAsign_Actu,
```

```
                    HoGr_Seccion       = @InSeccion,
                    HoGr_Dia_Sema    = @InDia_Sema,
                    HoGr_Hora_Clas    = @InHora_Clas,
                    HoGr_Descr_Hora = @InDescr_Hora,
                    HoGr_Usuario       = @InUsuario,
                    HoGr_FecUl_Modi  = getdate()
                  Where HoGr_Codigo  = @InCodigo
                  and   HoGr_Grupo     = @InGrupo

                  SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

                  If (@error_var <> 0)
                    Set @OutSql = 'NO'
                  else
                    if (@rowcount_var = 1)
                      Set @OutSql = 'OK'
                    else
                      Set @OutSql = 'NO'
              End
      End
    else
              Set @OutSql = 'AS'
  End
else
  Set @OutSql = 'DU'


CREATE  PROCEDURE SPMI_HORARIOS_ELIMINA
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Delete From TBMI_Horar_Grup

Delete From TBMI_Horar_Estu

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
```

```
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMI_TIPO_CURS_INSERTA
@INSubunidad          VARCHAR(4)        ,
@INDescripcio         VARCHAR(60)       ,
@INNumer_Parc         NUMERIC(2),
@INNumEx_Adic         NUMERIC(2),
@INMinAp_Asig                 NUMERIC(2),
@INMinPa_Susp         NUMERIC(2),
@INMinAp_Susp         NUMERIC(2),
@INMinPo_Asis         NUMERIC(2),
@INUsuario            VARCHAR(15)       ,
@OUTCodigo            VARCHAR(6) OUTPUT,
@OUTSql               VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiCu_Codigo),0)+1 as
VarChar(6)))
From TBMI_Tipo_Curs
Where TiCu_Subunidad = @InSubunidad

set @OutCodigo = replicate('0', 6 - len(@OutCodigo) ) + @OutCodigo

if @OutCodigo = '000001'
  begin
    set @OutCodigo = @InSubUnidad + '01'
  end

Insert into TBMI_Tipo_Curs
(TiCu_Codigo, TiCu_Descripcio, TiCu_Subunidad, TiCu_Numer_Parc,
TiCu_NumEx_Adic, TiCu_MinAp_Asig, TiCu_MinPa_Susp,
TiCu_MinAp_Susp, TiCu_MinPo_Asis,  TiCu_Usuario, TiCu_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InSubunidad, @InNumer_Parc,
@InNumEx_Adic, @InMinAp_Asig, @InMinPa_Susp,  @InMinAp_Susp,
@InMinPo_Asis,  @InUsuario, getdate())

SELECT @error_var = @@ERROR
```

```
If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMI_TIPO_CURS_MODIFICA
@INCodigo            VARCHAR(6)     ,
@INDescripcio        VARCHAR(60)      ,
@INSubunidad         VARCHAR(4)       ,
@INNumer_Parc        NUMERIC(2),
@INNumEx_Adic        NUMERIC(2),
@INMinAp_Asig        NUMERIC(2),
@INMinPa_Susp        NUMERIC(2),
@INMinAp_Susp        NUMERIC(2),
@INMinPo_Asis        NUMERIC(2),
@INUsuario           VARCHAR(15)      ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMI_Tipo_Curs
Set
   TiCu_Descripcio     = @InDescripcio,
   TiCu_Numer_Parc     = @InNumer_Parc,
   TiCu_NumEx_Adic     = @InNumEx_Adic,
   TiCu_MinAp_Asig     = @InMinAp_Asig,
   TiCu_MinPa_Susp     = @InMinPa_Susp,
   TiCu_MinAp_Susp     = @InMinAp_Susp,
   TiCu_MinPo_Asis     = @InMinPo_Asis,
   TiCu_Usuario        = @InUsuario,
   TiCu_FecUl_Modi     = getdate()
Where TiCu_SubUnidad       = @InSubunidad
and   TiCu_Codigo      = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
```

```
    if (@rowcount_var = 1)
      Set @OutSql = 'OK'
    else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMM_BANCO_INSERTA
@INNumer_Cuen              VARCHAR(10)      ,
@INDescripcio              VARCHAR(60)      ,
@INTipo_Cuen               VARCHAR(1)       ,
@INUsuario                 VARCHAR(15)      ,
@OUTCodigo                 VARCHAR(3) OUTPUT,
@OUTSql                    VARCHAR(2) OUTPUT
AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Banc_Codigo),0)+1 as
VarChar(3)))
From TBMM_Banco

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMM_Banco
(Banc_Codigo, Banc_Numer_Cuen, Banc_Descripcio, Banc_Tipo_Cuen,
Banc_Usuario, Banc_FecUl_Modi)
Values
(@OutCodigo, @InNumer_Cuen, @InDescripcio, @InTipo_Cuen,
@InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE PROCEDURE SPMM_BANCO_MODIFICA
@INCodigo             VARCHAR(3)     ,
@INNumer_Cuen         VARCHAR(10)      ,
@INDescripcio         VARCHAR(60)      ,
@INTipo_Cuen          VARCHAR(1)       ,
```

```
@INUsuario              VARCHAR(15)       ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Banco
Set
   Banc_Numer_Cuen  = @InNumer_Cuen,
   Banc_Descripcio      = @InDescripcio,
   Banc_Tipo_Cuen     = @InTipo_Cuen,
   Banc_Usuario                  = @InUsuario,
   Banc_FecUl_Modi    = getdate()
Where  Banc_Codigo   = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE   PROCEDURE SPMM_CATEGORIA_INSERTA
@INObservacio          VARCHAR(30),
@INUsuario             VARCHAR(15),
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                        VARCHAR(2) OUTPUT
AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Cate_Codigo),0)+1 as
VarChar(3)))
From TBMM_Categoria

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMM_Categoria
```

```
(Cate_Codigo, Cate_Observacio, Cate_Usuario, Cate_FecUl_Modi)
Values
(@OutCodigo, @InObservacio, @InUsuario, getdate())

select @error_var=@@ERROR

if (@error_var <> 0)
 set @OUTSql = 'NO'
ELSE
 set @OUTSql = 'OK'


CREATE  PROCEDURE SPMM_CATEGORIA_MODIFICA
@INCodigo               VARCHAR(3) OUTPUT,
@INObservacio           VARCHAR(30),
@INUsuario              VARCHAR(15),
@OUTSql                         VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Categoria
Set
   Cate_Observacio     = @InObservacio,
   Cate_Usuario= @InUsuario,
   Cate_FecUl_Modi     = getdate()
Where Cate_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMM_COLEGIO_INSERTA
@INCategoria            VARCHAR(3)        ,
@INNombre               VARCHAR(60)       ,
```

```
@INTipo                 VARCHAR(1)        ,
@INUsuario              VARCHAR(15)       ,
@OUTCodigo              VARCHAR(4) OUTPUT,
@OUTSql                         VARCHAR(2) OUTPUT
AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(Cole_Codigo),0)+1 as
VarChar(4)))
From TBMM_Colegio

set @OutCodigo = replicate('0', 4 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMM_Colegio
(Cole_Codigo, Cole_Categoria, Cole_Nombre, Cole_Tipo, Cole_Usuario,
Cole_FecUl_Modi)
Values
(@OutCodigo, @InCategoria, @InNombre, @InTipo,  @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE PROCEDURE SPMM_COLEGIO_MODIFICA
@INCodigo               VARCHAR(4)     ,
@INCategoria            VARCHAR(3)        ,
@INNombre               VARCHAR(60)       ,
@INTipo          VARCHAR(1)        ,
@INUsuario              VARCHAR(15)       ,
@OUTSql                 VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Colegio
Set
```

```
    Cole_Categoria      = @InCategoria,
    Cole_Nombre         = @InNombre,
    Cole_Tipo           = @InTipo,
    Cole_Usuario = @InUsuario,
    Cole_FecUl_Modi     = getdate()
Where  Cole_Codigo    = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE  PROCEDURE SPMM_CONCE_COBR_INSERTA
@INDescripcio          VARCHAR(60)       ,
@INUsuario             VARCHAR(15)       ,
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(CoCo_Codigo),0)+1 as
VarChar(3)))
From TBMM_Conce_Cobr

set @OutCodigo = replicate('0', 3 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMM_Conce_Cobr
(CoCo_Codigo, CoCo_Descripcio, CoCo_Usuario, CoCo_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
```

```
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMM_CONCE_COBR_MODIFICA
@INCodigo            VARCHAR(3)     ,
@INDescripcio        VARCHAR(60)    ,
@INUsuario           VARCHAR(15)    ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Conce_Cobr
Set
   CoCo_Descripcio    = @InDescripcio,
   CoCo_Usuario       = @InUsuario,
   CoCo_FecUl_Modi    = getdate()
Where CoCo_Codigo     = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE PROCEDURE SPMM_CUOTA_INSERTA
@INEstud_Subu        VARCHAR(10)    ,
@INNumer_Matr        VARCHAR(10)    ,
@INNúmero            VARCHAR(2)     ,
@INValor             NUMERIC(8,2)   ,
@INEstado            VARCHAR(1)     ,
@INPorce_Cuot        NUMERIC(8,2)   ,
@ININter_Mens        NUMERIC(8,2)   ,
@INFecha_Venc        datetime       ,
@INUsuario           VARCHAR(15)    ,
```

```
@OUTSql                 VARCHAR(2)  OUTPUT

AS

DECLARE @error_var int

--Inserto en la tabla tbmm_cuota
Insert into TBMM_Cuota
(Cuot_Estud_Subu, Cuot_Numer_Matr, Cuot_Número, Cuot_Valor,
Cuot_Estado, Cuot_Fecha_Venc, Cuot_Porce_Cuot, Cuot_Inter_Mens,
Cuot_Usuario, Cuot_FecUl_Modi)
Values
(@InEstud_Subu, @InNumer_Matr, @InNúmero, @InValor, @InEstado,
@InFecha_Venc, @InPorce_Cuot, @InInter_Mens, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE PROCEDURE SPMM_CUOTA_MODIFICA
@INEstud_Subu       VARCHAR(10)      ,
@INNumer_Matr       VARCHAR(10)      ,
@INNúmero           VARCHAR(2)       ,
@INValor            NUMERIC(8,2)     ,
@INPorce_Cuot       NUMERIC(8,2)     ,
@ININter_Mens       NUMERIC(8,2)     ,
@INFecha_Venc       datetime      ,
@INUsuario          VARCHAR(15)      ,
@OUTSql             VARCHAR(2)  OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

--Actualizo en la tabla tbmm_cuota
Update TBMM_Cuota
Set         Cuot_Valor = @InValor,
    Cuot_Fecha_Venc = @InFecha_Venc,
```

```
    Cuot_Porce_Cuot = @InPorce_Cuot,
    Cuot_Inter_Mens = @InInter_Mens,
    Cuot_Usuario = @InUsuario,
    Cuot_FecUl_Modi = getdate()
Where    Cuot_Estud_Subu = @InEstud_Subu
And      Cuot_Numer_Matr = @InNumer_Matr
And      Cuot_Número = @InNúmero


SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
         Set @OutSql = 'OK'
   else
         Set @OutSql = 'NO'


CREATE   PROCEDURE SPMM_DESCU_ARAN_INSERTA
@INDescripcio          VARCHAR(60),
@INLetra_Codi          VARCHAR(1),
@INTipo                VARCHAR(1),
@INValor         NUMERIC(8,2),
@INUsuario             VARCHAR(15),
@OUTCodigo             VARCHAR(3) OUTPUT,
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo =
RTrim(Cast(Coalesce(substring(Max(DeAr_codigo),2,3),0)+1 as
VarChar(3)))
From TBMM_Descu_Aran where
substring(DeAr_codigo,1,1)=@InLetra_Codi

set @OutCodigo = replicate('0', 2 - len(@OutCodigo) ) + @OutCodigo
set @OutCodigo = @InLetra_Codi + @OutCodigo

Insert into TBMM_Descu_Aran
```

```
(DeAr_Codigo, DeAr_Descripcio, DeAr_Tipo, DeAr_Valor, DeAr_Usuario,
DeAr_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @INTipo, @INValor, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE   PROCEDURE SPMM_DESCU_ARAN_MODIFICA
@INCodigo              VARCHAR(3),
@INDescripcio          VARCHAR(60),
@INTipo                VARCHAR(1),
@INValor         NUMERIC(8,2),
@INUsuario             VARCHAR(15),
@OUTSql                      VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Descu_Aran
Set
   DeAr_Descripcio     = @InDescripcio,
   DeAr_Tipo           = @InTipo,
   DeAr_valor          = @InValor,
   DeAr_Usuario        = @InUsuario,
   DeAr_FecUl_Modi     = getdate()
Where  Dear_Codigo     = @InCodigo


SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
```

```
    else
      Set @OutSql = 'NO'


CREATE PROCEDURE SPMM_DETAL_MATR_INSERTA
@INEstud_Subu        VARCHAR(10)      ,
@INNumer_Matr        VARCHAR(10)      ,
@INMotivo            VARCHAR(3)       ,
@INTipo          VARCHAR(1)       ,
@INMonto             NUMERIC(8,2)     ,
@INValor_Calc        NUMERIC(8,2)     ,
@INUsuario           VARCHAR(15)      ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Begin
       --Inserto en la tabla tbmm_detal_matr
    Insert into TBMM_Detal_Matr
    (DeMa_Estud_Subu, DeMa_Numer_Matr, DeMa_Motivo, DeMa_Tipo,
DeMa_Monto, DeMa_Valor_Calc, DeMa_Usuario, DeMa_FecUl_Modi)
    Values
    (@InEstud_Subu, @InNumer_Matr, @InMotivo, @InTipo, @InMonto,
@InValor_Calc, @InUsuario, getdate())

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'

End


CREATE  PROCEDURE SPMM_ENTRE_CALI_INSERTA
@INAnio_Lect         VARCHAR(4)       ,
@INCiclo         VARCHAR(2)       ,
@INTipo_Curs         VARCHAR(6)       ,
@INTipo_Cali         VARCHAR(6)       ,
@INFecMa_Entr        datetime      ,
```

```
@INUsuario          VARCHAR(15)       ,
@OUTSql             VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Insert into TBMC_Entre_Cali
(EnCa_Anio_Lect, EnCa_Ciclo, EnCa_Tipo_Curs, EnCa_Tipo_Cali,
EnCa_FecMa_Entr, EnCa_Usuario, EnCa_FecUl_Modi)
Values
(@InAnio_Lect, @InCiclo, @InTipo_Curs, @InTipo_Cali,
@InFecMa_Entr, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE    PROCEDURE SPMM_ENTRE_CALI_MODIFICA
@INAnio_Lect        VARCHAR(4)        ,
@INCiclo        VARCHAR(2)        ,
@INTipo_Curs        VARCHAR(6)        ,
@INTipo_Cali        VARCHAR(6)        ,
@INFecMa_Entr       datetime      ,
@INUsuario          VARCHAR(15)       ,
@OUTSql             VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMC_Entre_Cali
Set
   EnCa_FecMa_Entr  = @inFecMa_Entr,
   EnCa_Usuario     = @InUsuario,
   EnCa_FecUl_Modi  = getdate()
Where  EnCa_Anio_Lect= @InAnio_Lect
and EnCa_Ciclo = @InCiclo
and EnCa_Tipo_Curs = @InTipo_Curs
```

```
and EnCa_Tipo_Cali = @InTipo_Cali

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'
CREATE   PROCEDURE SPMM_INCRE_MATR_INSERTA
@INSubunidad          VARCHAR(4),
@INDescripcio         VARCHAR(30),
@INPorce_Incr         NUMERIC(8,2),
@INUsuario            VARCHAR(15),
@OUTTipo_Matr               VARCHAR(2) OUTPUT,
@OUTSql                     VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutTipo_Matr =
RTrim(Cast(Coalesce(Max(InMa_Tipo_Matr),0)+1 as VarChar(2)))
From TBMM_Incre_Matr

set @OutTipo_Matr = replicate('0', 2 - len(@OutTipo_Matr) ) +
@OutTipo_Matr

Insert into TBMM_Incre_Matr
(InMa_Tipo_Matr, InMa_Descripcio, InMa_Subunidad, InMa_Porce_Incr,
InMa_Usuario, InMa_FecUl_Modi)
Values
(@OutTipo_Matr, @InDescripcio, @InSubunidad, @InPorce_Incr,
@InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'
```

```
CREATE   PROCEDURE SPMM_INCRE_MATR_MODIFICA
@INTipo_Matr   VARCHAR(2),
@INDescripcio  VARCHAR(30),
@INSubunidad   VARCHAR(4),
@INPorce_Incr  NUMERIC(8,2),
@INUsuario     VARCHAR(15),
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Incre_Matr
Set
   InMa_Descripcio = @InDescripcio,
   InMa_Porce_Incr = @InPorce_Incr,
   InMa_Usuario    = @InUsuario,
   InMa_FecUl_Modi = getdate()
Where  InMa_Tipo_Matr = @InTipo_Matr
and    InMa_Subunidad = @InSubunidad

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
     Set @OutSql = 'OK'
   else
     Set @OutSql = 'NO'


CREATE   PROCEDURE SPMM_MATRI_ASIG_INSERTA
@INCategoria           VARCHAR(3)        ,
@INSubunidad           VARCHAR(4)        ,
@INValor               NUMERIC (8,2)     ,
@INLimite              NUMERIC (1)       ,
@INUsuario             VARCHAR(15)       ,
@OUTSql                        VARCHAR(2) OUTPUT
AS
```

```
DECLARE @error_var int

Insert into TBMM_Matri_Asig
(MaAs_Categoria, MaAs_Valor, MaAs_Limite, MaAs_Subunidad,
MaAs_Usuario, MaAs_FecUl_Modi)
Values
(@InCategoria, @InValor, @InLimite, @InSubunidad, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE   PROCEDURE SPMM_MATRI_ASIG_MODIFICA
@INCategoria          VARCHAR(3)       ,
@INValor              NUMERIC (8,2)    ,
@INLimite             NUMERIC (1)      ,
@INSubunidad          VARCHAR(4)       ,
@INUsuario            VARCHAR(15)      ,
@OUTSql                     VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Matri_Asig
Set   MaAs_Valor       = @InValor,
     MaAs_limite = @InLimite,
     MaAs_Usuario      = @InUsuario,
     MaAs_FecUl_Modi = getdate()
Where MaAs_Categoria       = @InCategoria
and   MaAs_Subunidad       = @InSubunidad

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
```

```
    Set @OutSql = 'OK'
  else
    Set @OutSql = 'NO'


CREATE   PROCEDURE SPMM_MATRI_GLOB_INSERTA
@INCategoria          VARCHAR(3)        ,
@INValor              NUMERIC (8,2)     ,
@INSubunidad          VARCHAR(4)        ,
@INUsuario            VARCHAR(15)       ,
@OUTSql                     VARCHAR(2) OUTPUT
AS

DECLARE @error_var int
Insert into TBMM_Matri_Glob
(MaGl_Categoria, MaGl_Valor, MaGl_Subunidad, MaGl_Usuario,
MaGl_FecUl_Modi)
Values
(@InCategoria, @InValor, @InSubunidad,  @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE   PROCEDURE SPMM_MATRI_GLOB_MODIFICA
@INCategoria          VARCHAR(3)        ,
@INValor              NUMERIC (8,2)     ,
@INSubunidad          VARCHAR(4)        ,
@INUsuario            VARCHAR(15)       ,
@OUTSql                     VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Matri_Glob
Set
   MaGl_Valor         = @InValor,
   MaGl_Usuario       = @InUsuario,
```

```
   MaGl_FecUl_Modi   = getdate()
Where  MaGl_Categoria       = @InCategoria
and    MaGl_Subunidad       = @InSubunidad

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE PROCEDURE SPMM_ORDEN_PAGO_ELIMINA
@INEstud_Subu        VARCHAR(10)       ,
@INNumer_Matr        VARCHAR(10)       ,
@OUTSql              VARCHAR(2)  OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

-- Elimino detalle de la matrícula
Delete from TBMM_Detal_Matr
Where     DeMa_Estud_Subu = @InEstud_Subu
And       DeMa_Numer_Matr = @InNumer_Matr

-- Elimino detalle de cuotas
Delete from TBMM_Cuota
Where     Cuot_Estud_Subu = @InEstud_Subu
And       Cuot_Numer_Matr = @InNumer_Matr

-- Elimino cabecera de orden
Delete from TBMM_Orden_Pago
Where     OrPa_Estud_Subu = @InEstud_Subu
And       OrPa_Numer_Matr = @InNumer_Matr

SELECT @rowcount_var = @@ROWCOUNT

If (@rowcount_var > 0)
   Set @OutSql = 'OK'
```

```
else
   Set @OutSql = 'NO'


CREATE PROCEDURE SPMM_ORDEN_PAGO_INSERTA
@INEstud_Subu        VARCHAR(10)      ,
@INNumer_Matr        VARCHAR(10)      ,
@INConce_Cobr        VARCHAR(3)       ,
@INBanco             VARCHAR(3)       ,
@INFecha_Emis        datetime         ,
@INValor_Tota        NUMERIC(8,2)     ,
@INObservacio        VARCHAR(120)     ,
@INUsuario           VARCHAR(15)      ,
@INAnio_Lect         VARCHAR(4)       ,
@INCiclo          VARCHAR(2)          ,
@INTipo_Curs         VARCHAR(6)       ,
@INNumer_Reci        VARCHAR(4)     ,
@OutNumer_Reci       VARCHAR(4)  OUTPUT,
@OUTSql              VARCHAR(2)  OUTPUT

AS

DECLARE @error_var int


If @InNumer_Reci <> ""
  Begin
    --Inserto en la tabla tbmm_orden_pago
    Insert into TBMM_Orden_Pago
    (OrPa_Estud_Subu, OrPa_Numer_Matr, OrPa_Conce_Cobr,
OrPa_Banco, OrPa_Numer_Reci, OrPa_Fecha_Emis, OrPa_Valor_Tota,
OrPa_Usuario, OrPa_FecUl_Modi)
    Values
    (@InEstud_Subu, @InNumer_Matr, @InConce_Cobr, @InBanco,
@InNumer_Reci, @InFecha_Emis, @InValor_Tota, @InUsuario,
getdate())

    set @OutNumer_Reci = @InNumer_Reci

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
       Set @OutSql = 'NO'
```

```
    else
       Set @OutSql = 'OK'

  End
else
  Begin
    Select @OutNumer_Reci =
RTrim(Cast(Coalesce(Max(PeLe_UltNu_Reci),0)+1 as VarChar(4)))
    From TBMM_Perio_Lect
    Where PeLe_Anio_Lect = @InAnio_Lect
    And   PeLe_Ciclo = @InCiclo
    And   PeLe_Tipo_Curs = @InTipo_Curs

    set @OutNumer_Reci = replicate('0', 4 - len(@OutNumer_Reci) ) +
@OutNumer_Reci

    Update TBMM_Perio_Lect
    Set PeLe_UltNu_Reci = @OutNumer_Reci
    Where PeLe_Anio_Lect = @InAnio_Lect
    And   PeLe_Ciclo = @InCiclo
    And   PeLe_Tipo_Curs = @InTipo_Curs

    --Inserto en la tabla tbmm_orden_pago
    Insert into TBMM_Orden_Pago
    (OrPa_Estud_Subu, OrPa_Numer_Matr, OrPa_Conce_Cobr,
OrPa_Banco, OrPa_Numer_Reci, OrPa_Fecha_Emis, OrPa_Valor_Tota,
OrPa_Usuario, OrPa_FecUl_Modi)
    Values
    (@InEstud_Subu, @InNumer_Matr, @InConce_Cobr, @InBanco,
@OutNumer_Reci, @InFecha_Emis, @InValor_Tota, @InUsuario,
getdate())

    SELECT @error_var = @@ERROR

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
       Set @OutSql = 'OK'
  End


CREATE    PROCEDURE SPMM_PERIO_LECT_INSERTA
@INAnio_Lect              VARCHAR(4)         ,
```

```
@INCiclo        VARCHAR(2)      ,
@INTipo_Curs        VARCHAR(6)      ,
@INFecha_Matr               datetime    ,
@INFecha_Inic       datetime    ,
@INFecha_Fin        datetime    ,
@INEstado           VARCHAR(1)      ,
@INMaxNu_Cred       numeric(1)  ,
@INUsuario          VARCHAR(15)     ,
@OUTSql                     VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Insert into TBMM_Perio_Lect
(PeLe_Anio_Lect, PeLe_Ciclo, PeLe_Tipo_Curs, PeLe_Fecha_Inic,
PeLe_Fecha_Fin, PeLe_Fecha_Matr, PeLe_MaxNu_Cred,
PeLe_UltNu_Reci, PeLe_Estado, PeLe_Usuario, PeLe_FecUl_Modi)
Values
(@InAnio_Lect, @InCiclo, @InTipo_Curs, @InFecha_Inic, @InFecha_Fin,
@InFecha_Matr, @InMaxNu_Cred, '0000',  @InEstado, @InUsuario,
getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE   PROCEDURE SPMM_PERIO_LECT_MODIFICA
@INAnio_Lect        VARCHAR(4)      ,
@INCiclo        VARCHAR(2)      ,
@INTipo_Curs        VARCHAR(6)      ,
@INFecha_Matr               datetime    ,
@INFecha_Inic       datetime    ,
@INFecha_Fin        datetime    ,
@INEstado           VARCHAR(1)      ,
@INMaxNu_Cred       Numeric(1)  ,
@INUsuario          VARCHAR(15)     ,
@OUTSql             VARCHAR(2) OUTPUT
```

```
AS

DECLARE @error_var int, @rowcount_var int

Update TBMM_Perio_Lect
Set
   PeLe_Fecha_Inic     = @inFecha_Inic,
   PeLe_Fecha_Fin      = @InFecha_Fin,
   PeLe_Fecha_Matr     = @InFecha_Matr,
   PeLe_Estado         = @INEstado,
   PeLe_MaxNu_Cred  = @InMaxNu_Cred,
   PeLe_Usuario        = @InUsuario,
   PeLe_FecUl_Modi     = getdate()
Where  PeLe_Anio_Lect= @InAnio_Lect
and PeLe_Ciclo = @InCiclo
and PeLe_Tipo_Curs = @InTipo_Curs

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'


CREATE PROCEDURE SPMM_PERIO_LECT_SECUENCIA
@InAnio_Lect            VARCHAR(4),
@InCiclo          VARCHAR(2),
@InTipo_Curs            VARCHAR(6),
@OutNumer_Reci      VARCHAR(4)  OUTPUT,
@OUTSql                 VARCHAR(2)  OUTPUT

AS

DECLARE @error_var int, @rowcount_var int

Begin
    Select @OutNumer_Reci =
RTrim(Cast(Coalesce(Max(PeLe_UltNu_Reci),0)+1 as VarChar(4)))
    From TBMM_Perio_Lect
```

```
    Where PeLe_Anio_Lect = @InAnio_Lect
    And   PeLe_Ciclo = @InCiclo
    And   PeLe_Tipo_Curs = @InTipo_Curs

    set @OutNumer_Reci = replicate('0', 4 - len(@OutNumer_Reci) ) +
@OutNumer_Reci

    Update TBMM_Perio_Lect
    Set PeLe_UltNu_Reci = @OutNumer_Reci
    Where PeLe_Anio_Lect = @InAnio_Lect
    And   PeLe_Ciclo = @InCiclo
    And   PeLe_Tipo_Curs = @InTipo_Curs

    SELECT @error_var = @@ERROR, @rowcount_var =
@@ROWCOUNT

    If (@error_var <> 0)
       Set @OutSql = 'NO'
    else
      if (@rowcount_var = 1)
         Set @OutSql = 'OK'
       else
         Set @OutSql = 'NO'
End


CREATE  PROCEDURE SPMN_TIPO_PROY_INSERTA
@INDescripcio          VARCHAR(30)      ,
@INUsuario             VARCHAR(15)      ,
@OUTCodigo             VARCHAR(1) OUTPUT,
@OUTSql                VARCHAR(2) OUTPUT

AS

DECLARE @error_var int

Select @OutCodigo = RTrim(Cast(Coalesce(Max(TiPr_Codigo),0)+1 as
VarChar(1)))
From TBMN_Tipo_Proy

set @OutCodigo = replicate('0', 1 - len(@OutCodigo) ) + @OutCodigo

Insert into TBMN_Tipo_Proy
```

```
(TiPr_Codigo, TiPr_Descripcio, TiPr_Usuario, TiPr_FecUl_Modi)
Values
(@OutCodigo, @InDescripcio, @InUsuario, getdate())

SELECT @error_var = @@ERROR

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   Set @OutSql = 'OK'


CREATE  PROCEDURE SPMN_TIPO_PROY_MODIFICA
@INCodigo            VARCHAR(1)     ,
@INDescripcio        VARCHAR(30)      ,
@INUsuario           VARCHAR(15)      ,
@OUTSql              VARCHAR(2) OUTPUT

AS

DECLARE @error_var int, @rowcount_var int
Update TBMN_Tipo_Proy
Set
   TiPr_Descripcio      = @InDescripcio,
   TiPr_Usuario         = @InUsuario,
   TiPr_FecUl_Modi      = getdate()
Where TiPr_Codigo       = @InCodigo

SELECT @error_var = @@ERROR, @rowcount_var = @@ROWCOUNT

If (@error_var <> 0)
   Set @OutSql = 'NO'
else
   if (@rowcount_var = 1)
      Set @OutSql = 'OK'
   else
      Set @OutSql = 'NO'
```

2.4. Scripts y Triggers para Auditoria del Sistema

Las tablas de Auditoria fueron creadas con la finalidad de controlar la actualización de los datos del sistema. De esta manera registramos una bitácora transparente para el usuario pero muy relevante para el administrador de la base de datos.

Para el funcionamiento de las tablas de auditoria, requerimos la creación de las tablas principales a auditar, como son:

```
CREATE TABLE [dbo].[AUDIT_CALIF_PARC] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_ESTUD_SUBU] [varchar] (10) NULL ,
    [AUDIT_NUMER_CRED] [numeric] (1,0) NULL ,
    [AUDIT_ASIGNATURA] [varchar] (7) NULL,
    [AUDIT_TIPO_CALI] [varchar] (6) NULL,
    [AUDIT_CALIF_TRAB] [varchar] (9)  NULL,
    [AUDIT_FECHA_ENTR] [datetime]  NULL,
    [AUDIT_CALIF_EXAM] [varchar] (9) NULL,
    [AUDIT_CALIF_TOTA] [varchar] (9) NULL,
    [AUDIT_ESTAD_REGI] [varchar] (1) NULL,
    [AUDIT_PORCE_ASIS] [varchar] (9) NULL,
    [AUDIT_OBSERVACIO] [varchar] (120) NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_CALIF_PARC] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT_CALIF_PARC] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
```

```
    )  ON [PRIMARY]
GO

CREATE TABLE [dbo].[AUDIT_CALIF_PROM] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_ESTUD_SUBU] [varchar] (10) NULL ,
    [AUDIT_ASIGNATURA] [varchar] (7) NULL ,
    [AUDIT_NUMER_CRED] [numeric] (1,0) NULL,
    [AUDIT_PROME_CALI] [varchar] (9) NULL,
    [AUDIT_PROME_ASIS] [varchar] (9)  NULL,
    [AUDIT_OBSERVACIO] [varchar] (120)  NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_CALIF_PROM] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT_CALIF_PROM] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_CUOTA] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_ESTUD_SUBU] [varchar] (10) NULL ,
    [AUDIT_NUMER_MATR] [varchar] (10) NULL ,
    [AUDIT_NÚMERO] [varchar] (2) NULL,
    [AUDIT_VALOR] [numeric] (8,2) NULL,
    [AUDIT_FECHA_VENC] [datetime]  NULL,
```

```
    [AUDIT_ESTADO] [varchar] (1) NULL,
    [AUDIT_PORCE_CUOT] [numeric] (8,2) NULL,
    [AUDIT_INTER_MENS] [numeric] (8,2) NULL

) ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_DETAL_MATR] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_ESTUD_SUBU] [varchar] (10) NULL ,
    [AUDIT_NUMER_MATR] [varchar] (10) NULL ,
    [AUDIT_TIPO] [varchar] (1) NULL,
    [AUDIT_MOTIVO] [varchar] (3) NULL,
    [AUDIT_MONTO] [numeric] (8,2) NULL,
    [AUDIT_VALOR_CALC] [numeric] (8,2) NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_DETAL_MATR] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT_DETAL_MATR] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_INCRE_MATR] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
```

```
    [AUDIT_TIPO_MATR] [varchar] (2) NULL ,
    [AUDIT_SUBUNIDAD] [varchar] (4) NULL ,
    [AUDIT_DESCRIPCIO] [varchar] (30) NULL ,
    [AUDIT_PORCE_INCR] [numeric] (8,2)  NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_INCRE_MATR] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT] PRIMARY KEY  NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_MATRI_ASIG] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_CATEGORIA] [varchar] (3) NULL ,
    [AUDIT_VALOR] [numeric] (8,2) NULL ,
    [AUDIT_SUBUNIDAD] [varchar] (4) NULL ,
    [AUDIT_LIMITE] [numeric] (1,0)  NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_MATRI_ASIG] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT_MATRI_ASIG] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_MATRI_GLOB] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
```

```
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_CATEGORIA] [varchar] (3) NULL ,
    [AUDIT_VALOR] [numeric] (8,2) NULL ,
    [AUDIT_SUBUNIDAD] [varchar] (4) NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_MATRI_GLOB] WITH NOCHECK ADD
    CONSTRAINT [PK_AUDIT_MATRI_GLOB] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO


CREATE TABLE [dbo].[AUDIT_ORDEN_PAGO] (
    [id_evento] [uniqueidentifier]  DEFAULT NEWID() NOT NULL,
    [tipo_evento] [char] (10) NOT NULL ,
    [fecha] [datetime] NOT NULL ,
    [descripcion] [char] (50)  NULL ,
    [usuario] [char] (30)  NULL ,
    [terminal] [char] (30)  NULL ,
    [aplicacion] [char] (30)  NULL,
    [AUDIT_ESTUD_SUBU] [varchar] (10) NULL ,
    [AUDIT_NUMER_MATR] [varchar] (10) NULL ,
    [AUDIT_FECHA_EMIS] [datetime]  NULL,
    [AUDIT_CONCE_COBR] [varchar] (3) NULL,
    [AUDIT_BANCO] [varchar] (3) NULL,
    [AUDIT_NUMER_RECI] [varchar] (4) NULL,
    [AUDIT_OBSERVACIO] [varchar] (120) NULL,
    [AUDIT_VALOR_TOTA] [numeric] (8,2) NULL

) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AUDIT_ORDEN_PAGO] WITH NOCHECK ADD
```

```
    CONSTRAINT [PK_AUDIT_ORDEN_PAGO] PRIMARY KEY
NONCLUSTERED
    (
            [id_evento]
    )  ON [PRIMARY]
GO
```

Detallamos a continuación los eventos o triggers a disparar cuando el usuario intente actualizar los datos:

```
-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMC_CALIF_PARC
ON TBMC_CALIF_PARC
FOR DELETE AS
begin
    Insert into AUDIT_CALIF_PARC
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_CRED,AUDIT_ASIGNATURA,AUDIT_TIPO_CA
LI,AUDIT_CALIF_TRAB,AUDIT_FECHA_ENTR,AUDIT_CALIF_EXAM,AU
DIT_CALIF_TOTA,AUDIT_ESTAD_REGI,AUDIT_PORCE_ASIS,AUDIT_
OBSERVACIO)
    select 'Delete', getdate(), 'Eliminacion en TBMC_CALIF_PARC',
SYSTEM_USER, host_name(), APP_NAME(), del.capa_estud_subu,
del.capa_numer_cred, del.capa_asignatura, del.capa_tipo_cali,
del.capa_calif_trab, del.capa_fecha_entr, del.capa_calif_exam,
del.capa_calif_tota, del.capa_estad_regi, del.capa_porce_asis,
del.capa_observacio from deleted del
end
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMC_CALIF_PROM
ON TBMC_CALIF_PROM
FOR DELETE AS
begin
    Insert into AUDIT_CALIF_PROM
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_ASIGNATURA,AUDIT_NUMER_CRED,AUDIT_PROME_
CALI,AUDIT_PROME_ASIS,AUDIT_OBSERVACIO)
```

```
    select 'Delete', getdate(), 'Eliminacion en TBMC_CALIF_PROM',
SYSTEM_USER, host_name(), APP_NAME(), del.capr_estud_subu,
del.capr_asignatura, del.capr_numer_cred, del.capr_prome_cali,
del.capr_prome_asis, del.capr_observacio from deleted del
end

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_CUOTA
ON TBMM_CUOTA
FOR DELETE AS
begin
    Insert into AUDIT_CUOTA
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_NÚMERO,AUDIT_VALOR,AUDI
T_FECHA_VENC,AUDIT_ESTADO,AUDIT_PORCE_CUOT,AUDIT_INTE
R_MENS)
    select 'Delete', getdate(), 'Eliminacion en TBMM_CUOTA',
SYSTEM_USER, host_name(), APP_NAME(), del.cuot_estud_subu,
del.cuot_numer_matr, del.cuot_Número, del.cuot_valor,
del.cuot_fecha_venc, del.cuot_estado, del.cuot_porce_cuot,
del.cuot_inter_mens from deleted del
end
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_DETAL_MATR
ON TBMM_DETAL_MATR
FOR DELETE AS
begin
    Insert into AUDIT_DETAL_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_TIPO,AUDIT_MOTIVO,AUDIT_M
ONTO,AUDIT_VALOR_CALC)
    select 'Delete', getdate(), 'Eliminacion en TBMM_DETAL_MATR',
SYSTEM_USER, host_name(), APP_NAME(), del.dema_estud_subu,
del.dema_numer_matr, del.dema_tipo, del.dema_motivo,
del.dema_monto, del.dema_valor_calc from deleted del
end
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_INCRE_MATR
ON TBMM_INCRE_MATR
```

```
FOR DELETE AS
begin
    Insert into AUDIT_INCRE_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_TIPO_
MATR,AUDIT_SUBUNIDAD,AUDIT_DESCRIPCIO,AUDIT_PORCE_INC
R)
    --select 'Delete', getdate(), 'Eliminacion en TBMM_INCRE_MATR',
SYSTEM_USER, host_name(), APP_NAME(), del.inma_tipo_matr,
del.inma_subunidad, del.inma_descripcio, del.inma_porce_incr from
deleted del
    select 'Delete', getdate(), 'Eliminacion en TBMM_INCRE_MATR',
SYSTEM_USER, host_name(), APP_NAME(), del.inma_tipo_matr,
del.inma_subunidad, del.inma_descripcio, del.inma_porce_incr from
deleted del
end
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_MATRI_ASIG
ON TBMM_MATRI_ASIG
FOR DELETE AS
begin
    Insert into AUDIT_MATRI_ASIG
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD,AUDIT_LIMITE)
    select 'Delete', getdate(), 'Eliminacion en TBMM_MATRI_ASIG',
SYSTEM_USER, host_name(), APP_NAME(), del.maas_categoria,
del.maas_valor, del.maas_subunidad, del.maas_limite from deleted del
end
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_MATRI_GLOB
ON TBMM_MATRI_GLOB
FOR DELETE AS
begin
    Insert into AUDIT_MATRI_GLOB
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD)
    select 'Delete', getdate(), 'Eliminacion en TBMM_MATRI_GLOB',
SYSTEM_USER, host_name(), APP_NAME(), del.magl_categoria,
del.magl_valor, del.magl_subunidad from deleted del
end
```

```
go

-- Creacion del trigger para ELIMINACION
CREATE TRIGGER AUDITdel_TBMM_ORDEN_PAGO
ON TBMM_ORDEN_PAGO
FOR DELETE AS
begin
    Insert into AUDIT_ORDEN_PAGO
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_FECHA_EMIS,AUDIT_CONCE_
COBR,AUDIT_BANCO,AUDIT_NUMER_RECI,AUDIT_OBSERVACIO,AU
DIT_VALOR_TOTA)
    select 'Delete', getdate(), 'Eliminacion en TBMM_ORDEN_PAGO',
SYSTEM_USER, host_name(), APP_NAME(), del.orpa_estud_subu,
del.orpa_numer_matr, del.orpa_fecha_emis, del.orpa_conce_cobr,
del.orpa_banco, del.orpa_numer_reci, del.orpa_observacio,
del.orpa_valor_tota from deleted del
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMC_CALIF_PARC
ON TBMC_CALIF_PARC
FOR INSERT AS
begin
    Insert into AUDIT_CALIF_PARC
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_CRED,AUDIT_ASIGNATURA,AUDIT_TIPO_CA
LI,AUDIT_CALIF_TRAB,AUDIT_FECHA_ENTR,AUDIT_CALIF_EXAM,AU
DIT_CALIF_TOTA,AUDIT_ESTAD_REGI,AUDIT_PORCE_ASIS,AUDIT_
OBSERVACIO)
    select 'Insert', getdate(), 'Insercion en TBMC_CALIF_PARC',
SYSTEM_USER, host_name(),APP_NAME(), ins.capa_estud_subu,
ins.capa_numer_cred, ins.capa_asignatura, ins.capa_tipo_cali,
ins.capa_calif_trab, ins.capa_fecha_entr, ins.capa_calif_exam,
ins.capa_calif_tota, ins.capa_estad_regi, ins.capa_porce_asis,
ins.capa_observacio from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMC_CALIF_PROM
ON TBMC_CALIF_PROM
```

```
FOR INSERT AS
begin
    Insert into AUDIT_CALIF_PROM
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_ASIGNATURA,AUDIT_NUMER_CRED,AUDIT_PROME_
CALI,AUDIT_PROME_ASIS,AUDIT_OBSERVACIO)
    select 'Insert', getdate(), 'Insercion en TBMC_CALIF_PROM',
SYSTEM_USER, host_name(),APP_NAME(), ins.capr_estud_subu,
ins.capr_asignatura, ins.capr_numer_cred, ins.capr_prome_cali,
ins.capr_prome_asis, ins.capr_observacio from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_CUOTA
ON TBMM_CUOTA
FOR INSERT AS
begin
    Insert into AUDIT_CUOTA
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_NÚMERO,AUDIT_VALOR,AUDI
T_FECHA_VENC,AUDIT_ESTADO,AUDIT_PORCE_CUOT,AUDIT_INTE
R_MENS)
    select 'Insert', getdate(), 'Insercion en TBMM_CUOTA',
SYSTEM_USER, host_name(),APP_NAME(), ins.cuot_estud_subu,
ins.cuot_numer_matr, ins.cuot_Número, ins.cuot_valor,
ins.cuot_fecha_venc, ins.cuot_estado, ins.cuot_porce_cuot,
ins.cuot_inter_mens from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_DETAL_MATR
ON TBMM_DETAL_MATR
FOR INSERT AS
begin
    Insert into AUDIT_DETAL_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_TIPO,AUDIT_MOTIVO,AUDIT_M
ONTO,AUDIT_VALOR_CALC)
    select 'Insert', getdate(), 'Insercion en TBMM_DETAL_MATR',
SYSTEM_USER, host_name(),APP_NAME(), ins.dema_estud_subu,
```

```
ins.dema_numer_matr, ins.dema_tipo, ins.dema_motivo,
ins.dema_monto, ins.dema_valor_calc from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_INCRE_MATR ON
TBMM_INCRE_MATR
FOR INSERT AS
begin
    Insert into AUDIT_INCRE_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_TIPO_
MATR,AUDIT_SUBUNIDAD,AUDIT_DESCRIPCIO,AUDIT_PORCE_INC
R)
    select 'Insert', getdate(), 'Insercion en TBMM_INCRE_MATR',
SYSTEM_USER, host_name(),APP_NAME(), ins.inma_tipo_matr,
ins.inma_subunidad, ins.inma_descripcio, ins.inma_porce_incr from
inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_MATRI_ASIG
ON TBMM_MATRI_ASIG
FOR INSERT AS
begin
    Insert into AUDIT_MATRI_ASIG
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD,AUDIT_LIMITE)
    select 'Insert', getdate(), 'Insercion en TBMM_MATRI_ASIG',
SYSTEM_USER, host_name(),APP_NAME(), ins.maas_categoria,
ins.maas_valor, ins.maas_subunidad, ins.maas_limite from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_MATRI_GLOB
ON TBMM_MATRI_GLOB
FOR INSERT AS
begin
    Insert into AUDIT_MATRI_GLOB
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD)
```

```
    select 'Insert', getdate(), 'Insercion en TBMM_MATRI_GLOB',
SYSTEM_USER, host_name(),APP_NAME(), ins.magl_categoria,
ins.magl_valor, ins.magl_subunidad from inserted ins
end
go

-- Creacion del trigger para INSERCION
CREATE TRIGGER AUDITins_TBMM_ORDEN_PAGO
ON TBMM_ORDEN_PAGO
FOR INSERT AS
begin
    Insert into AUDIT_ORDEN_PAGO
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_FECHA_EMIS,AUDIT_CONCE_
COBR,AUDIT_BANCO,AUDIT_NUMER_RECI,AUDIT_OBSERVACIO,AU
DIT_VALOR_TOTA)
    select 'Insert', getdate(), 'Insercion en TBMM_ORDEN_PAGO',
SYSTEM_USER, host_name(),APP_NAME(), ins.orpa_estud_subu,
ins.orpa_numer_matr, ins.orpa_fecha_emis, ins.orpa_conce_cobr,
ins.orpa_banco, ins.orpa_numer_reci, ins.orpa_observacio,
ins.orpa_valor_tota from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMC_CALIF_PARC
ON TBMC_CALIF_PARC
FOR UPDATE AS
begin
    Insert into AUDIT_CALIF_PARC
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_CRED,AUDIT_ASIGNATURA,AUDIT_TIPO_CA
LI,AUDIT_CALIF_TRAB,AUDIT_FECHA_ENTR,AUDIT_CALIF_EXAM,AU
DIT_CALIF_TOTA,AUDIT_ESTAD_REGI,AUDIT_PORCE_ASIS,AUDIT_
OBSERVACIO)
    select 'Update', getdate(), 'Modificacion en TBMC_CALIF_PARC:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.capa_estud_subu, del.capa_numer_cred, del.capa_asignatura,
del.capa_tipo_cali, del.capa_calif_trab, del.capa_fecha_entr,
del.capa_calif_exam, del.capa_calif_tota, del.capa_estad_regi,
del.capa_porce_asis, del.capa_observacio from deleted del
    Insert into AUDIT_CALIF_PARC
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
```

```
_SUBU,AUDIT_NUMER_CRED,AUDIT_ASIGNATURA,AUDIT_TIPO_CA
LI,AUDIT_CALIF_TRAB,AUDIT_FECHA_ENTR,AUDIT_CALIF_EXAM,AU
DIT_CALIF_TOTA,AUDIT_ESTAD_REGI,AUDIT_PORCE_ASIS,AUDIT_
OBSERVACIO)
    select 'Update', getdate(), 'Modificacion en TBMC_CALIF_PARC:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.capa_estud_subu, ins.capa_numer_cred, ins.capa_asignatura,
ins.capa_tipo_cali, ins.capa_calif_trab, ins.capa_fecha_entr,
ins.capa_calif_exam, ins.capa_calif_tota, ins.capa_estad_regi,
ins.capa_porce_asis, ins.capa_observacio from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMC_CALIF_PROM
ON TBMC_CALIF_PROM
FOR UPDATE AS
begin
    Insert into AUDIT_CALIF_PROM
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_ASIGNATURA,AUDIT_NUMER_CRED,AUDIT_PROME_
CALI,AUDIT_PROME_ASIS,AUDIT_OBSERVACIO)
    select 'Update', getdate(), 'Modificacion en TBMC_CALIF_PROM:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.capr_estud_subu, del.capr_asignatura, del.capr_numer_cred,
del.capr_prome_cali, del.capr_prome_asis, del.capr_observacio from
deleted del
    Insert into AUDIT_CALIF_PROM
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_ASIGNATURA,AUDIT_NUMER_CRED,AUDIT_PROME_
CALI,AUDIT_PROME_ASIS,AUDIT_OBSERVACIO)
    select 'Update', getdate(), 'Modificacion en TBMC_CALIF_PROM:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.capr_estud_subu, ins.capr_asignatura, ins.capr_numer_cred,
ins.capr_prome_cali, ins.capr_prome_asis, ins.capr_observacio from
inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_CUOTA
ON TBMM_CUOTA
FOR UPDATE AS
```

```
begin
    Insert into AUDIT_CUOTA
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_NÚMERO,AUDIT_VALOR,AUDI
T_FECHA_VENC,AUDIT_ESTADO,AUDIT_PORCE_CUOT,AUDIT_INTE
R_MENS)
    select 'Update', getdate(), 'Modificacion en TBMM_CUOTA: Original' ,
SYSTEM_USER, host_name(),APP_NAME(), del.cuot_estud_subu,
del.cuot_numer_matr, del.cuot_Número, del.cuot_valor,
del.cuot_fecha_venc, del.cuot_estado, del.cuot_porce_cuot,
del.cuot_inter_mens from deleted del
    Insert into AUDIT_CUOTA
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_NÚMERO,AUDIT_VALOR,AUDI
T_FECHA_VENC,AUDIT_ESTADO,AUDIT_PORCE_CUOT,AUDIT_INTE
R_MENS)
    select 'Update', getdate(), 'Modificacion en TBMM_CUOTA:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.cuot_estud_subu, ins.cuot_numer_matr, ins.cuot_Número,
ins.cuot_valor, ins.cuot_fecha_venc, ins.cuot_estado,
ins.cuot_porce_cuot, ins.cuot_inter_mens from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_DETAL_MATR
ON TBMM_DETAL_MATR
FOR UPDATE AS
begin
    Insert into AUDIT_DETAL_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_TIPO,AUDIT_MOTIVO,AUDIT_M
ONTO,AUDIT_VALOR_CALC)
    select 'Update', getdate(), 'Modificacion en TBMM_DETAL_MATR:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.dema_estud_subu, del.dema_numer_matr, del.dema_tipo,
del.dema_motivo, del.dema_monto, del.dema_valor_calc from deleted del
    Insert into AUDIT_DETAL_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_TIPO,AUDIT_MOTIVO,AUDIT_M
ONTO,AUDIT_VALOR_CALC)
    select 'Update', getdate(), 'Modificacion en TBMM_DETAL_MATR:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
```

```
ins.dema_estud_subu, ins.dema_numer_matr, ins.dema_tipo,
ins.dema_motivo, ins.dema_monto, ins.dema_valor_calc from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_INCRE_MATR ON
TBMM_INCRE_MATR
FOR UPDATE AS
begin
    Insert into AUDIT_INCRE_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_TIPO_
MATR,AUDIT_SUBUNIDAD,AUDIT_DESCRIPCIO,AUDIT_PORCE_INC
R)
    select 'Update', getdate(), 'Modificacion en TBMM_INCRE_MATR:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.inma_tipo_matr, del.inma_subunidad, del.inma_descripcio,
del.inma_porce_incr from deleted del
    Insert into AUDIT_INCRE_MATR
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_TIPO_
MATR,AUDIT_SUBUNIDAD,AUDIT_DESCRIPCIO,AUDIT_PORCE_INC
R)
    select 'Update', getdate(), 'Modificacion en TBMM_INCRE_MATR:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.inma_tipo_matr, ins.inma_subunidad, ins.inma_descripcio,
ins.inma_porce_incr from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_MATRI_ASIG
ON TBMM_MATRI_ASIG
FOR UPDATE AS
begin
    Insert into AUDIT_MATRI_ASIG
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD,AUDIT_LIMITE)
    select 'Update', getdate(), 'Modificacion en TBMM_MATRI_ASIG:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.maas_categoria, del.maas_valor, del.maas_subunidad,
del.maas_limite from deleted del
```

```
    Insert into AUDIT_MATRI_ASIG
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD,AUDIT_LIMITE)
    select 'Update', getdate(), 'Modificacion en TBMM_MATRI_ASIG:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.maas_categoria, ins.maas_valor, ins.maas_subunidad,
ins.maas_limite from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_MATRI_GLOB
ON TBMM_MATRI_GLOB
FOR UPDATE AS
begin
    Insert into AUDIT_MATRI_GLOB
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD)
    select 'Update', getdate(), 'Modificacion en TBMM_MATRI_GLOB:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.magl_categoria, del.magl_valor, del.magl_subunidad from deleted del
    Insert into AUDIT_MATRI_GLOB
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_CATEG
ORIA,AUDIT_VALOR,AUDIT_SUBUNIDAD)
    select 'Update', getdate(), 'Modificacion en TBMM_MATRI_GLOB:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.magl_categoria, ins.magl_valor, ins.magl_subunidad from inserted ins
end
go

-- Creacion del trigger para MODIFICACION
CREATE TRIGGER AUDITupd_TBMM_ORDEN_PAGO
ON TBMM_ORDEN_PAGO
FOR UPDATE AS
begin
    Insert into AUDIT_ORDEN_PAGO
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_FECHA_EMIS,AUDIT_CONCE_
COBR,AUDIT_BANCO,AUDIT_NUMER_RECI,AUDIT_OBSERVACIO,AU
DIT_VALOR_TOTA)
    select 'Update', getdate(), 'Modificacion en TBMM_ORDEN_PAGO:
Original' , SYSTEM_USER, host_name(),APP_NAME(),
del.orpa_estud_subu, del.orpa_numer_matr, del.orpa_fecha_emis,
```

```
del.orpa_conce_cobr, del.orpa_banco, del.orpa_numer_reci,
del.orpa_observacio, del.orpa_valor_tota from deleted del
    Insert into AUDIT_ORDEN_PAGO
(tipo_evento,fecha,descripcion,usuario,terminal,aplicacion,AUDIT_ESTUD
_SUBU,AUDIT_NUMER_MATR,AUDIT_FECHA_EMIS,AUDIT_CONCE_
COBR,AUDIT_BANCO,AUDIT_NUMER_RECI,AUDIT_OBSERVACIO,AU
DIT_VALOR_TOTA)
    select 'Update', getdate(), 'Modificacion en TBMM_ORDEN_PAGO:
Modificado' , SYSTEM_USER, host_name(),APP_NAME(),
ins.orpa_estud_subu, ins.orpa_numer_matr, ins.orpa_fecha_emis,
ins.orpa_conce_cobr, ins.orpa_banco, ins.orpa_numer_reci,
ins.orpa_observacio, ins.orpa_valor_tota from inserted ins
end
go
```

CAPITULO 3

3. REQUERIMIENTOS TECNICOS DEL SISTEMA

3.1. Requerimientos de Hardware

El Hardware para este sistema no necesita tener alguna particular en común, o específicamente ser de una marca, proveedor o tecnología. Las recomendaciones de las características que debería tener la máquina que actuaría como servidor sería el siguiente:

- Un procesador preferiblemente Pentium mayor de 3 GHz.
- Memoria Ram de 512 MB.
- Un Disco Duro con un mínimo de 5 GB disponible.

Las recomendaciones mínimas para las máquinas clientes deberían ser las siguientes:

- Un procesador preferiblemente Pentium de 1.7 GHz.
- Memoria Ram de 128 MB.
- Un Disco Duro con un mínimo de 500 MB disponibles.

3.2. Requerimientos de Software

Como hemos mencionado anteriormente, este sistema está desarrollado sobre una plataforma Windows tanto en el servidor como en el cliente, por esta razón son necesarios los siguientes requerimientos:

- Sistema Operativo Windows 2000 Server o versiones siguientes.
- SQL Server 2000.
- Librerías de Visual Studio.Net para el funcionamiento del sistema.

3.3. Requerimientos Generales

Antes de la Ejecución del Sistema Académico se tienen que seguir los siguientes pasos:

- Revisar el archivo Base_Acad.ini en cada máquina cliente, que contiene el nombre de la máquina servidor y la ruta de la base de datos.
- Comprobar la existencia del archivo PrjConexion.dll.
- Ajustar la configuración regional (Menú Inicio → Configuración → Panel de Control) de cada computador de los usuarios de la siguiente manera:

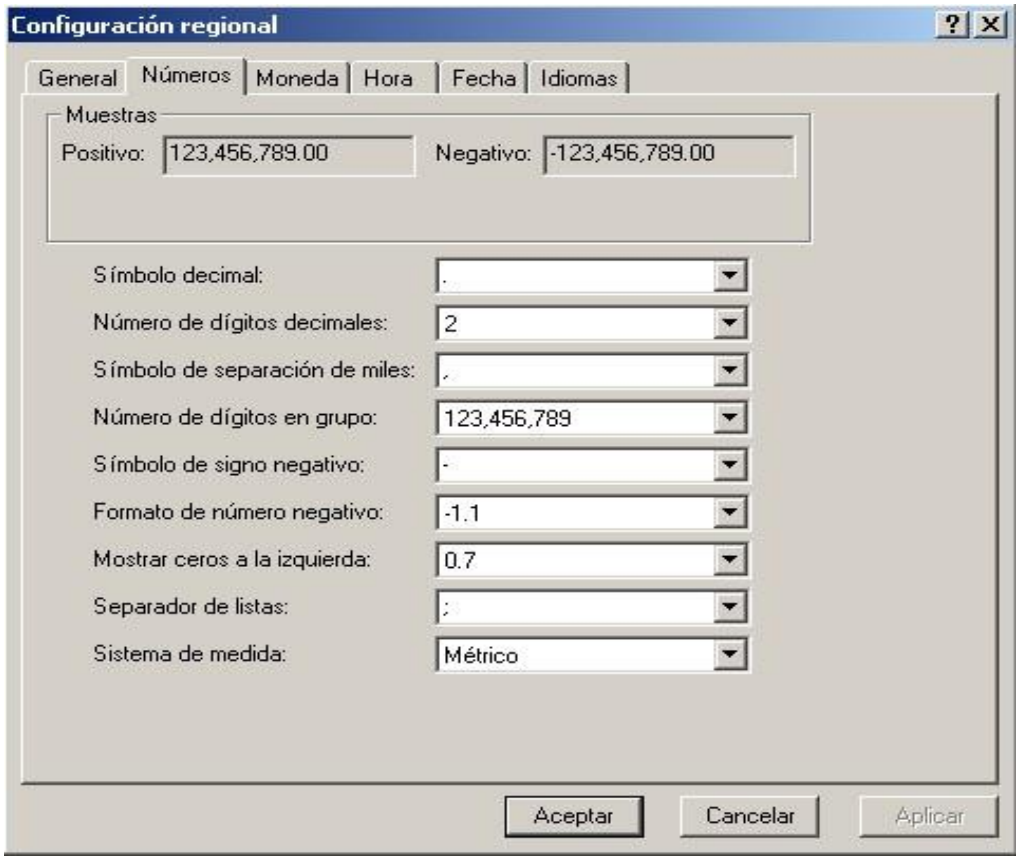


Figura 3.1. Configuración Regional- Números

En el campo de símbolo decimal se escribe sólo un punto (.), en el número de dígitos decimales se ingresa 2 y el símbolo de separador de miles se escribe una coma (,).

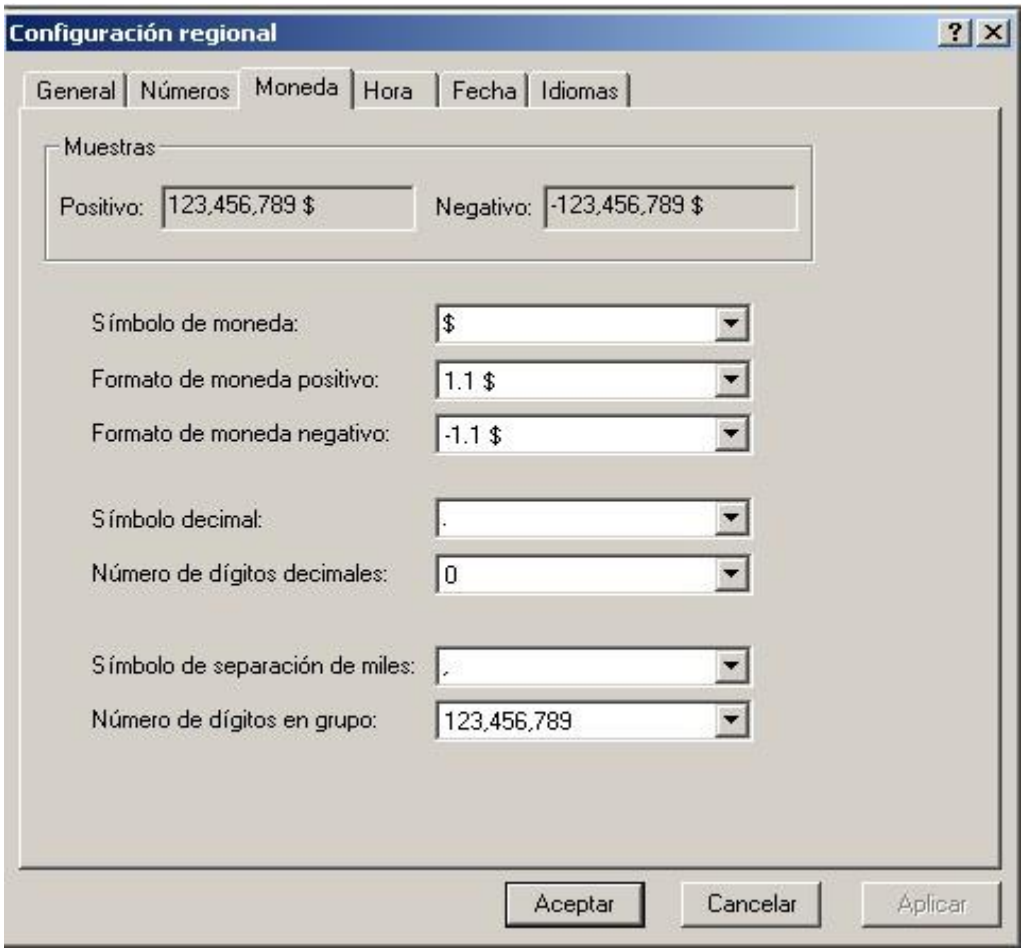


Figura 3.2. Configuración Regional- Moneda

En el campo del símbolo de moneda se elige el símbolo de dólar ($), en el símbolo decimal se elige el punto (.) y en el símbolo de separador de miles se elige la coma (,).

Figura 3.3. Configuración Regional- Fecha

Para la configuración de la fecha, se digita el formato de fecha corta de la siguiente manera: aaaa/MM/dd, tomando en cuenta que el separador de fecha sea: /.

- En la base de datos SQL Server, crear los usuarios del sistema antes de que ellos ingresen al sistema. Una vez creados ingresarlos en la tabla de usuarios del sistema con el código de la subunidad a la que pertenece. Cabe recalcar que el primer usuario a crear e ingresar es

uno del tipo ADM_SYS (administrador del sistema) y no debe pertenecer a ninguna subunidad académica.

- Al ingresar por primera vez al sistema, entrar como ADM_SYS y almacenar la información más importante respecto al sistema. Cabe destacar que este usuario no pertenece a ninguna subunidad, por lo tanto, los parámetros que puede ingresar son generales.

CAPITULO 4

4. INSTALACION DE COMPONENTES

4.1. Instalación de SQL Server en el Servidor

Hay tres formas de instalar SQL Server. Se puede realizar instalación local, remota o desatendida. Una instalación local instala SQL Server en el equipo que se está utilizando actualmente. Mediante una instalación remota se puede instalar SQL Server en otro equipo que se encuentre en la red. Una instalación desatendida permite instalar SQL Server sin tener que estar presente para responder a cualquier petición de datos. En vez de eso, se almacenan por adelantado todas las respuestas a las peticiones de datos en un archivo, y el programa de instalación las lee automáticamente cuando las necesita.

Si es la primera vez que se instala SQL Server 2000, conviene realizar una instalación local antes de intentar cualquiera de las otras dos opciones. De esta forma nos podemos familiarizar con el procedimiento

general de instalación.

Para realizar una instalación local hay que seguir estos simples pasos para tener SQL Server instalado y funcionando en el servidor:

1. Introducir el CD de SQL Server en una unidad de CDs. Si el sistema operativo del servidor está configurado para iniciar automáticamente los CDs, el cuadro de diálogo de la instalación de Microsoft SQL Server 2000 aparecerá, como se muestra en la Figura 4.l. En otro caso, será necesario ejecutar manualmente el programa Autorun.exe (situado en el directorio de más alto nivel del CD).

Figura 4.1. Menú inicial para la instalación de Sql Server

2. Si no se han instalado los services packs necesarios para el sistema operativo o la versión requerida de Microsoft Internet Explorer, o si simplemente se desea examinar la lista de requisitos previos, hay que pulsar en Requisitos previos de SQL Server 2000 para mostrar el cuadro de dialogo Requisitos previos de SQL Server 2000.

   Hay que pulsar en el sistema operativo apropiado para ver sus requisitos previos, y pulsar después en el requisito previo que se desea instalar. Si ya está cargado todo el software apropiado se puede ir al paso 3.

   Nota: Si es necesario instalar Microsoft Internet Explorer o los services packs requeridos por Microsoft Windows 2000 o Microsoft Windows NT 4, es posible que haya que reiniciar el sistema y volver a ejecutar Autorun.exe antes de poder proceder con la instalación de SQL Server.

   Una vez que se hayan instalado los requisitos previos hay que volver al cuadro de diálogo principal de la instalación pulsando Atrás.

3. Pulsar en Componentes de SQL Server 2000 para comenzar la instalación de SQL Server.

4. Aparece el cuadro de diálogo Instalar componentes, como se muestra en la Figura 4.2. Hay que pulsar en Instalar Servidor de bases de datos para que comience la instalación de los componentes principales de SQL Server.

Figura 4.2. Instalar Componentes

5. Aparece el cuadro de diálogo de bienvenida al Asistente para la instalación de SQL Server. Si se están ejecutando otros programas sería conveniente cerrarlos. Hay que pulsar en Siguiente para que continúe el proceso de instalación.

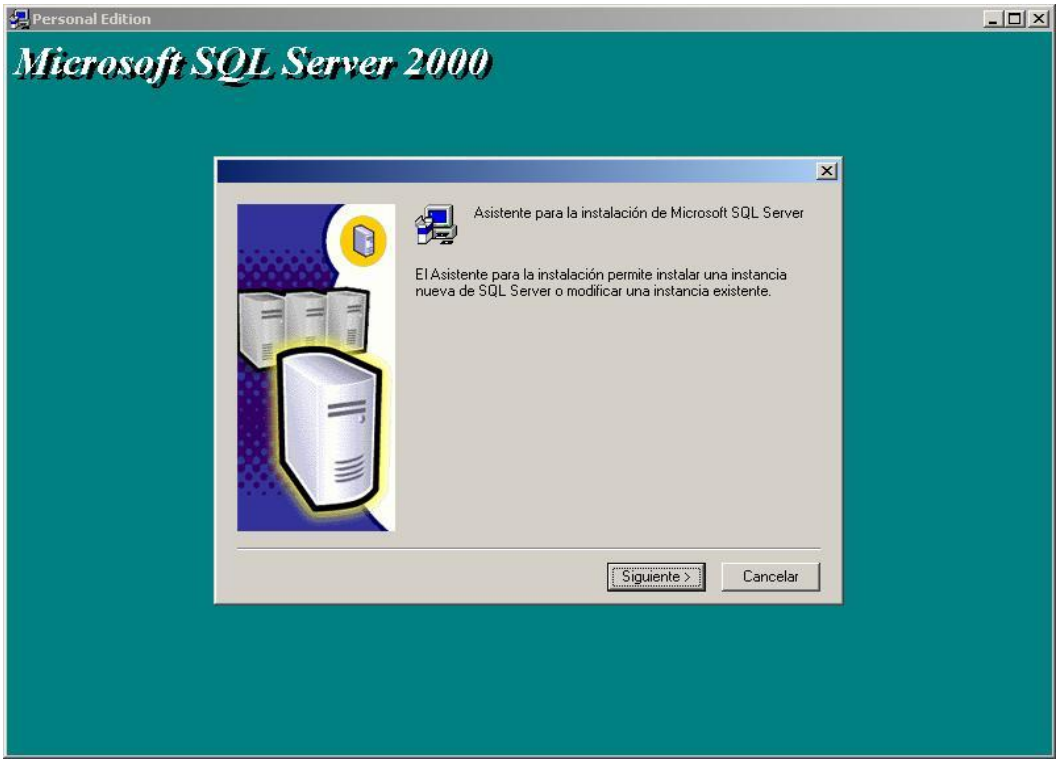


Figura 4.3. Pantalla inicial del instalador

6. Aparece el cuadro de diálogo Nombre del equipo. Hay que pulsar en Equipo local (aparece automáticamente el nombre del equipo) y pulsar después Siguiente.

Figura 4.4. Nombre del equipo

7. Se muestra el cuadro de diálogo Selección de instalación. Hay que pulsar en Crear una nueva instancia de SQL Server y pulsar entonces Siguiente para continuar.

Figura 4.5. Selección de instalación

8. Aparece el cuadro de diálogo Información del usuario. Hay que comprobar que el nombre y la empresa son correctos Pulsar Siguiente para continuar.

Figura 4.6. Información del usuario

9. Se muestra el cuadro de diálogo Contrato de licencia de software. Hay que pulsar Sí para aceptar el contrato de licencia y continuar con el proceso de instalación.

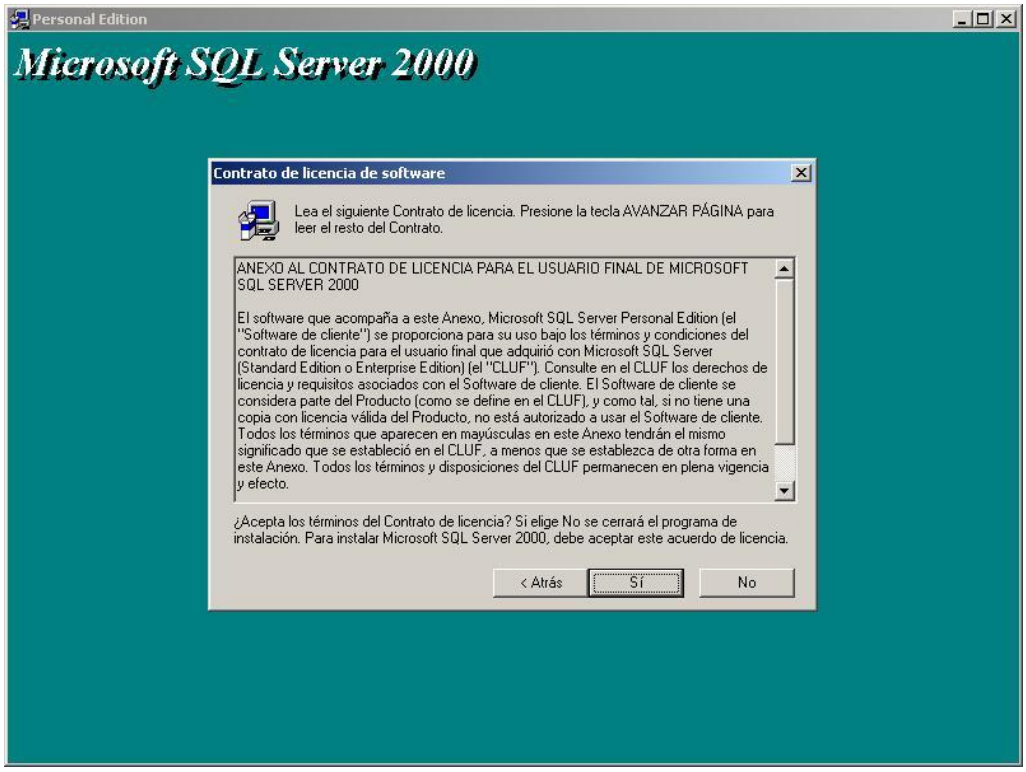


Figura 4.7. Contrato de licencia de software

10. Aparece el cuadro de diálogo Clave del CD. Hay que introducir la clave de CD de 25 caracteres que se encuentra en la etiqueta amarilla de la funda del CD y pulsar Siguiente.

11. Se muestra ahora el cuadro de diálogo Definición de la instalación. Hay que pulsar Herramientas cliente y servidor y después Siguiente.

Figura 4.8. Definición de instalación

12. Aparece el cuadro de diálogo Nombre de instancia. Si se desea dar nombre a esta instancia de SQL Server de forma distinta a la predeterminada, hay que deshabilitar la casilla de verificación Predeterminada y escribir el nombre deseado. Hay que pulsar Siguiente para continuar con el proceso de instalación.

Nota: Recuerde el nombre de la instancia que va a colocar porque de ello dependerá la configuración inicial del sistema.

Figura 4.9. Nombre de la instancia

13. Aparece el cuadro de diálogo Tipo de instalación, como se muestra en la Figura 4.10. Se puede seleccionar instalación Típica, Mínima o Personalizada. La instalación Típica incluye todas las opciones excepto las Herramientas de desarrollo y las opciones de Búsqueda de texto. La instalación Personalizada permite añadir esas opciones además de eliminar algunas no deseadas. La instalación mínima es equivalente a la Típica sin las opciones Herramientas de actualización, Libros en pantalla y Herramientas de administración.

Seleccione instalación típica. También se puede especificar dónde instalar el programa y los archivos de datos de SQL Server pulsando

en los botones Examinar del grupo Carpeta de destino. Hay que pulsar Siguiente para continuar.

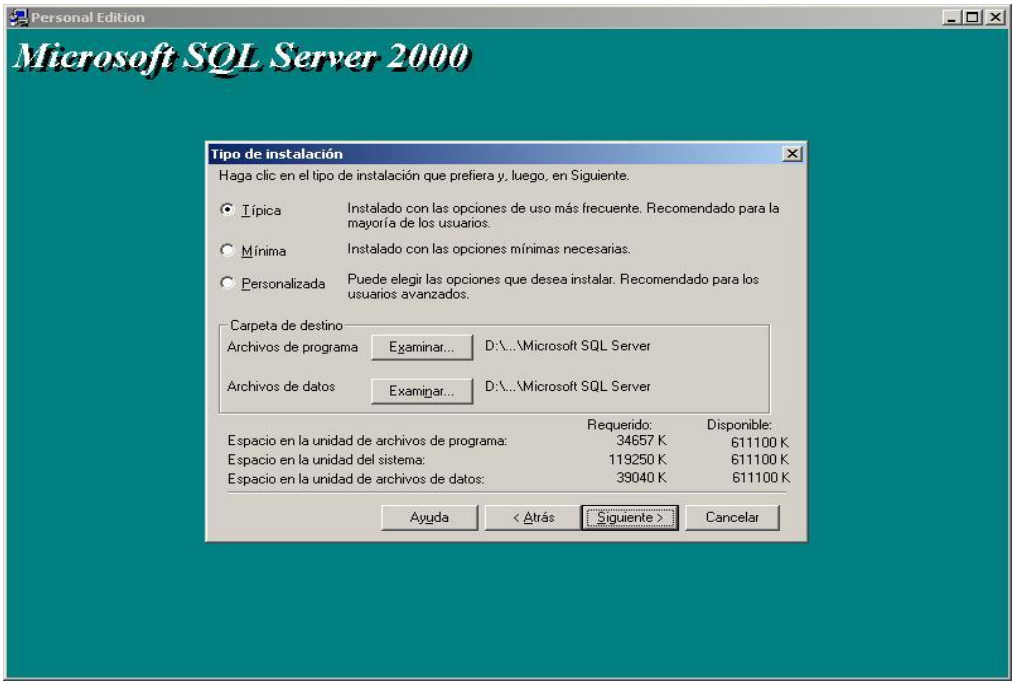


Figura 4.10. Tipo de instalación

14. Aparece el cuadro de diálogo Cuentas de servicios, como se muestra en la Figura 4.11. Se puede utilizar o bien una cuenta de usuario de Windows NT o Windows 2000, o la cuenta Administrador. En cualquier caso, la cuenta debe tener el permiso Inicio de sesión como servicio. Si no se está seguro sobre cómo crear estas cuentas de usuario, conviene consultar al administrador o la documentación de Windows NT o Windows 2000. Hay que escribir el nombre y la contraseña de la cuenta creada para el servicio SQL Server en los cuadros de textos

apropiados. Si se está instalando SQL Server en una estación de trabajo independiente, hay que pulsar Utilizar la cuenta del sistema local. Hay que pulsar Siguiente para continuar. Para este caso, vamos a emplear la cuenta de usuario del dominio “Administrador”.

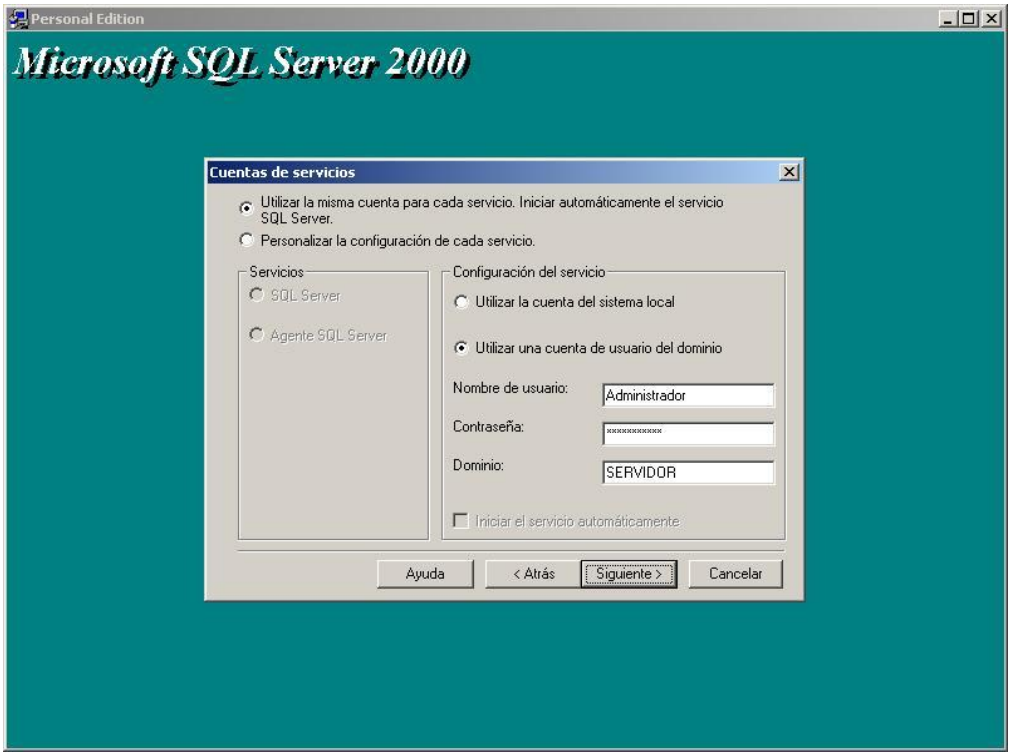


Figura 4.11. Cuentas de servicios

15. A continuación aparece el cuadro de diálogo Modo de autenticación, el cual determina el nivel de seguridad de la instalación de SQL Server. Se puede escoger Modo de autenticación de Windows o Modo mixto. Si se selecciona Modo de autenticación de Windows, todos los permisos de usuario sobre la base de datos se heredarán de la

Configuración de seguridad de Windows. Si se selecciona mixto, habrá que establecer una contraseña para la cuenta sa, o administrador del sistema de SQL Server. Se puede optar por dejar esta contraseña en blanco, pero haciendo eso disminuirá gravemente la seguridad de la instalación de SQL Server. Después de seleccionar un modo de autenticación hay que pulsar Siguiente para continuar.

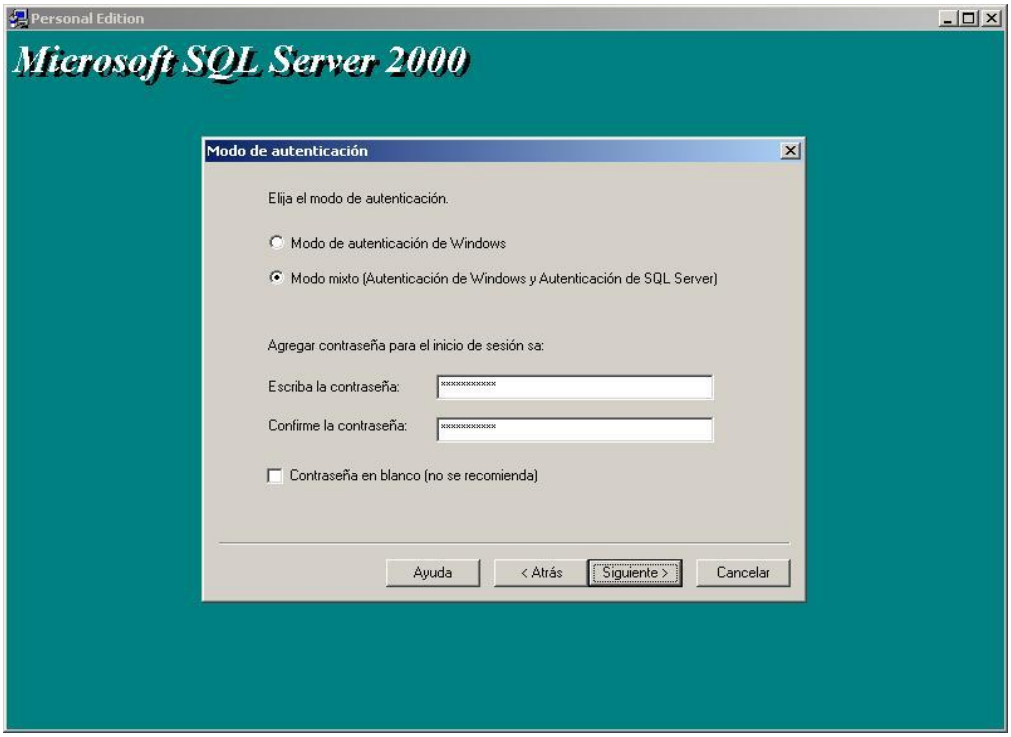


Figura 4.12. Modo de autenticación

Como se muestra en la Figura 4.12, hemos elegido el Modo Mixto de Autenticación SQL y escribimos la contraseña para el usuario sa.

16. Aparece el cuadro de diálogo Iniciar la copia de archivos. Hay que pulsar Siguiente para continuar.

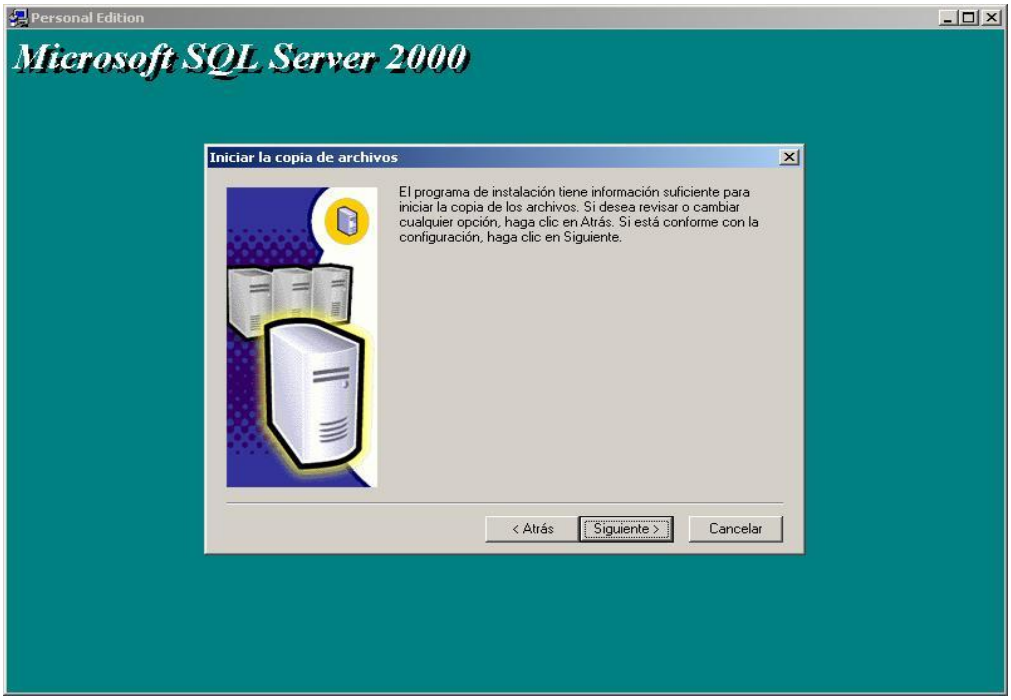


Figura 4.13. Iniciar la copia de archivos

17. Se muestra el cuadro de diálogo de modo de licencia. Hay dos opciones para otorgar licencias a los clientes de SQL Server. Se pueden licenciar clientes por puesto o por procesador. Las licencias por puesto requieren una Licencia de acceso de cliente para cada equipo cliente que accederá al servidor. Una vez que se otorga una licencia a un equipo, puede acceder a cualquier equipo de la red que ejecute SQL Server 2000 sin cargo adicional. Las licencias por procesador requieren una licencia para cada procesador en el que se

ejecutará SQL Server. Por ejemplo, si se ejecuta SQL Server en una máquina con cuatro procesadores, es necesario adquirir cuatro licencias de procesador para utilizar los cuatro procesadores. Se podría optar por limitar SQL Server a tan sólo dos de los cuatro procesadores. En ese caso sólo sería necesario adquirir dos licencias de procesador.

Después de adquirir el número apropiado de licencias de procesador se pueden conectar un número ilimitado de clientes.

Hay que pulsar Continuar para comenzar la instalación de la aplicación y los archivos de datos de SQL Server. SQL Server instalará los archivos requeridos en el sistema y configurará los componentes necesarios. La instalación tardará tan sólo unos pocos minutos, o podría durar más, dependiendo de la velocidad del sistema.

18. Una vez que la instalación está completa, se muestra el cuadro de diálogo Fin de la instalación. Hay que pulsar Finalizar para completar el proceso de instalación.

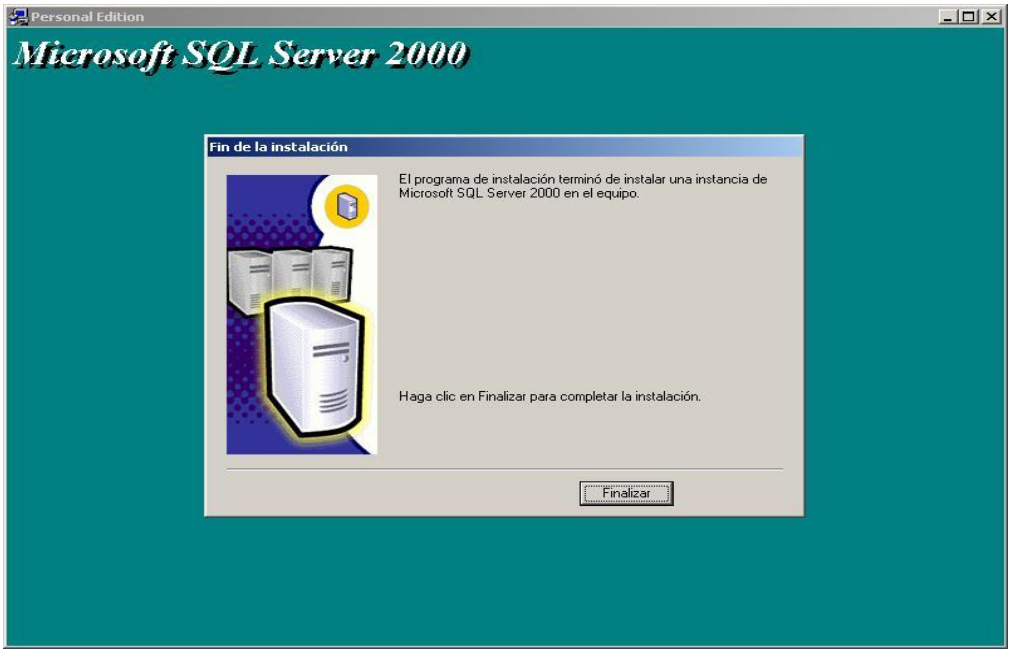


Figura 4.14. Fin de la instalación

19. Damos click en el botón “Inicio” de Windows y buscamos el menú de Microsoft Sql Server para ejecutar el programa “Administrador de Servicios de Sql Server” para verificar que la base esté iniciada.

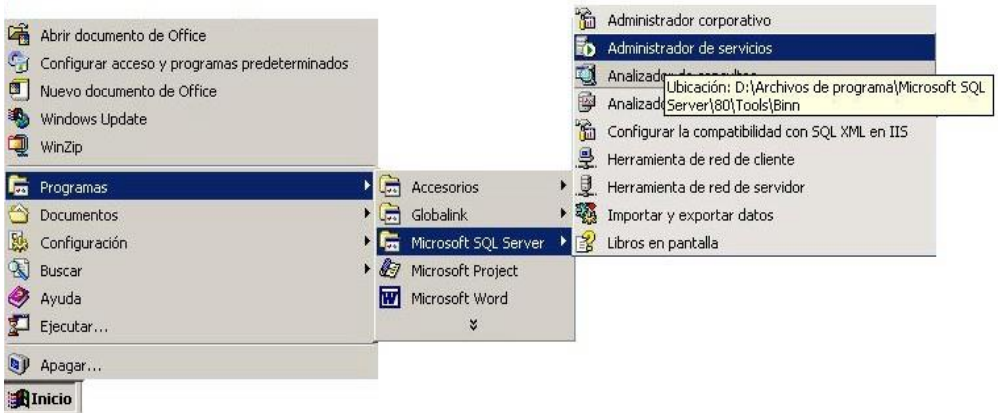


Figura 4.15. Menú de Microsoft Sql Server

20. Si la pantalla aparece tal como en la figura 4.16, damos click en el botón “Iniciar o continuar”. Cabe destacar que nuestro servidor SQL creado aprece con el nombre SERVIDOR\ACADEMICO, porque toma el nombre del equipo donde se ha instalado (SERVIDOR, ver Figura 4.4) y el nombre de la instancia (ACADEMICO, ver Figura 4.9)

Figura 4.16. Administrador de servicios Sql-Detenido:Servidor\Academico

21. Al presionar el botón “Actualizar servicios” la pantalla cambia de apariencia, tal como muestra la figura 4.17.

Figura 4.17. Administrador de servicios Sql-En ejecución:Servidor\Academico

4.2. Registro del servidor SQL

1. Entramos al "SQL Server Enterprise Manager" ("Administrador Corporativo" en el menú de Microsoft Sql) donde se encuentran todos los servidores del SQL y damos click derecho sobre el icono "Grupo de SQL server" ubicado en el panel derecho.

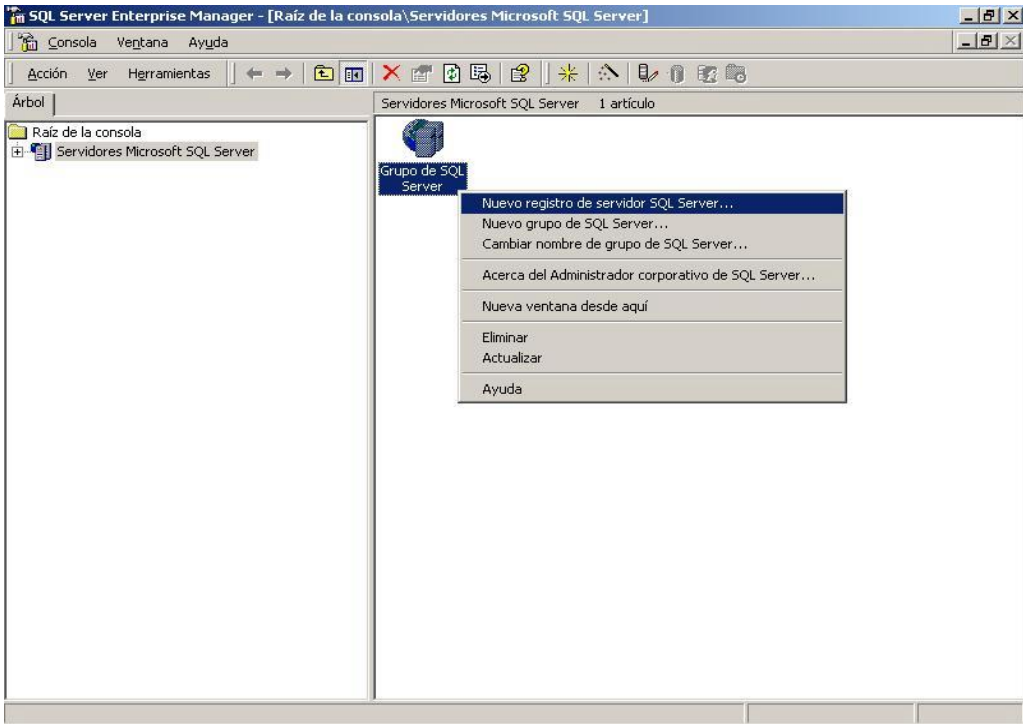


Figura 4.18. SQL Server Enterprise Manager

2. Elegimos “Nuevo registro de servidor SQL Server” y damos click en el botón “Siguiente”.

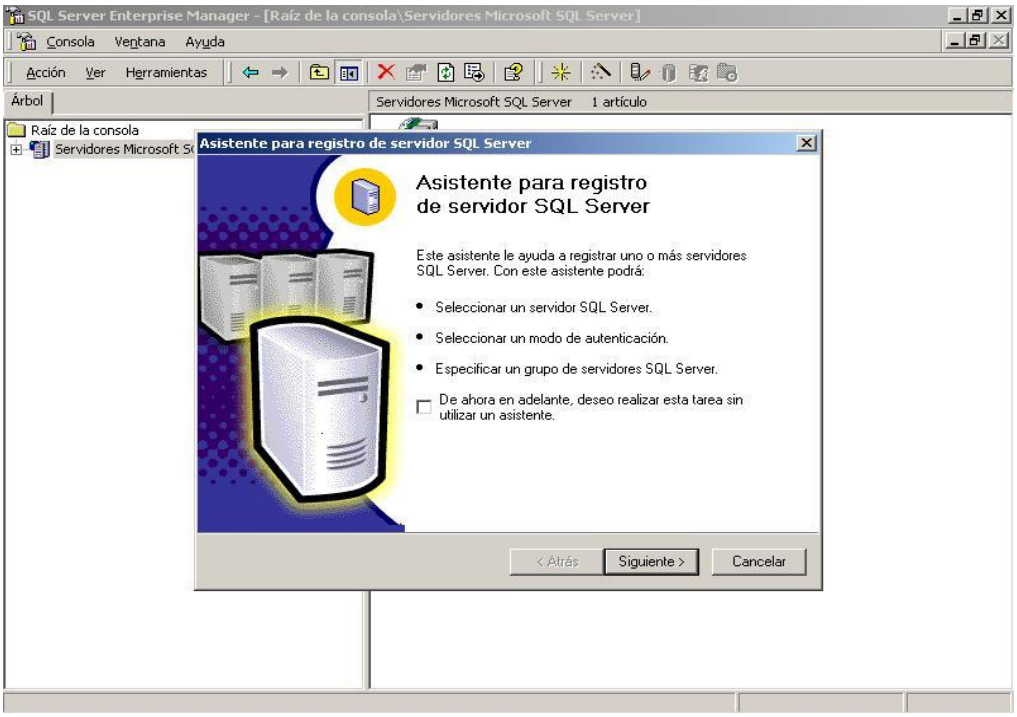


Figura 4.19. Asistente para registro de servidor SQL Server

3. En la caja de texto “Servidores disponibles” tenemos que escribir el nombre del servidor SQL creado (en este caso SERVIDOR\ACADEMICO). Damos click en “Agregar” y presionamos el botón “Siguiente”.

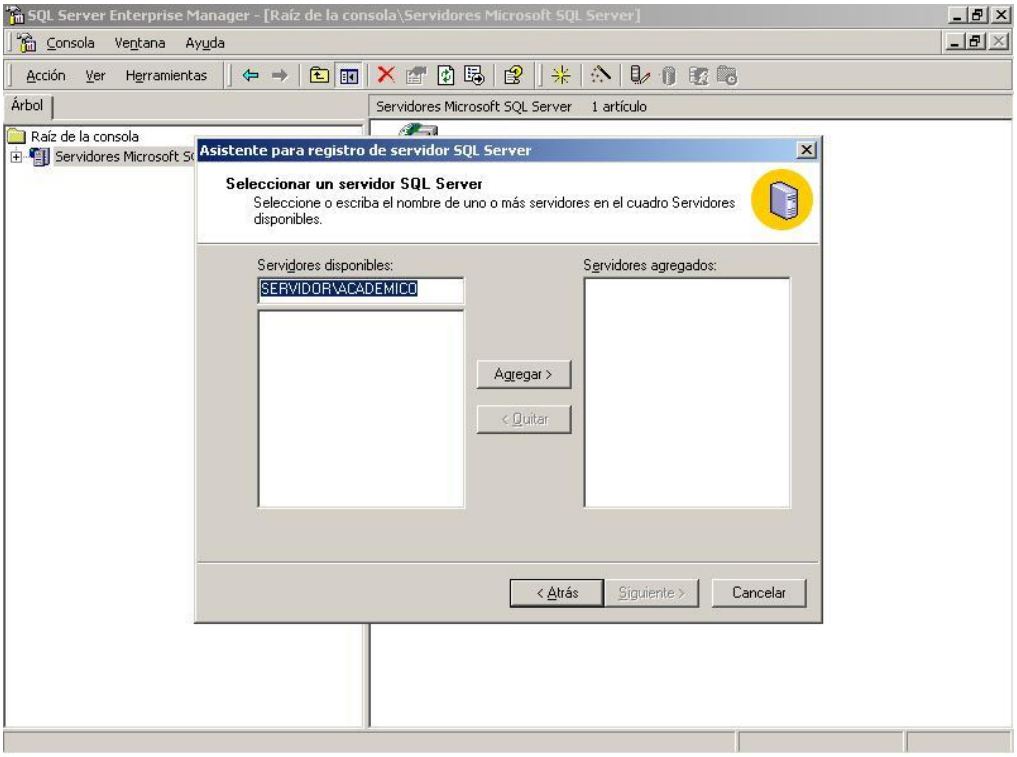


Figura 4.20. Seleccionar un servidor SQL

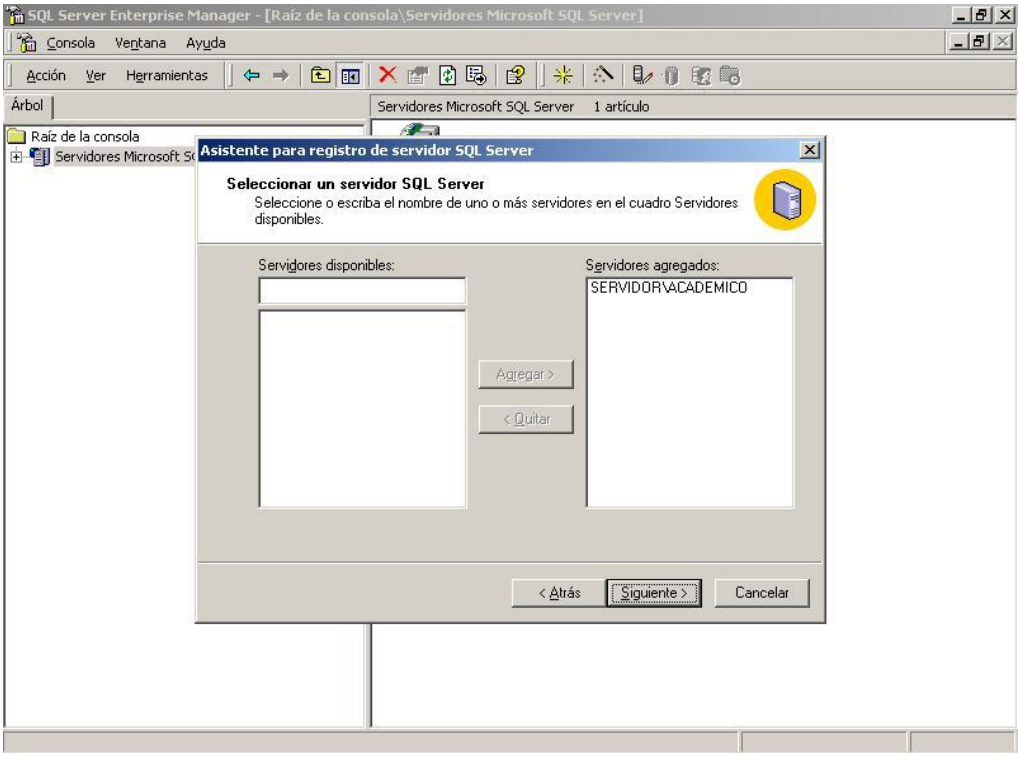


Figura 4.21. Agregar un servidor SQL

4. Respecto al modo de autenticación, elegimos ingresar sólo con el usuario asignado en la instalación y damos click en “Siguiente”.

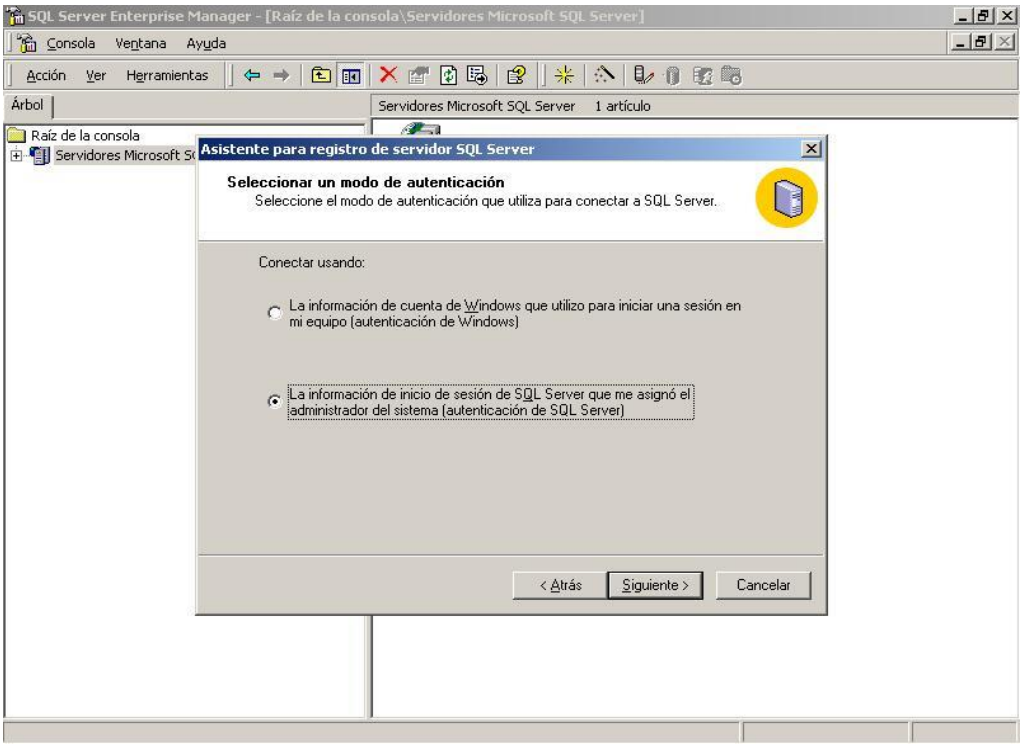


Figura 4.22. Seleccionar un modo de autenticación

5. En el modo de conexión escogemos que Sql pida la conexión al conectar a la base de datos para mayor seguridad.

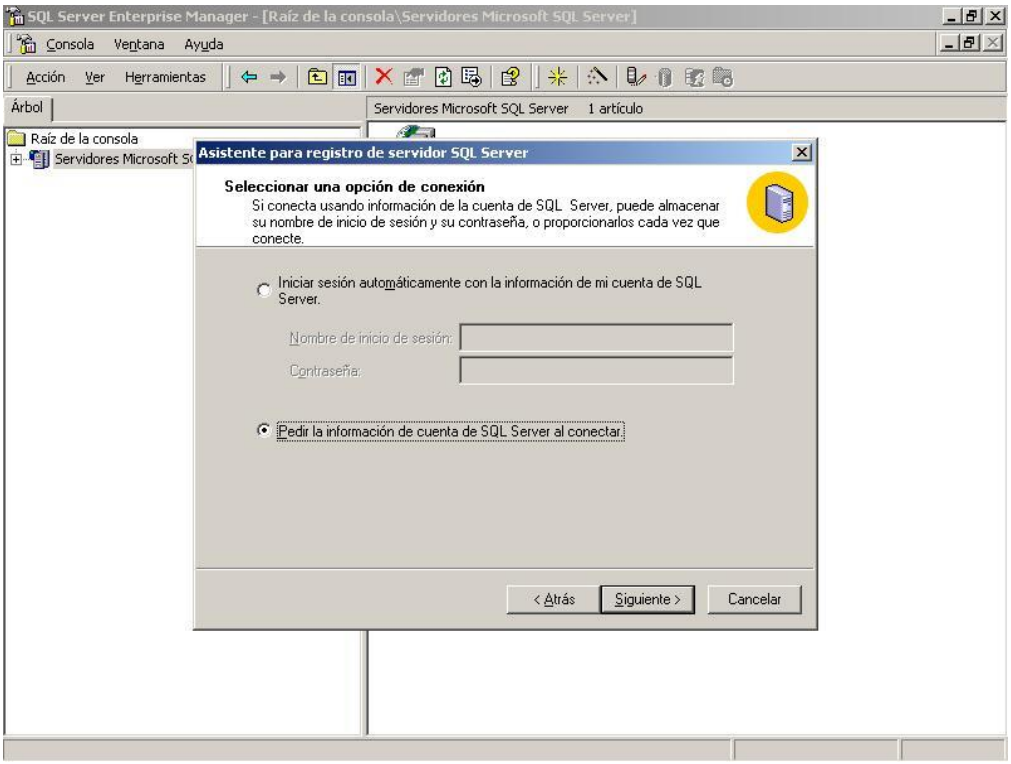


Figura 4.23. Seleccionar una opción de conexión

6. En la pantalla siguiente escogemos agregar nuestra instancia a un nuevo grupo de servidores SQL.

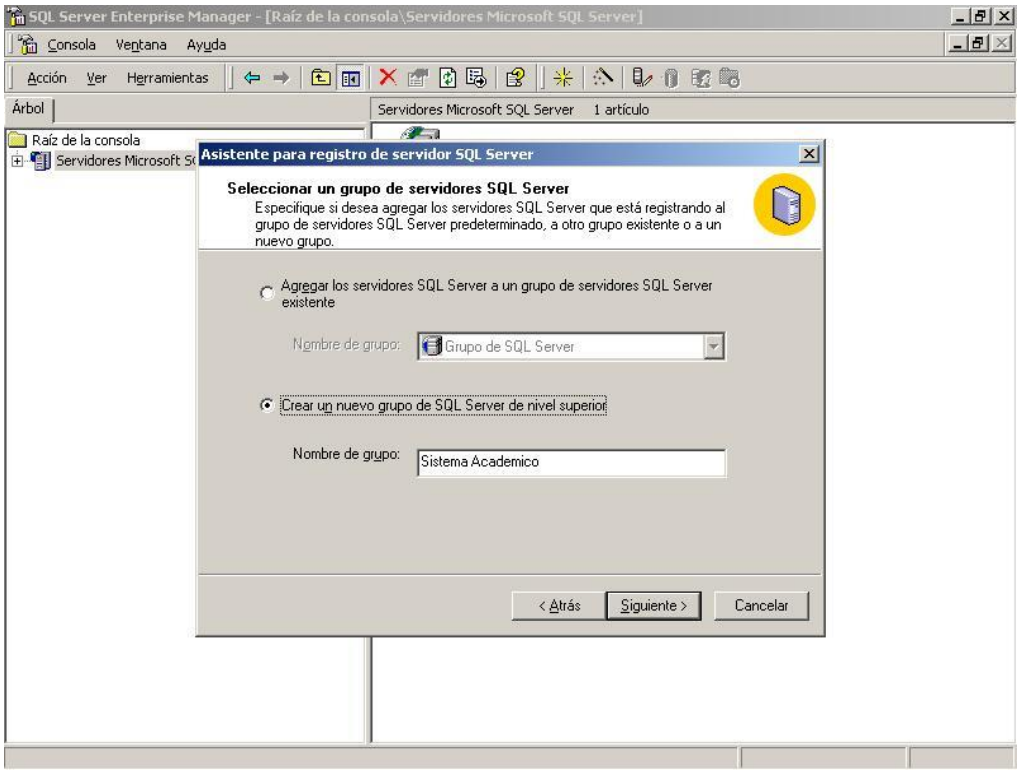


Figura 4.24. Seleccionar un grupo de servidores SQL

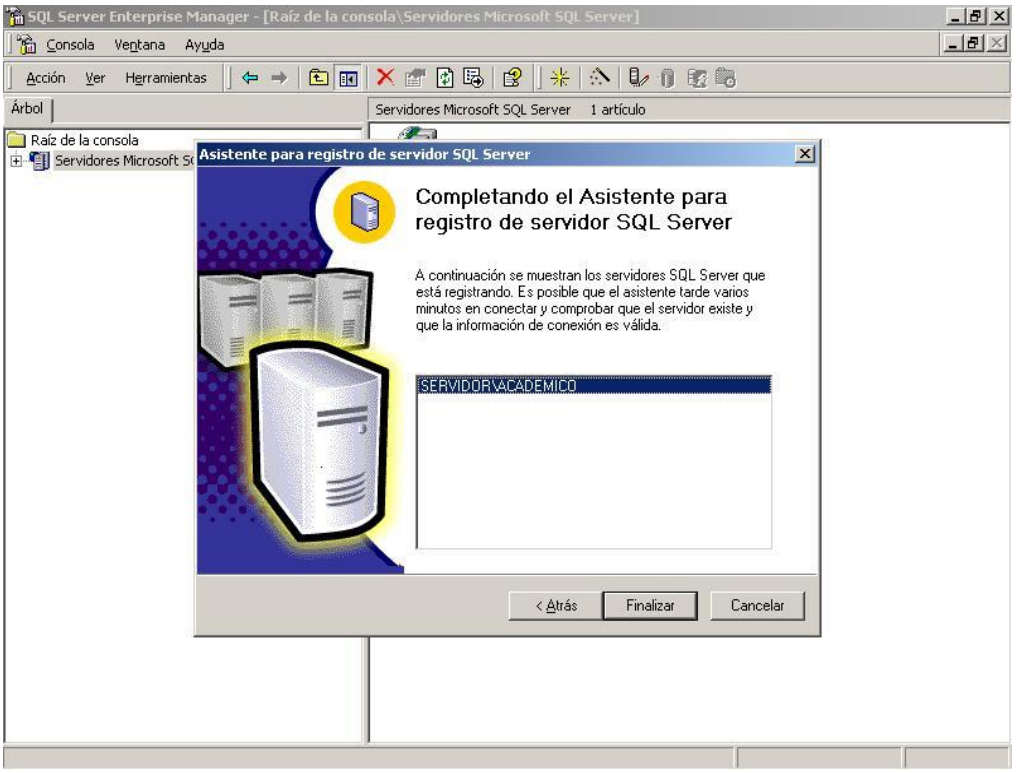


Figura 4.25. Asistente para registro de servidor SQL Finalizado

7. Una vez finalizado el registro del nuevo servidor aparece la pantalla tal como se muestra en la figura 4.26

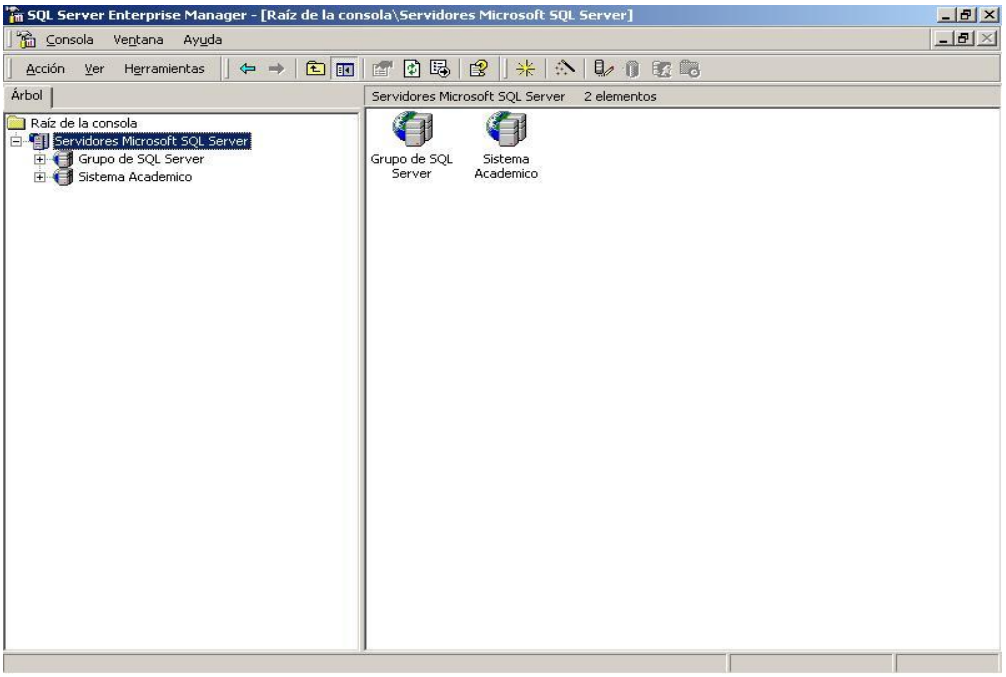


Figura 4.26. Nuevo servidor SQL registrado – Sistema Academico

8. Tratamos de conectarnos a la instancia SERVIDOR\ACADEMICO y en el cuadro de conexión escribimos el usuario “sa” y la contraseña ingresada para este usuario (Figura 4.12), presionamos el botón “Conectar” para crear una base de datos.

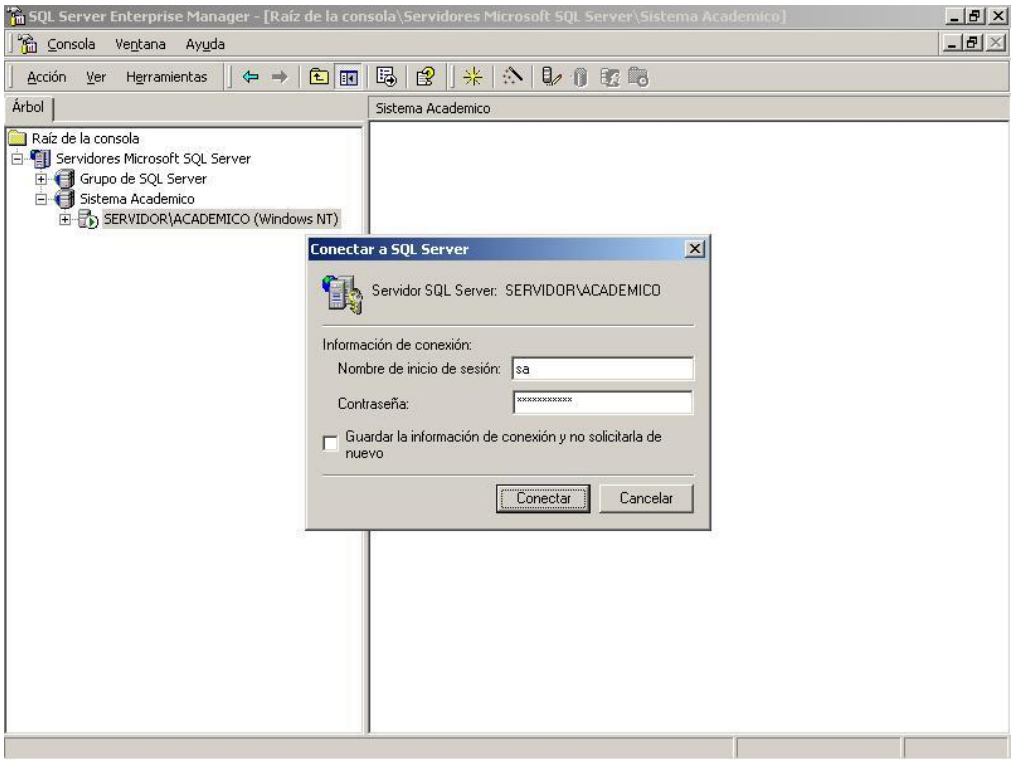


Figura 4.27. Conectar a SQL Server

4.3. Creación de la Base de Datos en SQL Server

SQL Server proporciona tres métodos para la creación de una base de datos: el Asistente para creación de bases de datos, el Administrador corporativo de SQL Server y comandos T-SQL que pueden guardarse en un archivo y ejecutarse como un archivo de comandos:

El Asistente para creación de bases de datos tiene algunas limitaciones que deben tenerse en cuenta. El asistente coloca todos los archivos de datos que crea en una sola unidad de disco en la carpeta que se le especifique. No se puede poner archivos de datos en posiciones físicas

diferentes (ni en unidades diferentes ni en carpetas diferentes) si se utiliza el asistente. Se pueden poner archivos de registro en una unidad o una carpeta separada de los archivos de datos pero, nuevamente, en una única posición física. No pueden especificarse grupos de archivos definidos por el usuario, y todos los archivos heredan las mismas opciones de crecimiento. Debido a estas limitaciones, el Asistente para creación de bases de datos es la mejor elección si sólo se necesita un archivo de datos principal y un archivo de registro de transacciones en la base de datos. (Por otra parte, siempre se pueden añadir archivos y grupos de archivos a la base de datos posteriormente si se necesitan)

Debería usarse el Administrador corporativo o un archivo de comandos T-SQL para crear la base de datos si se tienen archivos de datos secundarios que se quieren colocar en una unidad de disco distinta de la que contenga el archivo de datos principal, si se quieren añadir grupos de archivos definidos por el usuario o si se necesitan opciones de crecimiento distintas para los diferentes archivos.

Para la creación de base de datos que utiliza nuestro sistema, hemos utilizado el Administrador Corporativo. Considere los siguientes pasos:

1. Abrir el Administrador corporativo. En el panel izquierdo, expandir el grupo de SQL Server que contenga el nombre del servidor sobre el

que se quiere crear la base de datos y expandir el propio nodo del servidor. A continuación pulsar con el botón derecho del ratón en la carpeta Bases de datos y elegir Nueva base de datos.

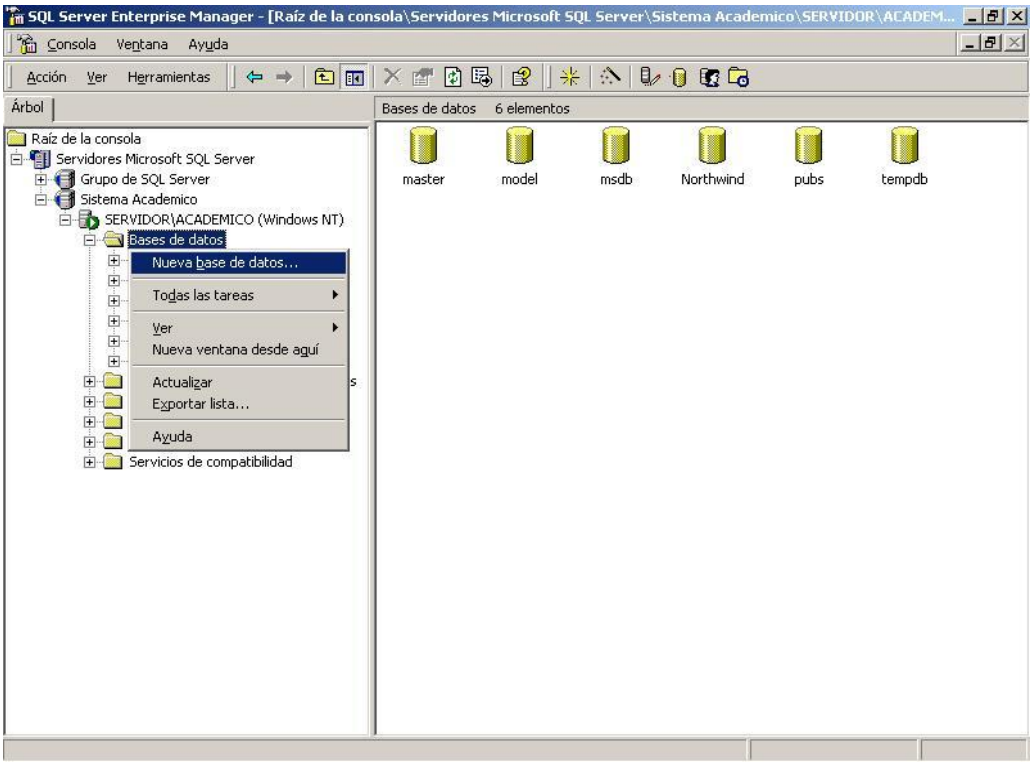


Figura 4.28. Creación de una base de datos

2. Se abre la ventana Propiedades de la base de datos, con la pestaña General en primer plano, como se ve en la figura siguiente. Escribir el nombre de la base de datos en el cuadro Nombre. El nombre de archivo será “Universidad”

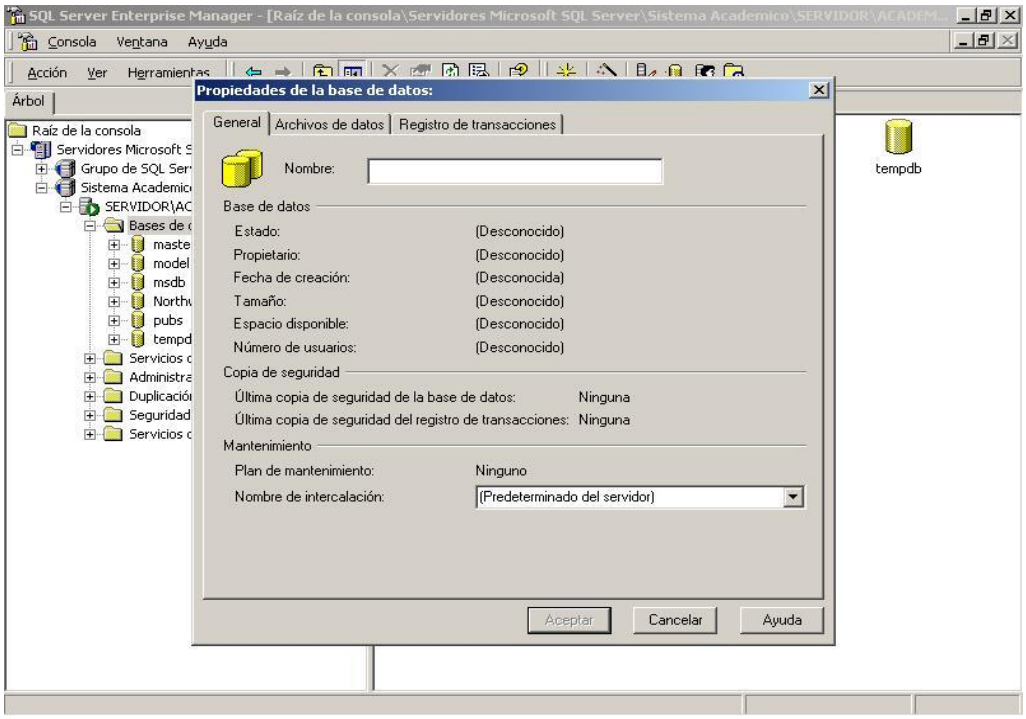


Figura 4.29. Propiedades de la nueva base de datos

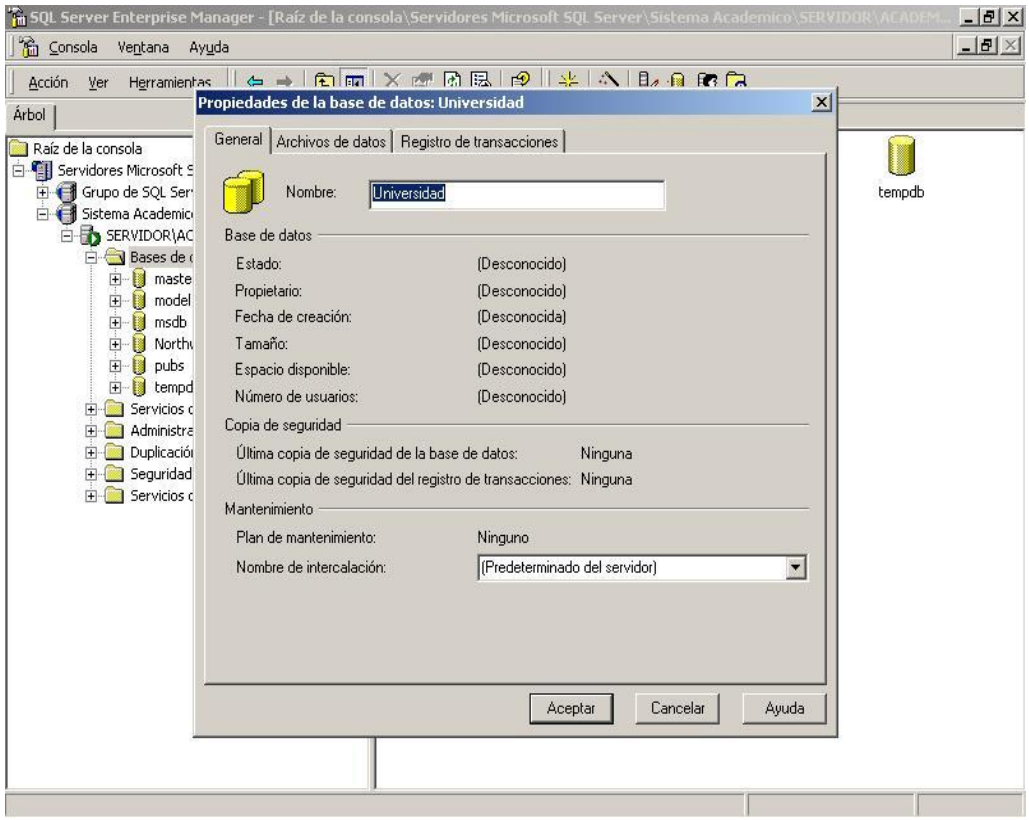


Figura 4.30. Propiedades de la base de datos Universidad–General

3. Pulsar la pestaña Archivos de datos. Como se muestra en la figura siguiente, el Administrador corporativo crea automáticamente el archivo de datos principal, con el nombre de la base de datos como prefijo y PRIMARY como grupo de archivos. Se puede cambiar el nombre, posición y tamaño del archivo principal, pero no se puede cambiar el grupo de archivos del archivo de datos principal. Escribir el nombre de archivo (nombre lógico), ubicación (nombre físico), tamaño y grupo de archivos de cada archivo que se desee crear. Para cada archivo distinto del archivo principal se puede escribir un nombre de

grupo de archivos definido por el usuario, y dicho grupo de archivos será creado.

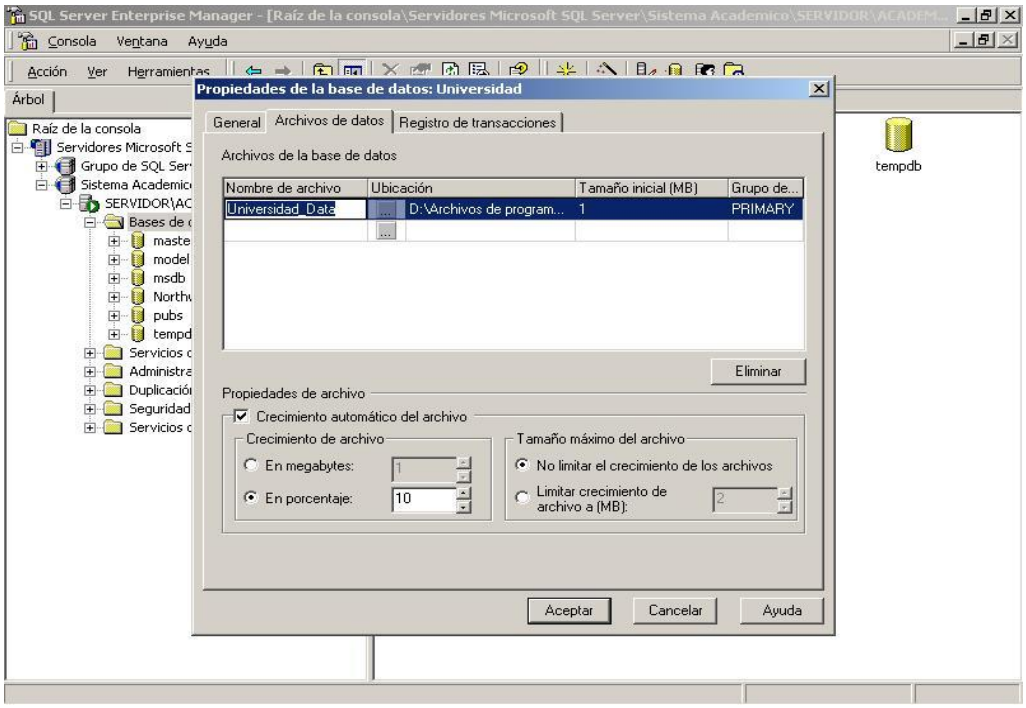


Figura 4.31. Propiedades de la base de datos Universidad–Archivos de datos

La ubicación de cada archivo es de manera predeterminada la carpeta en la unidad en que SQL Server está instalado. Se puede cambiar esta configuración escribiendo una nueva ruta o mediante el uso del botón de examinar.

4. En el área Propiedades de archivo en la parte baja de la ventana se pueden seleccionar las opciones de crecimiento automático según

cada archivo. Seleccionar el nombre del archivo para el que se quiera configurar las opciones de crecimiento. Seleccionar la casilla de verificación Crecimiento automático del archivo para habilitar el crecimiento automático para ese archivo. A continuación se puede especificar el incremento de crecimiento en megabytes o como porcentaje del espacio libre restante en el archivo. También se puede especificar un tamaño máximo de archivo pulsando Limitar crecimiento de archivo a (MB) e introducir un tamaño en megabytes o se puede dejar el tamaño del archivo sin limitar. Se puede configurar estas opciones a medida que se crea cada archivo o se pueden dejar los valores predeterminados y cambiarlos posteriormente mediante la ventana Propiedades de la base de datos de Administrador de archivos. Si se necesita borrar un archivo de la lista, seleccionar el nombre de ese archivo y presionar la tecla Suprimir.

5. Tras terminar la configuración de todos los archivos de datos, pulsar la pestaña Registro de transacciones de la ventana Propiedades de la base de datos para configurar los archivos del registro de transacciones. Los archivos de registro se configuran de la misma manera que los archivos de datos, excepto que no se puede seleccionar un grupo de archivos para los archivos de registro dado que éstos no pertenecen a ningún grupo de archivos. Escribir el nombre de archivo (nombre lógico), la ubicación (nombre físico) y el

tamaño inicial del archivo o los archivos de registro. Hay que elegir también las opciones de crecimiento automático para los archivos de registro como se ha descrito en el paso 4 para archivos de datos.

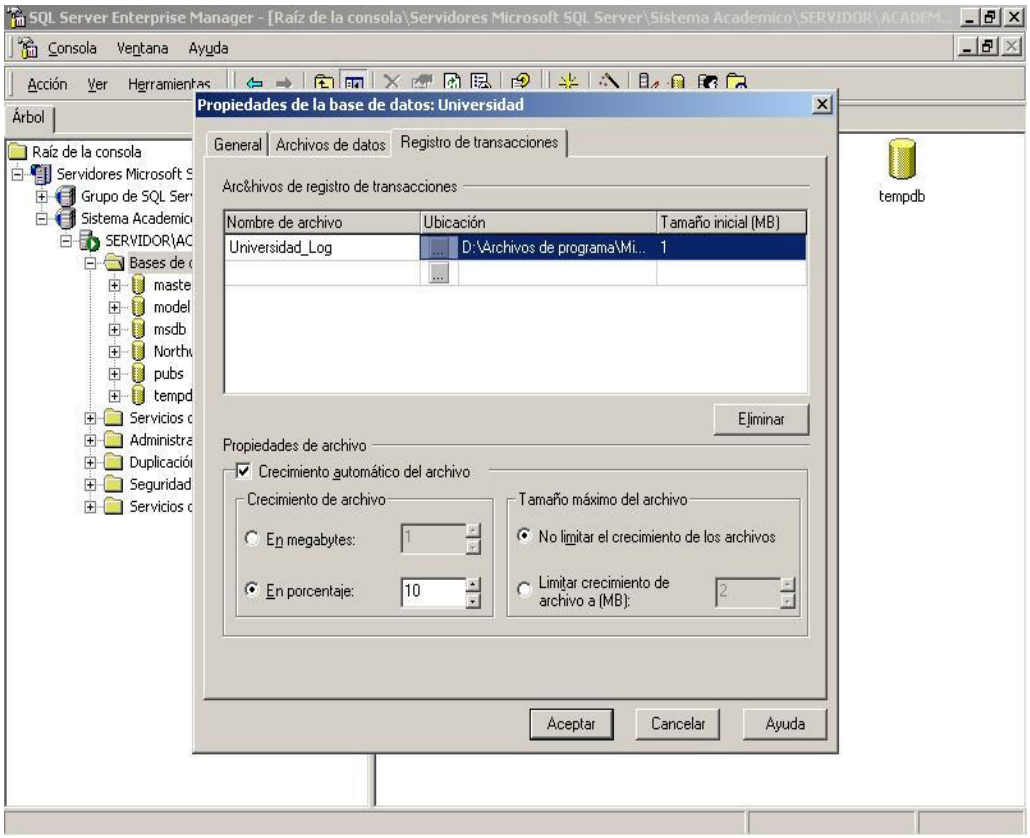


Figura 4.32. Propiedades de la base de datos Universidad–Registros de transacciones

Una vez se hayan definido tantos archivos como se desee, pulsar Aceptar. SQL Server creará la base de datos. Volver al Administrador corporativo y pulsar la carpeta Bases de datos del servidor al que se

ha añadido la base de datos. En el panel derecho del Administrador corporativo se verá que SQL Server ha añadido un icono para esa base de datos.

4.4. Creación de las Tablas en la Base de Datos

Es crítico recordar que cuando se diseñe la base de datos se debe diseñar toda la base de datos y sus relaciones antes de empezar a crearlas. Como tenemos los scripts de las tablas, crearemos las tablas utilizando el Analizador de consultas siguiendo estos pasos:

1. En el Analizador de Consultas, desplegar y seleccione el nombre del Servidor y la instancia en donde se encuentra la Base de Datos (SERVIDOR\ACADEMICO). También ingrese el nombre de inicio de sesión (nombre del administrador sa) y su respectiva clave.

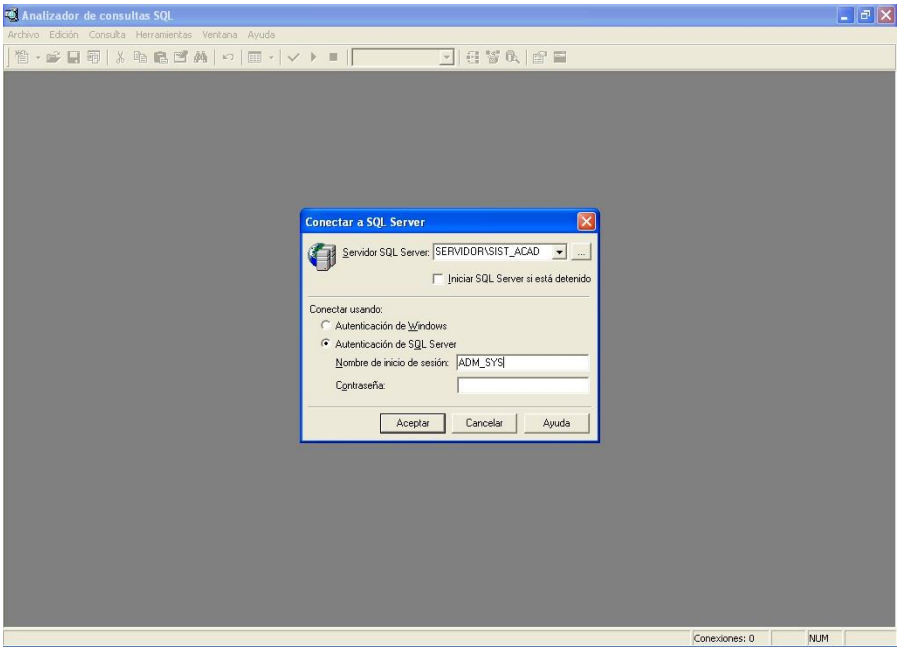


Figura 4.33. Analizador de Consultas SQL

2. Despliegue el listado de las Bases de Datos para seleccionar la Base de Datos que creó con anterioridad (Universidad).

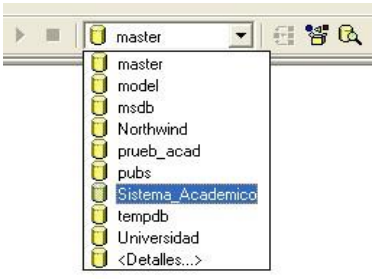


Figura 4.34. Selección de Base de Datos

3. En el menú Archivo-Abrir, busque el archivo “scripts.sql”, este archivo contiene los scripts de creación de las tablas.

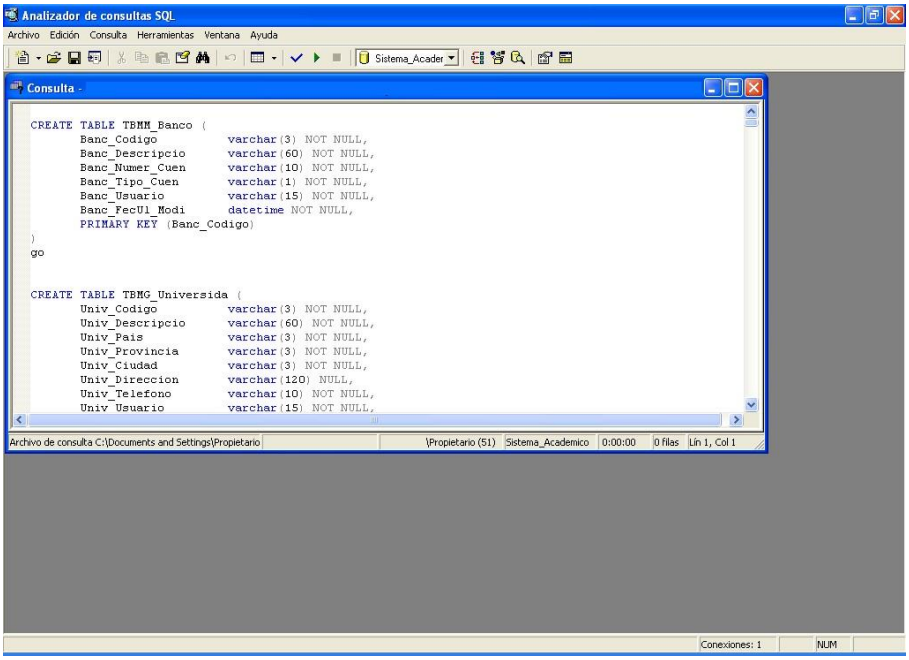


Figura 4.35. Script de Creación de Tablas

4. Proceda a ejecutar el archivo abierto.
5. Aparecerá un mensaje en la pantalla indicando “Comandos completados con éxito”.
6. Ahora puede cerrar el Analizador de Consultas. Vaya al Administrador Corporativo, seleccione la Base de Datos y en la carpeta tablas visualizará las tablas creadas.

4.5. Creación de Stored Procedures en la Base de Datos

1. Realice los pasos 1 y 2 indicados en la Creación de Tablas.
2. En el menú Archivo-Abrir, busque el archivo "StoredProcedures.sql", este archivo contiene los scripts de creación de las Stored Procedures.
3. Proceda a ejecutar el archivo abierto.
4. Aparecerá un mensaje en la pantalla indicando "Comandos completados con éxito".
5. Ahora puede cerrar el Analizador de Consultas. Vaya al Administrador Corporativo, seleccione la Base de Datos y en la carpeta de "Procedimientos almacenados" visualizará los Stored Procedures creados.

4.6. Creación de Tablas de Auditoria y Triggers

1. Realice los pasos 1 y 2 indicados en la Creación de Tablas.
2. En el menú Archivo-Abrir, busque la carpeta de scripts de auditoria, que contiene la creación de las tablas y disparadores.
3. Proceda a ejecutar el archivo abierto. . Recuerde primero ejecutar los archivos de creación de tablas de auditoria y luego los scripts de triggers.

4. Por cada archivo ejecutado aparecerá un mensaje en la pantalla indicando “Comandos completados con éxito”.

4.7. Creación de usuario ADM_SYS (SYSTEM ADMINISTRATOR)

Sólo para el usuario ADM_SYS (administrador del sistema) se procede de la siguiente manera (para el resto de usuarios se lo puede hacer desde el sistema).

1. Nos dirigimos a la carpeta “Seguridad” y sobre el ícono de “Inicios de sesión” damos un click derecho y escogemos la opción “Nuevo inicio”.

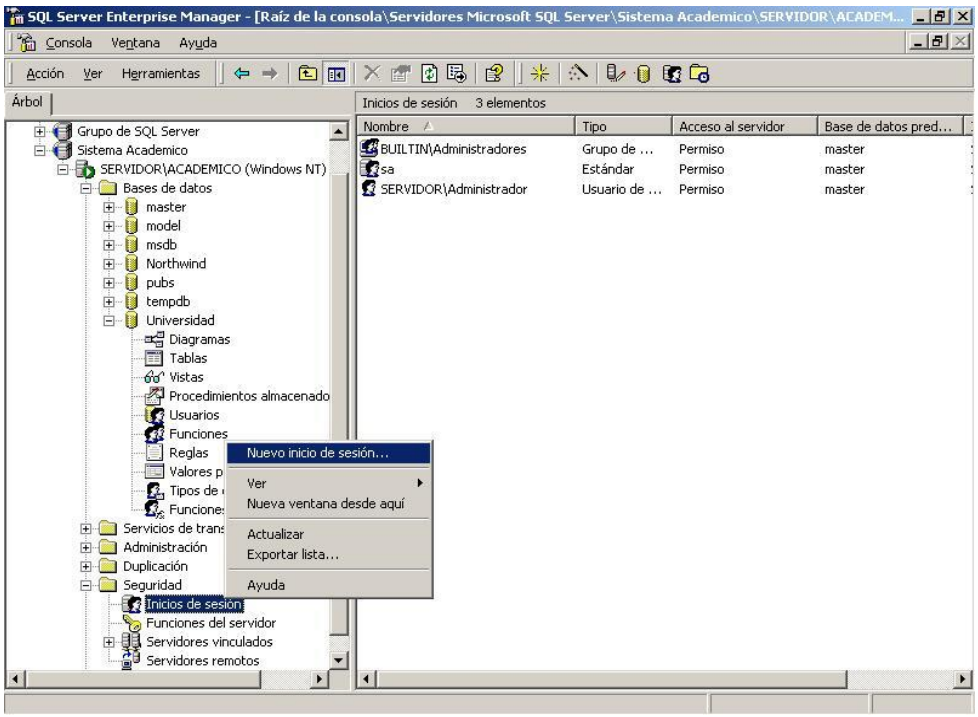


Figura 4.36. Nuevo inicio de sesión

2. Le asignamos el nombre “ADM_SYS”, escogemos la opción “Autenticación de SQL Server” y señalamos la base de datos predeterminada para este usuario que es “Universidad”.

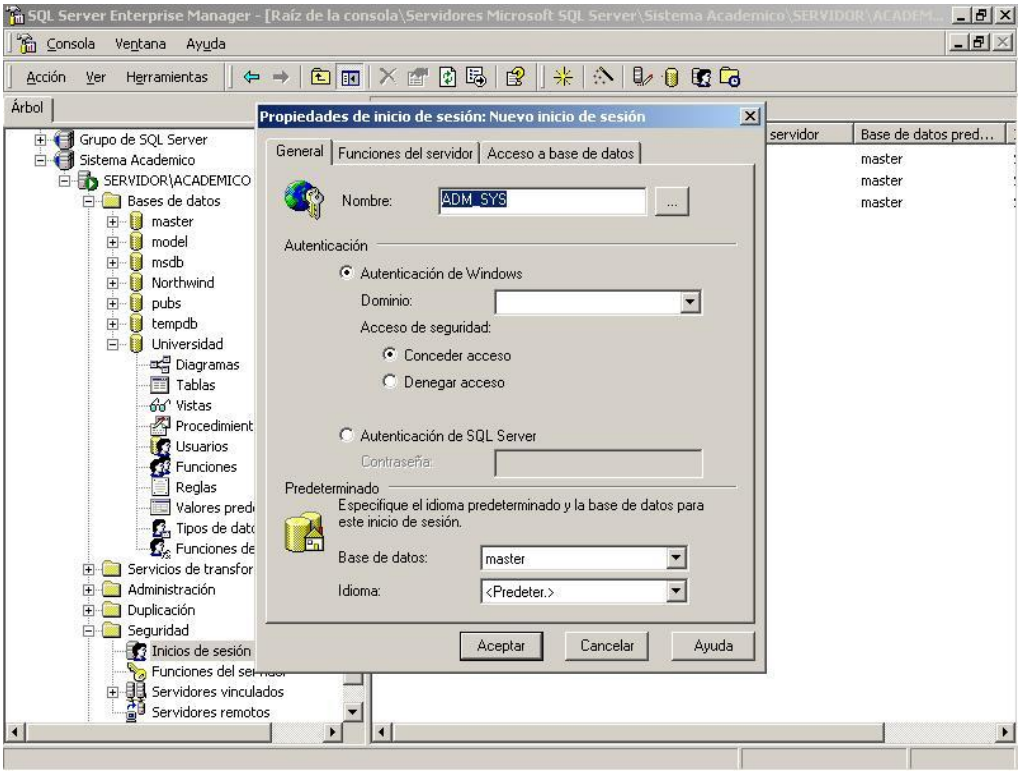


Figura 4.37. Propiedades de inicio de sesión-Nombre

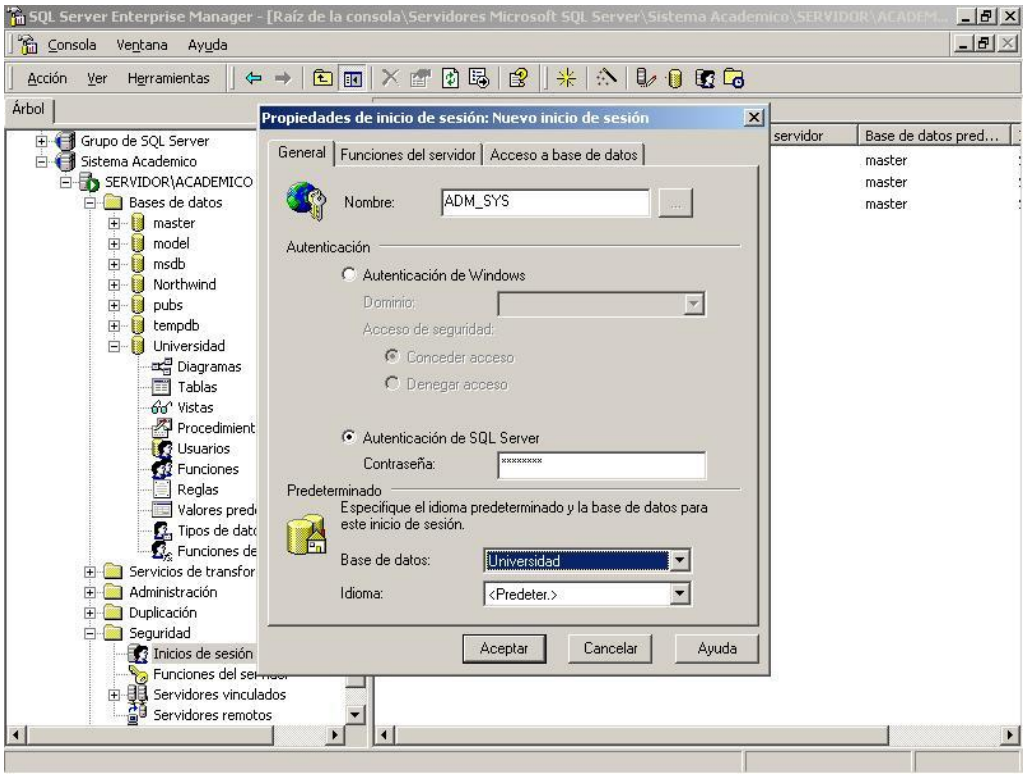


Figura 4.38. Propiedades de inicio de sesión-Autenticación

3. En la ficha “Funciones del servidor” señalamos qué funciones van a ser asignadas para este usuario, visteamos la opción “System Administrators”.

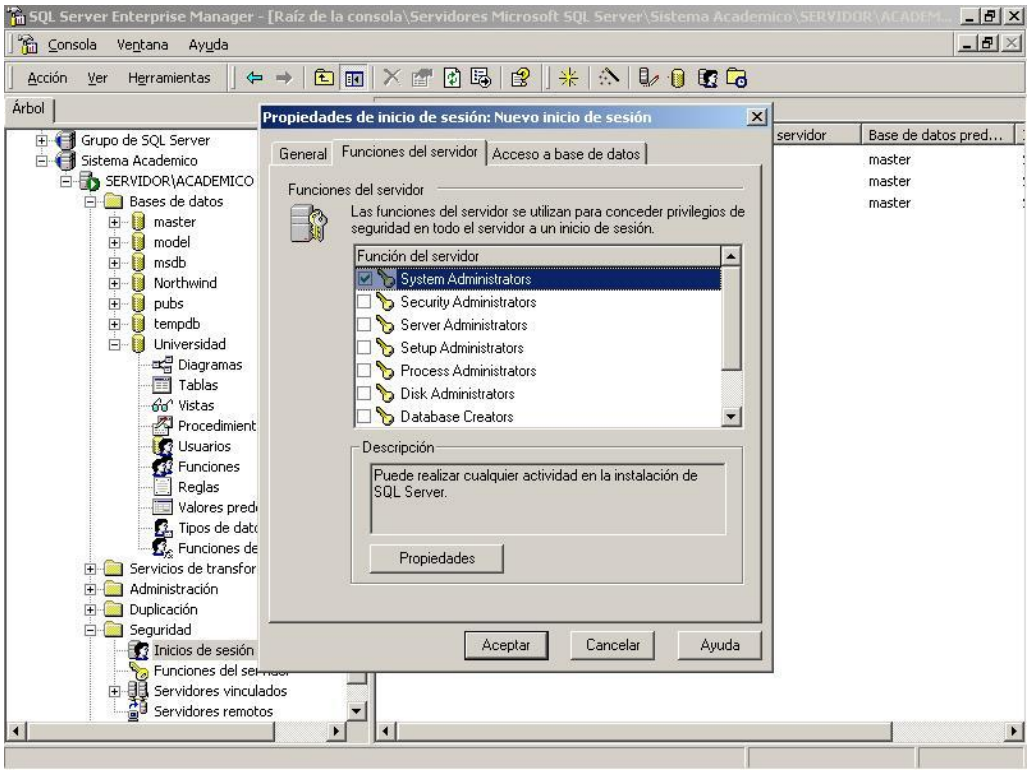


Figura 4.39. Propiedades de inicio de sesión-Funciones del servidor

4. En la ficha “Acceso a base de datos” tenemos que indicar en que bases de datos va a trabajar este usuario, señalamos “Universidad” y damos click sobre las siguientes opciones:

- Public (por defecto ya sale visteada)
- db_owner
- db_datareader
- db_datawriter

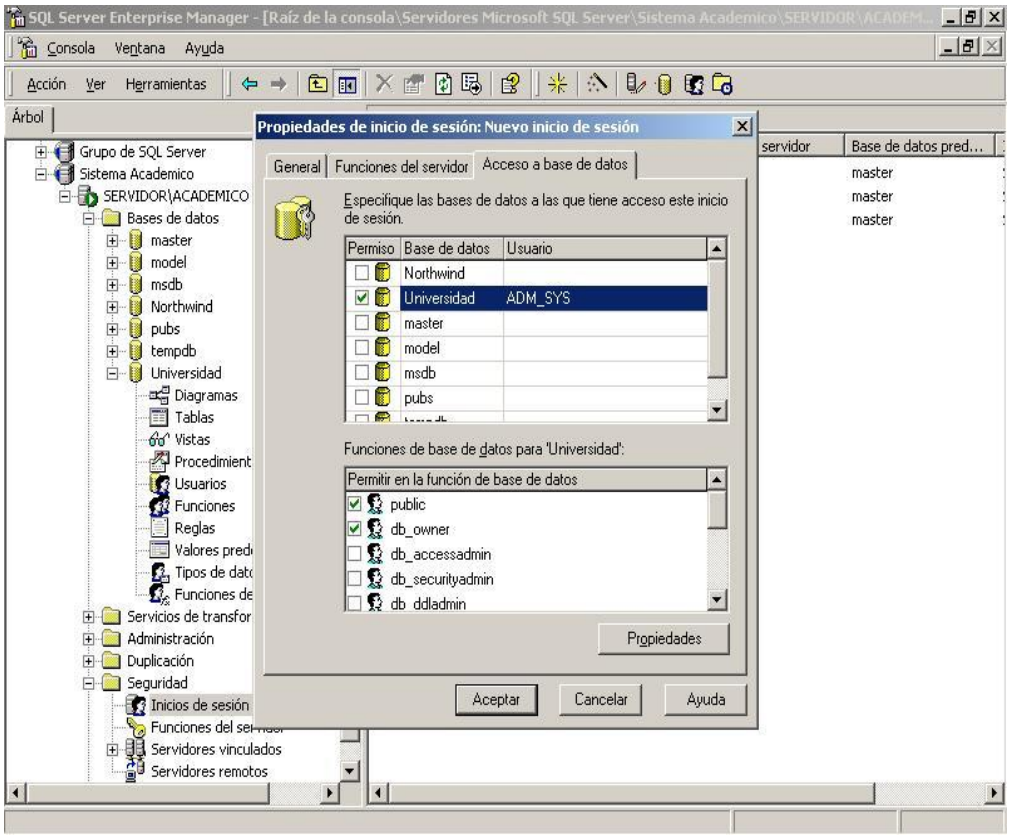


Figura 4.40. Propiedades de inicio de sesión-Acceso a base de datos

5. Damos click en el botón “Aceptar” y confirmamos la contraseña ingresada para este usuario. El siguiente y último paso es ejecutar el script AdmSys.sql tal como se ha ejecutado los scripts anteriores.

4.8. Instalación de Aplicación en el Cliente

El Sistema Académico funcionará si el equipo cliente tiene instalada el runtime de .NET Framework. El proyecto de instalación fue creado en Visual .Net y el proyecto se encargará de instalar nuestra aplicación y los

demás ficheros que necesitamos para el funcionamiento del sistema pero no instalará el runtime.

El runtime se instala en el equipo de destino al ejecutar el archivo dotnetfx.exe antes que el programa de instalación empiece a instalar. El archivo dotnetfx.exe se encuentra disponible en los medios de Visual Studio .Net "Windows Components Update" en www.microsoft.com y se redistribuye con el programa de instalación.

Para obtener el archivo antes indicado, primero debe detectar el idioma de instalación del sistema operativo del cliente, es decir, si es Windows 2000/NT/XP y si la versión está en español o inglés. Para mayor facilidad hemos incorporado en nuestro CD de los instaladores, el runtime .NET Framework en español e inglés versión 1.1

Bien, considerando la instalación previa del runtime, proceda a seleccionar el archivo Setup. Exe del CD de instalación de la aplicación:

1. En la carpeta de Sistema Académico, busque el archivo "Setup.exe" y presione Enter en el archivo seleccionado o doble clic en el mismo.

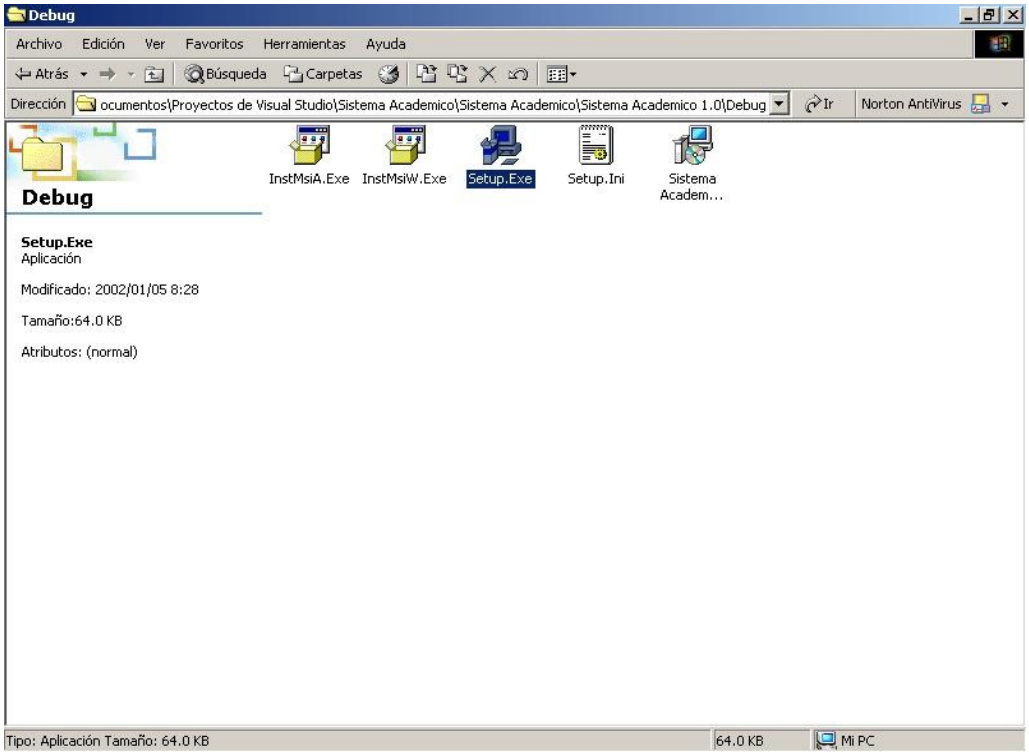


Figura 4.41. Archivo de instalación del Sistema Académico

2. Aparecerá un Asistente de Instalación del Sistema, presione Siguiente.

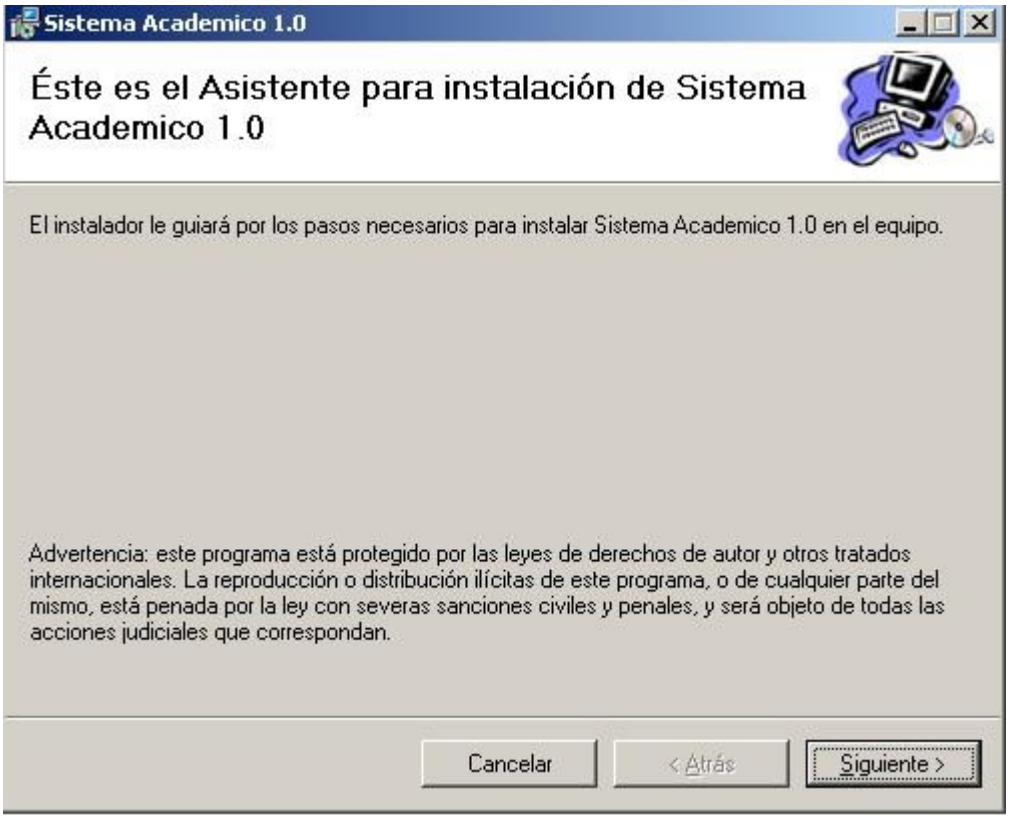


Figura 4.42. Pantalla inicial del instalador del Sistema Académico 1.0

3. Seleccione la carpeta de instalación. Por defecto, aparecerá una ruta preestablecida, si desea grabar los archivos en otra ruta, de clic en Examinar y seleccione la ruta deseada, caso contrario presione Siguiente. Note también que existe la opción para instalar a todos los usuarios creados en el equipo. Esta decisión dependerá del criterio del usuario.

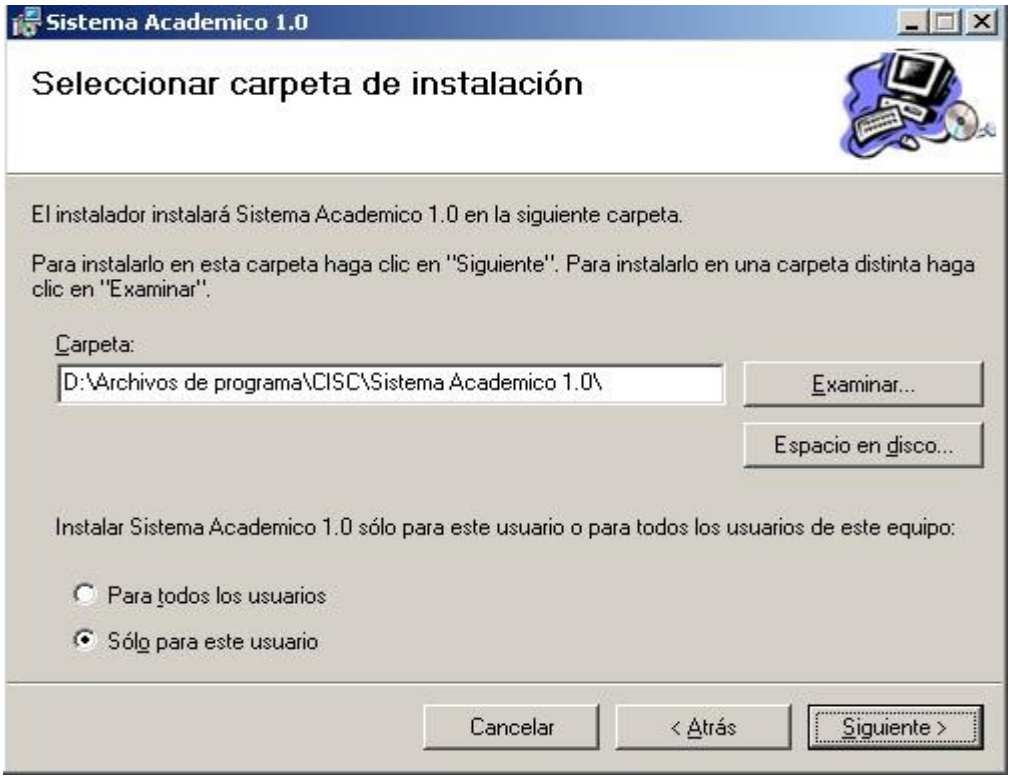


Figura 4.43 Seleccionar carpeta de instalación

4. Aparecerá una pantalla para confirmar la instalación, presione Siguiente.

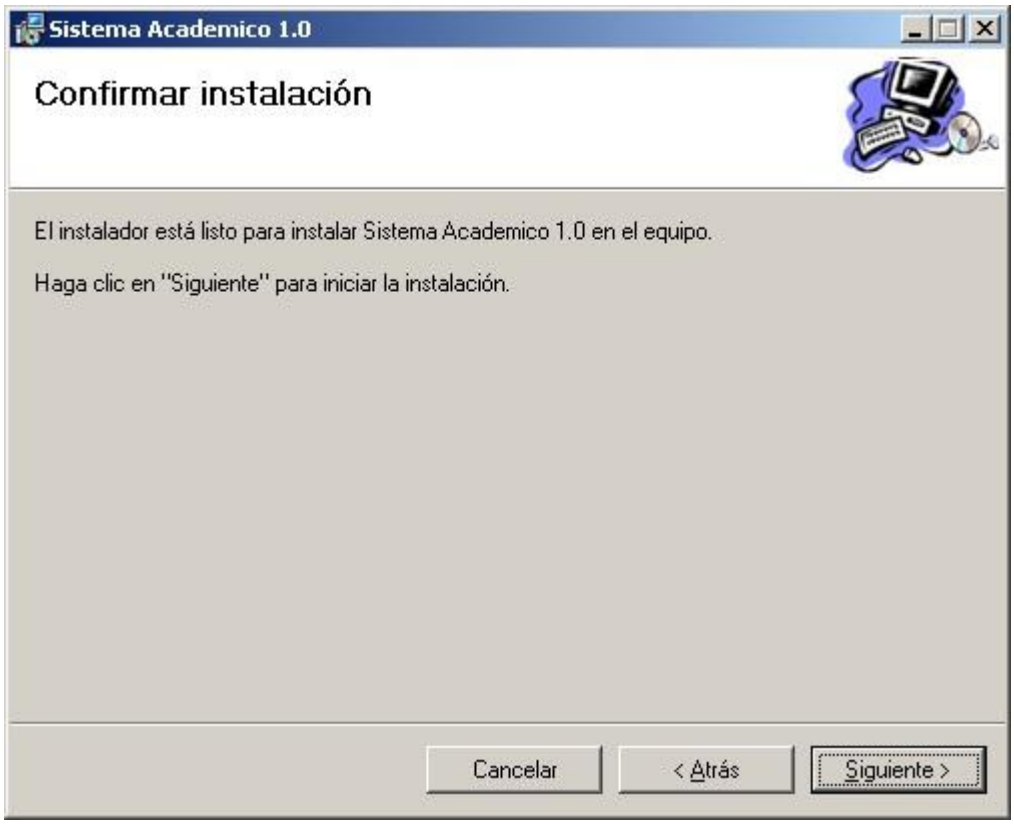


Figura 4.44. Confirmar la instalación del producto

5. Iniciado la instalación, espere hasta que se complete la copia de los archivos a la PC del cliente.

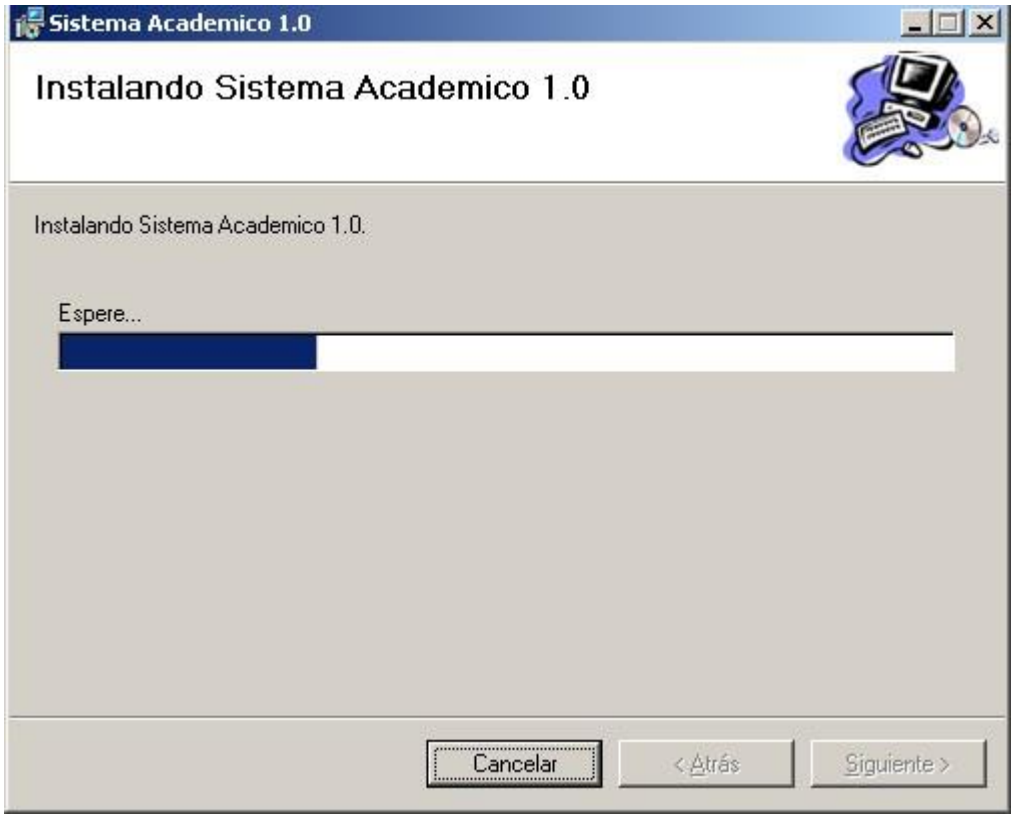


Figura 4.45. Instalando el Sistema Academico 1.0

6. Posteriormente, aparecerá una pantalla indicando que la Instalación está completada, presione Cerrar.

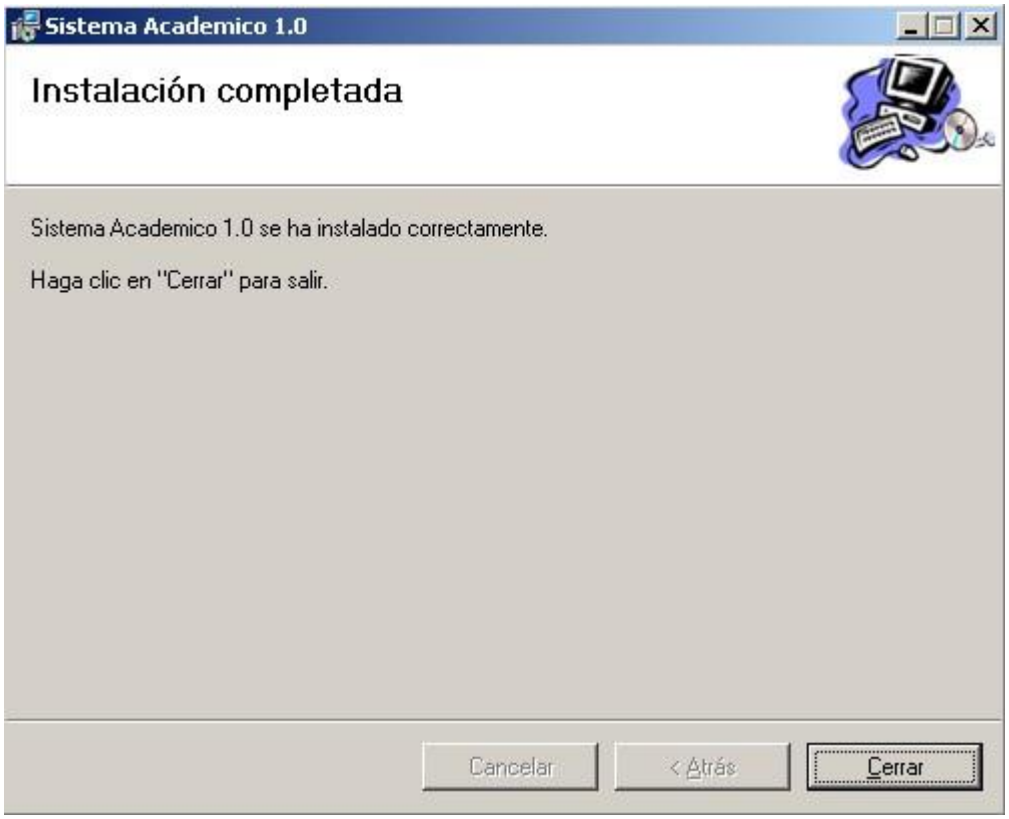


Figura 4.46. Instalación completada

7. Notará que en la carpeta “Archivos de programa” se ha creado una carpeta con el nombre de la compañía (depende del nombre del equipo del servidor) y en la misma se han copiado los archivos necesarios para el funcionamiento del Sistema Académico. Revise si se copió el ejecutable y cree un acceso directo en el escritorio.

4.9. Precauciones con el Archivo Base_Acad.ini

Este archivo contiene el nombre de la Base de Datos, el nombre del servidor y la instancia de la Base de Datos de SQL Server. Estos

parámetros pueden modificarse en el caso que cambien el nombre de la Base o la ruta del servidor. Tenga mucha precaución en el manejo de este archivo ya que es la entrada a los datos almacenados en la base.

Figura 4.47. Archivo Base_Acad.ini

CAPITULO 5

5. MIGRACION

5.1. Herramientas para exportación de datos

La Escuela de Ingeniería Civil no cuenta con un sistema automatizado de matriculación, por tal razón, para el empleo del nuevo sistema deberá requerir un personal de digitación para sistematizar su información.

El sistema que maneja la Carrera de Ingeniería de Sistemas está elaborado en Fox para DOS al igual que la base de datos.

Para la migración utilizamos Microsoft Visual FoxPro 6.0 porque es una herramienta más sofisticada y de mejor interfaz gráfica para el usuario. Mediante esta herramienta manejamos ciertos comandos para exportar las tablas a Excel y depurar los datos.

1. Abra la Tabla de Fox que procederá a migrar. Utilice el comando USE en la ventana de comando.
2. En la misma ventana de comando digite:

   COPY TO c:\tabla_exportar.XLS TYPE XL5
3. Automáticamente se copiarán los campos de la tabla como columnas en una hoja de Excel. En la ruta especificada, abra el archivo de excel.
4. En dicha hoja eliminar los símbolos especiales de todas las columnas tales como: ‘ , Ð, “,}, ╔. Estos símbolos corresponden a caracteres que poseen conflictos entre los sistemas operativos DOS y Windows, por ejemplo, Ñ, Á, á, ü
5. Eliminar de la hoja de Excel, las columnas que no sean necesarias, y a su vez incorporar las básicas –dependiendo de los constraints- previo a su conocimiento sobre la nueva estructura.

5.2. Tablas proporcionadas para migración

Detallamos las tablas que la Carrera de Ingeniería en Sistemas Computacionales nos proporcionó para la migración:

- Categorías
- Colegios
- Países

- Estudiantes
- Especializaciones
- Docentes
- Asignaturas
- Calificaciones

5.3. Depuración de datos

Como indicamos anteriormente, para la migración de datos se requiere de una exportación de las tablas de Fox a Excel para iniciar la actualización de los datos. A continuación indicamos las tablas que exportamos y depuramos en Excel:

- **TABLA CATEGORÍA:** colegio.dbf
  - La categoría de cada colegio oscila entre 1 y 7. Previo a la importación, el número debe tener el formato 000, es decir que la categoría oscilará entre “001” y “007”.
  - Debe crear un número secuencial.
  - Cree un registro con código 000 para indicar “Categoría Ninguna”, en el caso de que un estudiante no tenga especificado la categoría.
  - Se agruparon los colegios para determinar el número de categorías existentes.

- **TABLA COLEGIO:** colegio.dbf

- Existen colegios sin categoría, por defecto corresponden a colegios extranjeros. Para solucionar este inconveniente, previamente deberá crear categorías adicionales (en la nueva estructura) para unidades educativas extranjeras y otra categoría para estudiantes profesionales.
- Existen colegios con categoría 0 o código desconocido, estos colegios no fueron migrados porque en la base de Estudiantes no están registrados y son:

Código	Colegio	Provincia	Categoría
0848	CAD	0	0
0992	P	0	0
0799	LBER	GUAYAQUIL	0
0856	TENIENTE MAXIMILIANO RODRIGUEZ	LOJA	0
0900	DR. ENRIQUE NOBOA ARIZAGA	LA TRONCAL	C
0938	OCHO DE NOVIEMBRE	BOLIVAR	D
0896	CD	D	D
1077	BEST	VINCES	D
0927	TEC.PASAJE	EL ORO	N
1104	N	GUASYAS	N

Tabla 5.1. Datos inconsistentes de la Tabla Colegio (1)

0939	REDEMPTIO	GUAYAS	P
1084	TCRN. LAURO GUERRERO	LOJA	P
1103	TE	S	S
1076	N A	G	V

Tabla 5.2. Datos inconsistentes de la Tabla Colegio (2)

- En el archivo, notará que no existe un campo que indique si el colegio es P o F, es decir, Particular o Fiscal, para ello hemos considerado de la categoría 001 al 005 la letra P para colegios Particulares y del 006 al 007 la letra F para colegios Fiscales.
- Crear el registro 0000 para indicar “Colegio Ninguno”, en el caso de que un estudiante no tenga especificado el colegio.

➢ **TABLA PAIS:** pais.dbf

- Considere el formato 000 para el código de país.
- Crear el registro 000 para “Ningún país”, en el caso de que no exista este dato en las tablas que lo requieran.
- Existen ciertos registros sin código de país. Especificamos los datos no migrados por dicho inconveniente:

Código País	País	Nacionalidad
017		ARGENTINA-ECUATORIANA
018		BOLIVIANO-ECUATORIANO
019		VENEZOLANO-ECUATORIANO
020		BRASILEÑO-ECUATORIANO
021		COLOMBIANO-ECUATORIANO
022		CHILENO-ECUATORIANO
023		ESTADOUNIDENSE-ECUATORIANO
024		MÉXICO-ECUATORIANO
025		PARAGUAYO-ECUATORIANO
026		PERUANO-ECUATORIANO
027		URUGUAYO-ECUATORIANO
028		CANADIENSE-ECUATORIANO
029		FRANCO-ECUATORIANO

Tabla 5.3. Datos inconsistentes de la Tabla País

- **TABLA ESTUDIANTE:** dgene.dbf

- Cree un número secuencial para el código del estudiante, cumpliendo con el formato 0000, de acuerdo al tamaño del campo de la nueva estructura.

- Para ciertos registros no existen datos que en la nueva estructura son selectos. Considere lo siguiente:

Campo	**Solución**
Código del Estudiante	Registre un número secuencial
Sexo	Revisar manualmente e ingresar el campo 1 o 2 (Masculino/Femenino)
Fecha de Nacimiento	Temporalmente ingresar fecha ficticia 1900/01/01
Estado Civil	1 Soltero; 2 Casado; 3 Unión Libre; 4 Viudo
Record Policial	Ingrese 0
Cédula Militar	Ingrese 0
Domicilio	Ingrese --
Teléfono	####
Colegio	Ingrese código 0000 (Previamente ingresado en nueva estructura)
Categoría	Ingrese código 000 (Previamente ingresado en nueva estructura)
Año de Graduación	Ingrese 1900

Tabla 5.4. Datos no encontrados en la Tabla Estudiante (1)

Carnet de Afiliación	Digite 0
Nacionalidad	Digite código 000 (Previamente ingresado en nueva estructura)
Provincia de Nacimiento	Digite 000 (Previamente ingresado en nueva estructura)
Ciudad de Nacimiento	Digite 000 (Previamente ingresado en nueva estructura)
E-mail	Identificación ficticia ug@migra.com
Grupo Sanguíneo	O+
Examen de Gracia	Ingrese “N”
Nota de Grado	0
Universidad	Ingrese código 000 (Previamente ingresado en nueva estructura)

Tabla 5.5. Datos no encontrados en la Tabla Estudiante (2)

- Existe un registro en blanco que no fue migrado.
- No se migraron los siguientes estudiantes por las siguientes observaciones detalladas:

Cedula	Apellido	Nombre	Observación
0914783154	AMAYA JURADO	CARLOS RONALD	Colegio con código LOCO
0921268322	HURTADO ESCALANTE	LOURDES DEL ROCIO	Colegio inexistente
0801405820	ANGELA MERCEDES	MOTATO TUMBACO	Cédula duplicada M-1250

Tabla 5.6. Datos inconsistentes de la Tabla Estudiante

- **TABLA ESPECIALIZACION:** especia.dbf

- Se migraron todas las especializaciones y se reemplazó su código con la equivalencia que a continuación detallamos:

Código	Descripción	Código Nuevo
	NINGUNO	00
11	FISICO MATEMATICO	01
12	QUIMICO BIOLOGICAS	02
13	CIENCIAS SOCIALES	03
14	NORMALISTAS (EDUCACION)	04
15	INFORMATICA	05

Tabla 5.7. Datos de la Tabla Especialización (1)

16	MATEMATICAS FISICAS	06
17	MATEMATICAS QUIMICAS	07
18	MATEMATICAS BIOLOGICAS	08
19	OTROS EN AREA CIENCIAS	09
21	SECRETARIADO EN ESPAÑOL	10
22	SECRETARIADO BILINGÜE	11
23	CANTABILIDAD	12
24	ADMINISTRACION	13
25	COMERCIALIZACION	14
26	ARCHIVOLOGIA	15
27	COMPUTACION	16
28	MERCADEO	17
29	TURISMO	18
31	MECANICA GENERAL	19
32	MECANICA INDUSTRIAL	20
33	MECANICA AUTOMOTRIZ Y DIESEL	21
34	METALMECANICA	22
35	ELECTROMECANICA	23
36	MOTORES DE AVIACION	24
37	MATRICERIA	25

Tabla 5.8. Datos de la Tabla Especialización (2)

38	ELECTRICIDAD	26
39	ELECTRONICA	27
41	AGRICOLA	28
42	PECUARIA	29
43	AGROINDUSTRIA DE LOS ALIMENTOS	30
44	MECANICA AGRICOLA	31
45	FORESTAL	32
46	ADMINISTRACION DE GRANJAS	33
47	AGRONOMIA	34
48	RIEGO Y DRENAJE	35
49	PISCICULTURA	36
51	MUSICA	37
52	TEATRO	38
53	DANZA	39
54	PINTURA	40
55	ESCULTURA	41
56	ARTE GRAFICO	42
57	CERAMICA	43
58	DISEÑO PLASTICO	44

Tabla 5.9. Datos de la Tabla Especialización (3)

59	DECORACION	45
61	PROFESOR EDUCACION PRIMARIA	46
62	PROFESOR EDUCACION SECUNDARIA	47
63	OTROS AREA FORMACION DOCENTE	48
71	OTROS	49
210	BIBLIOTECOLOGIA	50
211	OTROS EN AREA COMERCIO	51
310	REFRIGERACION,A. ACONDICIONADO	52
311	TECNOLOGIA PETROQUIMICA	53
312	TECNOLOGIA PESQUERA	54
313	CONSTRUCCION NAVAL	55
314	TOPOGRAFIA	56
315	INDUSTRIA DE LA CONSTRUCCION	57
316	CARPINTERIA Y EBANISTERIA	58
317	INDUSTRIA DEL VESTIDO	59
318	EDUCACION PARA EL HOGAR	60
319	PEQUEÑAS INDUSTRIAS	61
320	DISEÑO Y DECORACION INTER.	62

Tabla 5.10. Datos de la Tabla Especialización (4)

321	ARTES MANUALES	63
322	OTROS AREA TECNICO INDUSTRIAL	64
410	ACUACULTURA	65
411	AGRIMENSURA	66
412	OTROS EN AREA AGROPECUARIA	67
510	OTRO EN AREA ARTE	68

Tabla 5.11. Datos de la Tabla Especialización (5)

- **TABLA DOCENTE:** paralelo.dbf

- Los únicos datos válidos proporcionados de los archivos del sistema académico para migrar son los campos nombre y apellidos de los docentes, así también como el título universitario de cada uno de ellos.
- Cree un número secuencial para el código del docente, cumpliendo con el formato 0000, de acuerdo al tamaño del campo de la nueva estructura.
- Existe duplicidad en los nombres de los docentes ya que esta tabla es para registrar al docente en que grupo va a dar, aún así los nombres están mal ingresados damos como ejemplo:

IN. ITURBURU

ING. DOUGLAS ITURBURU

- Para la tabla de docentes no existen los datos correspondientes a los campos:

Campo	**Solución**
Código del Docente	Registre un número secuencial
Cédula/Pasaporte del docente	9999999999
Nacionalidad	Ingrese código 000 (Previamente ingresado en nueva estructura)
Provincia de Nacimiento	Ingrese código 000 (Previamente ingresado en nueva estructura)
Ciudad de Nacimiento	Ingrese código 000 (Previamente ingresado en nueva estructura)
Fecha de Nacimiento	Temporalmente ingresar fecha ficticia 1900/01/01
Cédula Militar	9999999999 / 0
Record Policial	9999999999 / 0
Provincia de Residencia	Ingrese código 000 (Previamente ingresado en nueva estructura)
Ciudad de Residencia	Ingrese código 000 (Previamente ingresado en nueva estructura)
Domicilio	- -

Tabla 5.12. Datos no encontrados en la Tabla Docente (1)

Teléfonos	9999999999 / 0
Empresa	- -
Dirección de Trabajo	- -
Carnet de Afiliación	9999999999 / 0
Universidad	Ingrese código 000 (Previamente ingresado en nueva estructura)
Título Universitario	Registre el código que le corresponda al docente
Logros Obtenidos	- -

Tabla 5.13. Datos no encontrados en la Tabla Docente (2)

- Los títulos universitarios de los docentes debe ingresarse manualmente y previo a la migración de la tabla de docentes debido a que no existe la tabla de títulos siendo este necesario para el funcionamiento en la pantalla de mantenimiento de docentes en nuestro sistema.
- Para mayor facilidad, revise el contenido del campo nombre y apellido del docente porque se incluye el título en dicho campo.
- Existen títulos de docentes no actualizados, por ejemplo:

ECON. CARLOS ORTIZ

MSC. CARLOS ORTIZ

- **TABLA ASIGNATURA:** pensum.dbf

- Para migrar esta tabla deberá tener datos ingresados en la tabla Facultad y Subunidad de donde proviene dichas asignaturas.
- Ingrese en la nueva estructura, “Area Académica ninguna” porque no existe dicho dato que es importante durante la migración.
- Cree un número secuencial para el código de la asignatura, cumpliendo con el formato 000, de acuerdo al tamaño del campo de la nueva estructura.
- Especifique el nivel en que se cursa la asignatura. Para la tabla de asignaturas no existen los datos correspondientes a los campos:

Campo	**Solución**
Código de la Asignatura	Registre No. secuencial respetando formato 000
Subunidad	Código de unidad académica
Area Académica	Ingrese código 000 (Previamente ingresado en nueva estructura)
Horas Semanales	Digite 0
Tipo de Asignatura	Digite 0
No. Créditos ganados x Materia	Digite 0

Tabla 5.14. Datos no encontrados en la Tabla Asignatura

- No existe tabla de prerrequisitos de asignaturas, esta tabla es necesaria tanto para el mantenimiento de asignaturas como para los horarios en nuestro sistema.
- No existe una tabla de malla curricular que será necesario para poder crear la tabla prerrequisito.

➢ **TABLA DOCENTE ASIGNATURA:** paralelo.dbf

Docente Asignatura es una relación indispensable en nuestro sistema, las tablas que se requiere son: tabla docente y tabla asignatura.

Para efectuar la migración de estas dos tablas revise la información de cada una de las tablas indicada anteriormente.

- Emplee el archivo paralelo.dbf para determinar los grupos existentes, el periodo lectivo, el ciclo y el cupo de estudiantes en cada asignatura.
- El campo grupo debe encadenarse con el código de subunidad, es decir, grupo estará formado por código de subunidad + identificación de grupo.
- Un campo relevante en la nueva estructura es “tipo de curso”, esto es, las diferentes clases de cursos, seminarios, y periodos lectivos que pueden crearse periódicamente.

- El campo “tipo de curso” también estará encadenado con el código de la subunidad.

➢ **TABLA ESTUDIANTE SUBUNIDAD:** regma.dbf

La relación de las tablas Estudiante-Subunidad permite indicar a qué unidad académica está registrado el estudiante. Recuerde que la tabla estudiante en la nueva estructura constituye una ficha global de la universidad, pero estudiante-subunidad comprende el conjunto de estudiantes por cada unidad.

- La nueva estructura requiere de un campo “último nivel aprobado” y este dato se encuentra en el archivo regma.dbf
- Es necesario el campo “pensum” que no existe en las tablas proporcionadas por la carrera. Para solucionar el inconveniente, ingrese un código ficticio en la tabla pénsum de la nueva estructura previo a la migración de la tabla estudiante-subunidad.
- Para la tabla estudiante-subunidad no existen los datos correspondientes a los campos:

Campo	**Solución**
Código de Subunidad-Estudiante	Registre código de subunidad + código de estudiante. Verifique código a utilizar en referencia de cada tabla.
Código de Subunidad	Código de Subunidad
Código de Estudiante	Ingrese código 000 (Previamente ingresado en nueva estructura)
Código de Pénsum Actual	Digite “ “
No. Cruce de horario permitido	Digite 0
Fecha de Ingreso a Subunidad	Ingrese fecha ficticia 1900/01/01
Fecha de Egreso a Subunidad	Ingrese fecha ficticia 1900/01/01
Condición Académica	Digite “A“ (Activo)
Promedio de Egresado	Digite 0
Promedio de Grado	Digite 0
Promedio Final	Digite 0
Código Proyecto de Grado	Digite 0
No. Promoción	Digite 0
Observaciones	Digite “ “

Tabla 5.15. Datos no encontrados en la Tabla Estudiante-Subunidad

➢ **TABLA ESTUDIANTE ASIGNATURA:** regma.dbf

La relación de las tablas Estudiante-Asignatura depende de la migración de Estudiante-Subunidad (revise anteriores indicaciones) y de la tabla asignatura.

- En el archivo regma.dbf obtenemos los campos: estudiante, asignatura, año lectivo, ciclo, grupo, promedio.
- Recuerde agregar el código de subunidad a los campos: estudiante, asignatura, grupo para cumplir con el formato solicitado.
- Para la tabla estudiante-asignatura no existen los datos correspondientes a los campos:

Campo	**Solución**
Subgrupo	Digite “ “
Asignatura Anulada	Digite “ “
Asignatura Aprobada	Digite “ “
Observaciones	Digite “ “

Tabla 5.16. Datos no encontrados en la Tabla Estudiante-Asignatura

➢ **TABLA ESTUDIANTE NIVEL:** regma.dbf

La relación de las tablas Estudiante-Nivel depende de la migración de Estudiante-Subunidad (revisa indicaciones dadas en el manual). Esta

relación contiene información histórica del nivel que cursa el estudiante así como las asignaturas cursadas, aprobadas y reprobadas.

- Como fuente utilice el archivo regma.dbf para obtener los campos: estudiante, año lectivo, ciclo, grupo, nivel, número de asignaturas tomadas, número de suspensos.
- Cree un número secuencial para el código de matrícula. El formato de código de matrícula está conformado por: periodo lectivo (año y mes de matriculación) + número secuencial (formato 0000)
- Recuerde agregar el código de subunidad al campo estudiante.
- En el campo “tipo de curso” utilice el esquema explicado en “Tabla Docente Asignatura”
- Para la tabla estudiante-nivel no existen los datos correspondientes a los campos:

Campo	**Solución**
No. Matrícula	Registre número secuencial
No. Vez	Digite 0
Promedio	Digite 0
No. Asignaturas aprobadas	Digite 0
No. Asignaturas reprobadas	Digite 0

Tabla 5.17. Datos no encontrados en la Tabla Estudiante-Nivel (1)

Premiado	Digite “N”
Sancionado	Digite “N”
Observaciones	Ingrese “ ”

Tabla 5.18. Datos no encontrados en la Tabla Estudiante-Nivel (2)

5.4. Importación de datos

Posteriormente a la depuración de todos los datos relevantes para el correcto funcionamiento del Sistema Académico, realice la importación de datos a la nueva estructura considerando los siguientes pasos:

1. Utilice el Wizard para importación:

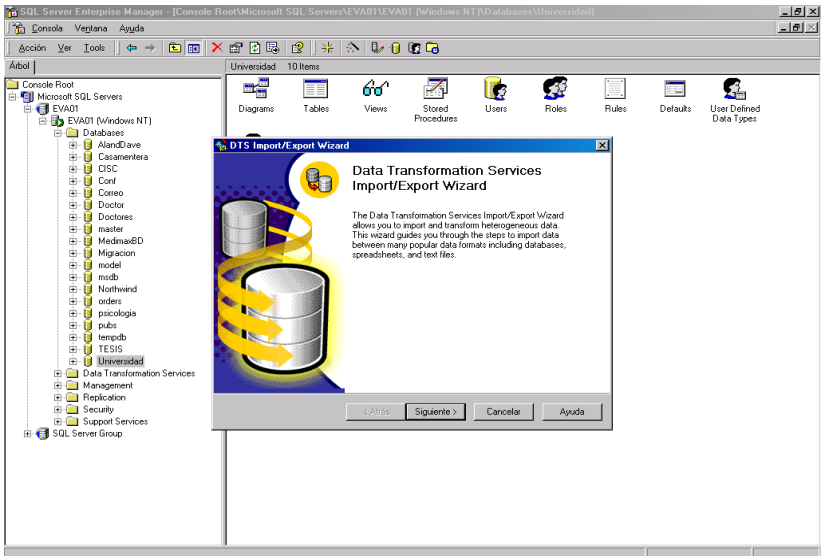


Figura 5.1. Asistente para la Importación/Exportación de datos

2. Seleccione el origen de datos que en este caso sería el archivo de Excel

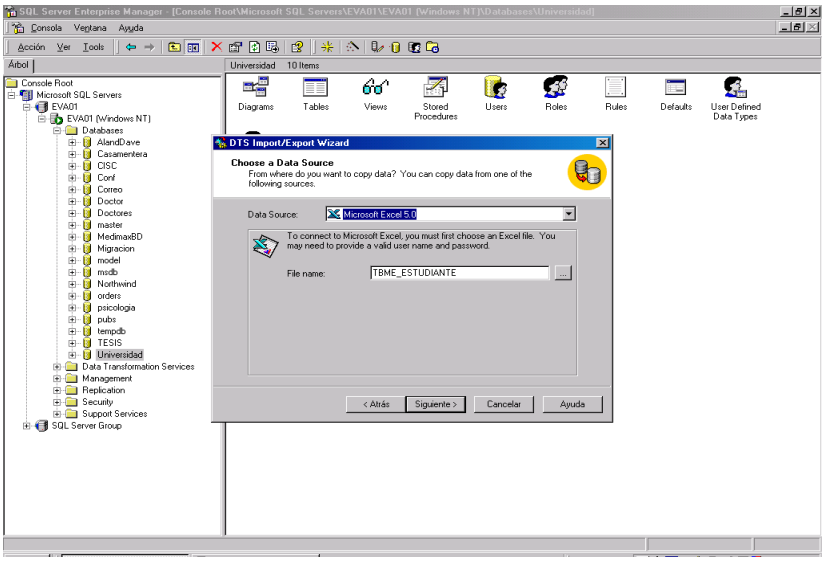


Figura 5.2. Seleccionando Origen de Datos

3. Seleccione el destino, es decir, la base donde se almacenará los datos.

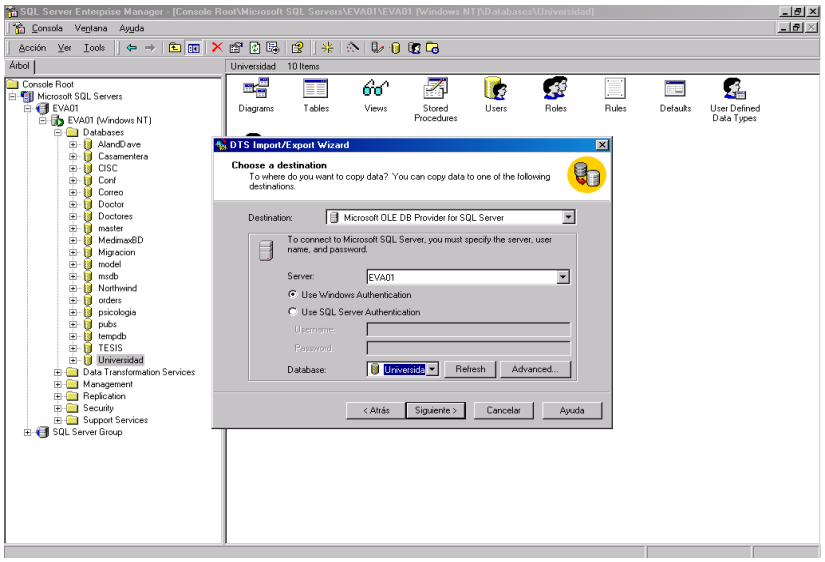


Figura 5.3. Seleccionando el destino de Datos

4. Especifique la opción de copia a nivel de tablas

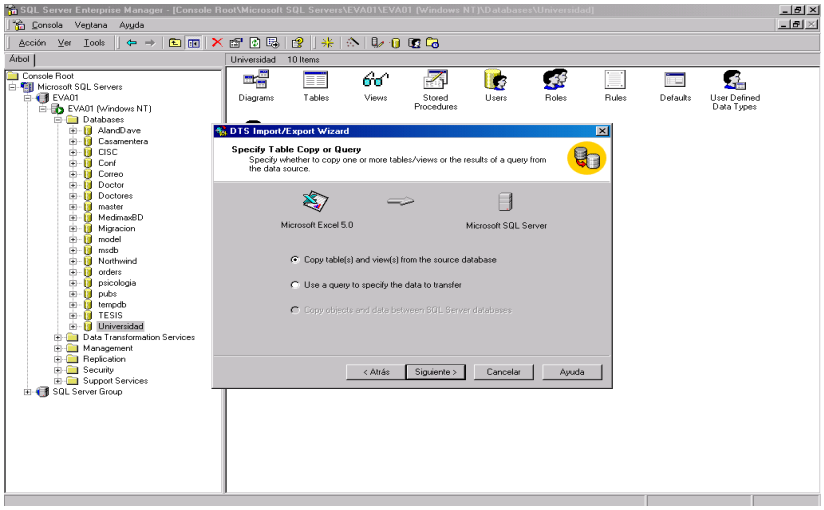


Figura 5.4. Seleccionando una Opcion de Copia

5. Indique la(s) hoja(s) de Excel que procederá a migrar. Por cada hoja seleccionada, especifique el nombre de la tabla en donde se guardarán los datos (destino)

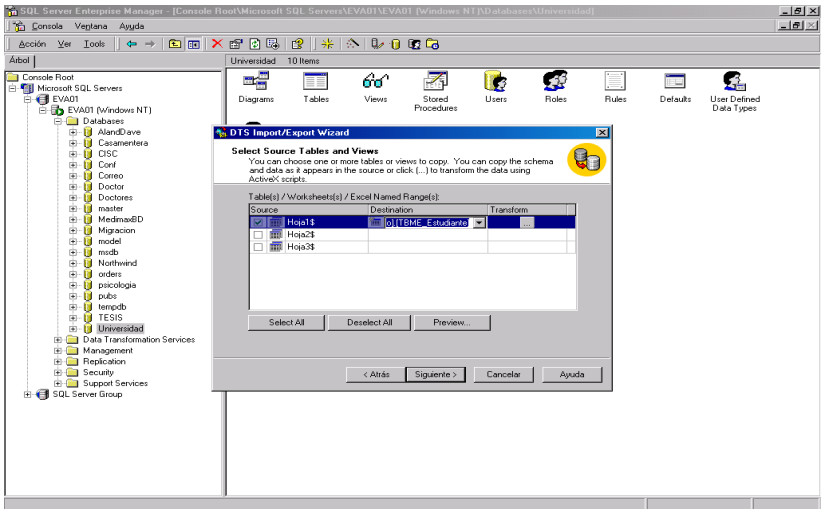


Figura 5.5. Señalando los archivos a migrar y tablas de destino

6. Seleccione “Siguiente” para ejecutar inmediatamente la migración

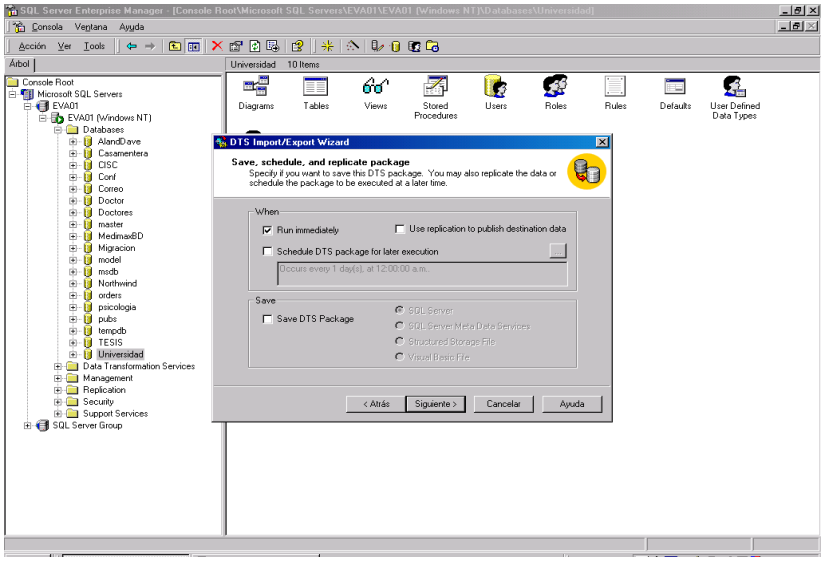


Figura 5.6. Ejecutar la migración

7. Seleccione Finalizar para Iniciar el proceso de migración:

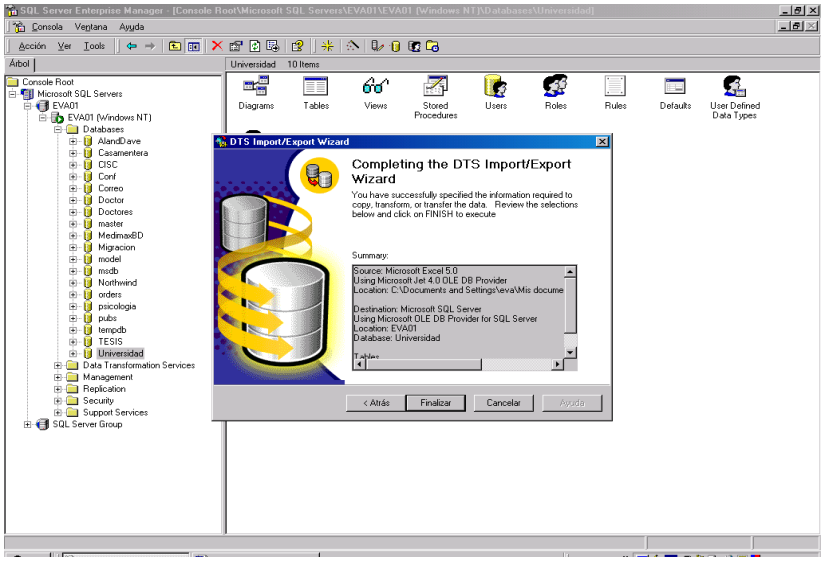


Figura 5.7. Iniciar el proceso de migración

CAPITULO 6

6. Copias de Seguridad: BACKUP

6.1. Copias de Seguridad de SQL Server 2000

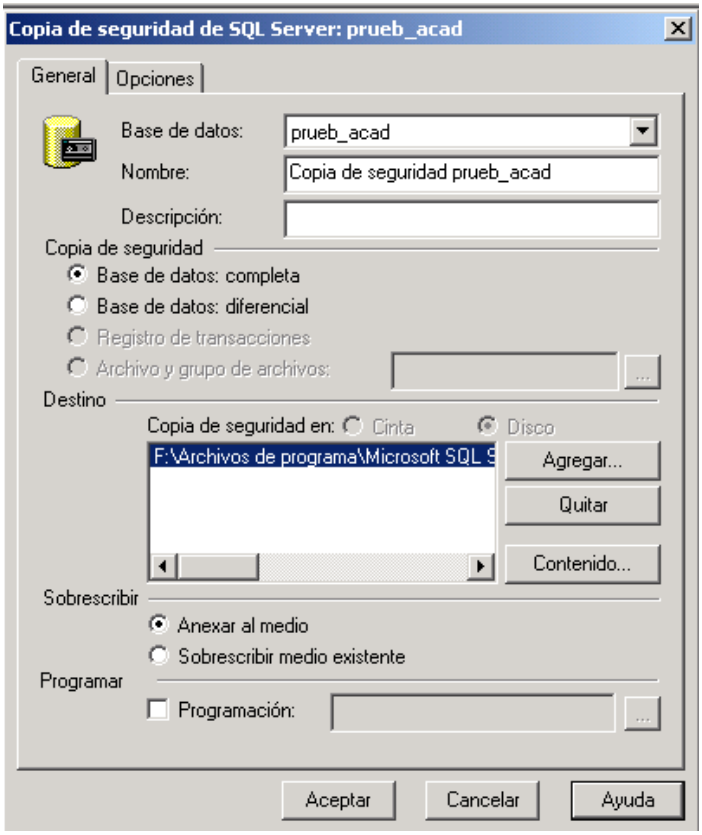


Figura 6.1. Acceso a la configuración de Backups

Permite realizar la copia de seguridad de una base de datos completa, del registro de transacciones o de uno o más archivos o grupos de archivos.

6.2. Pasos para realizar Backup de la Base de Datos

1. En el escritorio buscar el icono Administrador Corporativo de SQL SERVER 2000 y dar doble clic

Figura 6.2. Acceso Directo del Adm. Corporativo

2. Cargará la pantalla del SQL Server Enterprise Manager

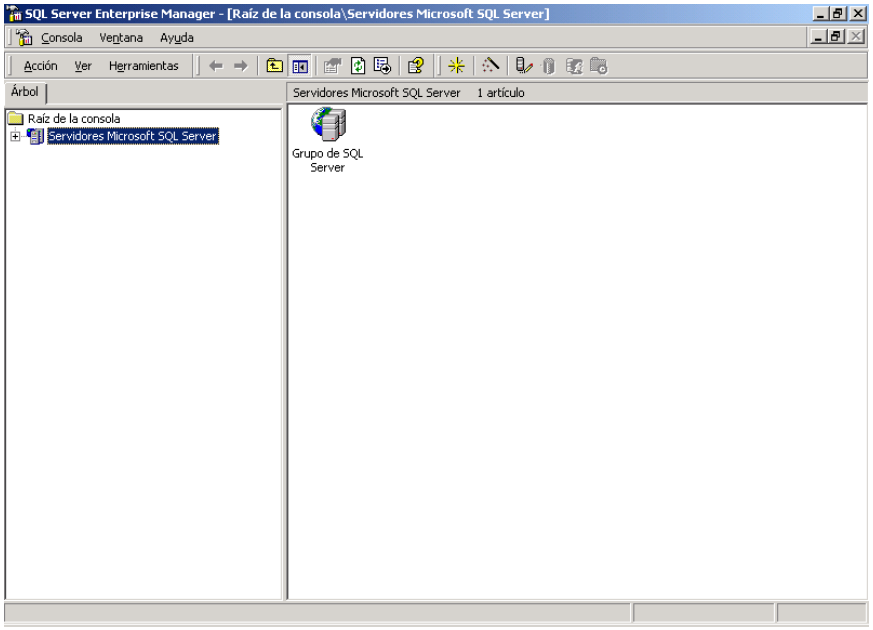


Figura 6.3. Pantalla SQL Server Enterprise Manager

3. En el Frame izquierdo busque el servidor donde se encuentra la base de datos dando clic en cada una de las raíces del árbol, es decir que el SQL SERVER 2000 puede administrar varios servidores sean locales o en red y estas a su vez pueden tener varias bases de datos.

Figura 6.4. Servidor de la Base de Datos

4. Dar clic derecho en el ítem Bases de Datos, luego dar clic en el ítem Todas las Tareas y luego en Copia de Seguridad de la Base de Datos.

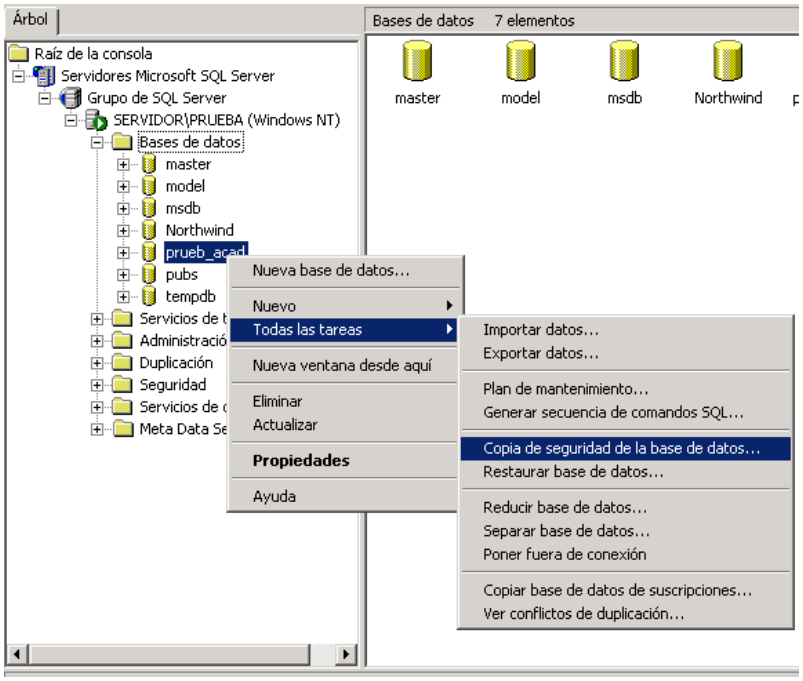


Figura 6.5. Acceso al Backup

5. Se cargará la Ficha General de Backup

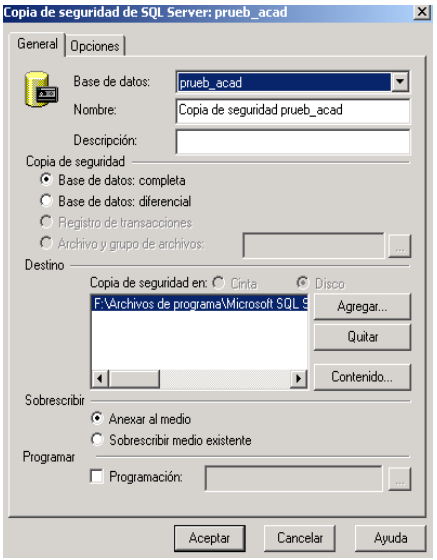


Figura 6.6. Ficha General de Backup

Describiremos cada uno de los ítems de esta ficha a continuación:

- **Base de datos:** Especifica la base de datos de la que desea realizar una copia de seguridad.
- **Nombre**: Especifica el nombre de la copia de seguridad de la base de datos.
- **Descripción:** Describe la copia de seguridad de la base de datos.
- **Base de datos – completa:** Ejecuta una copia de seguridad completa de la base de datos.
- **Base de datos – diferencial:** Ejecuta una copia de seguridad diferencial de la base de datos. La copia de seguridad diferencial registra sólo los cambios de la información de la base de datos realizados después de la última copia de seguridad completa de la base de datos.
- **Registro de transacciones:** Realiza una copia de seguridad del registro de transacciones.
- **Archivo y grupo de archivos:** Especifica el nombre y grupo de archivos de los que desea realizar una copia de seguridad. Haga clic en el botón Examinar (...) para buscar un archivo o un grupo de archivos.

- **Cinta:** Realiza la copia de seguridad de la base de datos en un dispositivo de cinta.
- **Disco:** Realiza la copia de seguridad de la base de datos en un dispositivo de disco.
- **Agregar:** Agrega un destino de copia de seguridad.
- **Quitar:** Quita el destino seleccionado de copia de seguridad.
- **Contenido:** Presenta el contenido de la cinta o disco seleccionados.
- **Anexar al medio:** Anexa la copia de seguridad a un medio existente. El contenido anterior del medio permanece intacto y la nueva copia de seguridad se escribe a continuación del final de la última copia de seguridad del medio.
- **Sobrescribir medio existente:** Especifica que se sobrescriba cualquier medio existente. La nueva copia de seguridad sobrescribe el contenido existente en el medio de copia de seguridad y, por tanto, éste deja de estar disponible.
- **Programación:** Programa una copia de seguridad de una base de datos. Haga clic en el botón Examinar (...) para ver la programación actual de las copias de seguridad.

6. En el ítem Programación daremos clic en el botón derecho. Utilice este cuadro de diálogo para ver o especificar las siguientes opciones.

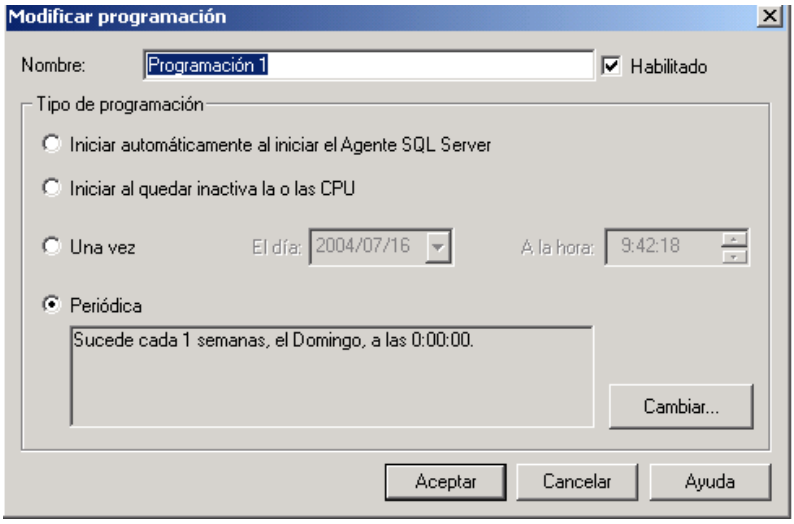


Figura 6.7. Ficha de Modificación de Programación

- **Nombre:** Especifica el nombre de la programación. El nombre está limitado a 128 caracteres. Cada nombre de programación en un trabajo debe ser exclusivo.
- **Enabled:** Habilita la nueva programación de trabajo.
- **Iniciar automáticamente al iniciar el Agente SQL Server:** Inicia automáticamente el trabajo cuando se inicia el Agente de SQL Server.
- **Iniciar al quedar inactiva la o las CPU:** Inicia el trabajo cuando las CPU están inactivas. El tiempo de inactividad de las

CPU se especifica en la ficha Avanzado del cuadro de diálogo Propiedades de Agente SQL Server.

- **Una vez:** Inicia el trabajo una vez en la fecha y hora especificadas.
- **El día:** Especifica la fecha en la que debe iniciarse el trabajo.
- **A la hora:** Especifica la hora a la que debe iniciarse el trabajo.
- **Periódica:** Inicia el trabajo según la programación periódica mostrada.
- **Change:** Muestra el cuadro de diálogo Modificar la programación periódica del trabajo, donde se puede cambiar la programación periódica actual del trabajo.

7. Elegir la opción “Periódica”, damos clic el botón Cambiar para modificar los Periodos de Respaldo. Utilice este cuadro de diálogo para ver o especificar las siguientes opciones.

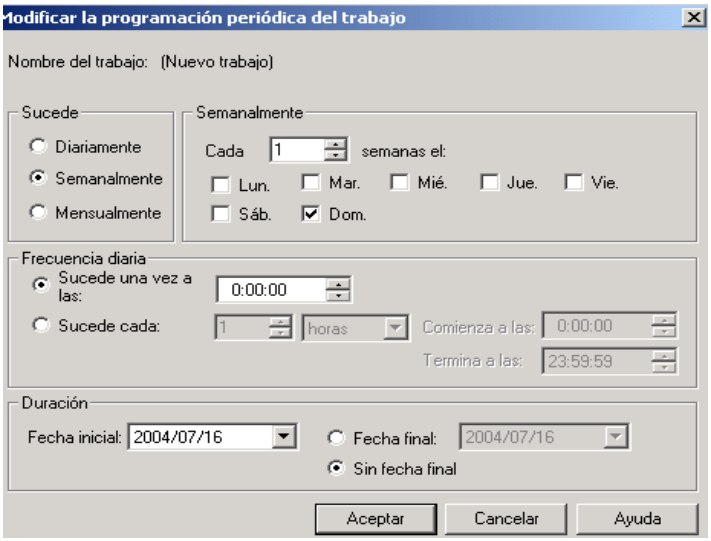


Figura 6.8. Ficha de Modificación Periódica

- **Nombre del trabajo:** Presenta el nombre del trabajo para el que se desea establecer una programación periódica.
- **Diariamente:** Establece una ejecución diaria del trabajo.
- **Semanalmente:** Establece una ejecución semanal del trabajo.
- **Mensualmente:** Establece una ejecución mensual del trabajo.
- **Cada semana:** Especifica la frecuencia del trabajo en incremento de semanas.
- **Lun.** Establece que el trabajo tenga lugar un lunes.
- **Mar.** Establece que el trabajo tenga lugar un martes.
- **Mié.** Establece que el trabajo tenga lugar un miércoles.
- **Jue.** Establece que el trabajo tenga lugar un jueves.
- **Vie.** Establece que el trabajo tenga lugar un viernes.

- **Sáb.** Establece que el trabajo tenga lugar un sábado.
- **Dom.** Establece que el trabajo tenga lugar un domingo.
- **Sucede una vez a las:** Establece la hora para que el trabajo se produzca una vez diariamente.
- **Sucede cada:** Establece el número de horas o minutos entre ejecuciones del trabajo.
- **Comienza a las:** Establece la hora en que se inicia la frecuencia del trabajo cada día.
- **Termina a las:** Establece la hora a la que finaliza la frecuencia del trabajo cada día.
- **Fecha inicial:** Establece la fecha en que comienza a ser efectiva esta programación.
- **Fecha final:** Establece la fecha en que deja de ser efectiva esta programación.
- **Sin fecha final:** Especifica que la programación será efectiva indefinidamente.

8. Dar clic en Aceptar en las dos fichas anteriores para regresar a la ficha principal de backup, Recomendamos configurar de esta forma las dos pantallas anteriores para que exista mayor seguridad en el respaldo de información. El usuario autorizado puede cambiar la configuración según sus necesidades.

9. En la siguiente ficha Opciones describiremos lo siguiente:

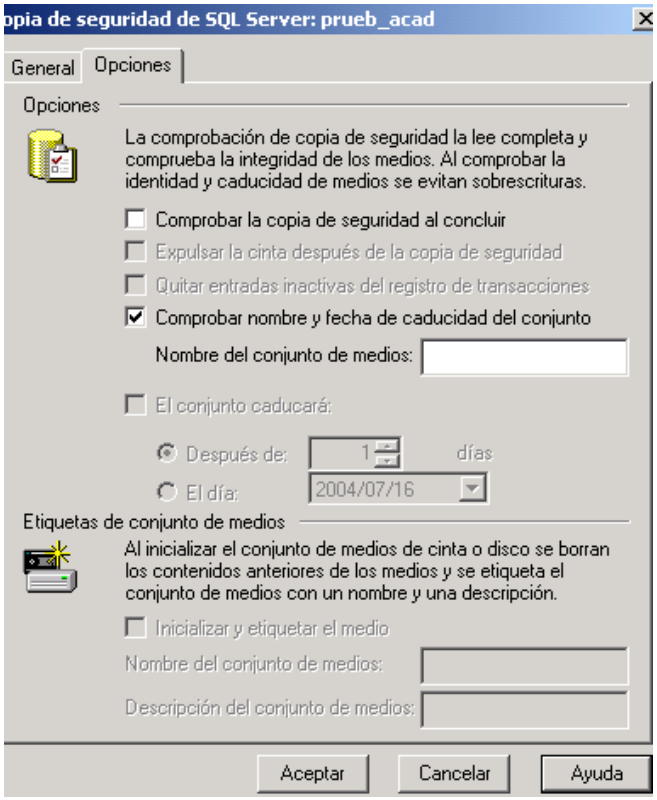


Figura 6.9. Ficha Opciones del Backup (1)

- **Comprobar la copia de seguridad al concluir:** Especifica que la integridad de los medios de copia de seguridad se compruebe al terminar.
- **Expulsar la cinta después de la copia de seguridad:** Expulsa la cinta del medio de copia de seguridad después de que finalice la copia de seguridad.
- **Quitar entradas inactivas del registro de transacciones:** Quita del registro de transacciones todas las entradas de las

transacciones completadas, después de terminar la copia de seguridad.

- **Comprobar nombre y fecha de caducidad del conjunto:** Comprueba el nombre del conjunto de medios y la fecha de caducidad del conjunto de copias de seguridad antes de sobrescribir los medios.
- **Nombre del conjunto de medios:** Especifica el nombre que debe tener el conjunto de medios antes de permitir que se sobrescriba.
- **El conjunto caducará:** Establecer las condiciones de caducidad del conjunto de copias de seguridad. SQL Server 2000 sólo utiliza la información de caducidad de la copia de seguridad del primer conjunto de copias de seguridad del medio para determinar si se puede sobrescribir el medio completo.
- **Después de:** Especifica el número de días que deben pasar, una vez completada la copia de seguridad, antes de que se puedan sobrescribir los datos.
- **El día:** Especifica la fecha en la que se puede sobrescribir el medio.
- **Inicializar y etiquetar el medio:** Escribe el encabezado del formato de cinta de Microsoft (MTF) al principio del medio. Esto borrará todo el contenido y cualquier información de

encabezado anterior del medio. La caducidad del conjunto de copias de seguridad y el nombre del conjunto de medios no se comprueban cuando se inicializa un medio.

- **Nombre del conjunto de medios:** Escribe el nombre del medio que se debe escribir en éste como parte del encabezado MTF.
- **Descripción del conjunto de medios:** Escribe la descripción del medio que se debe escribir en éste como parte del encabezado MTF. Normalmente, suele consistir en el uso que se dará al medio o el lugar en que se almacena.

10. Dar clic en Aceptar, así hemos terminado la configuración de backup en una base de datos especifica, también existen backups de un grupo de base de datos y grupo de servidores.

11. Dar clic en el ítem de Administración que se encuentra en el árbol y luego en Agente SQL Server, con clic derecho damos el estado de Iniciar para que se ejecuten todos los procesos configurados anteriormente.

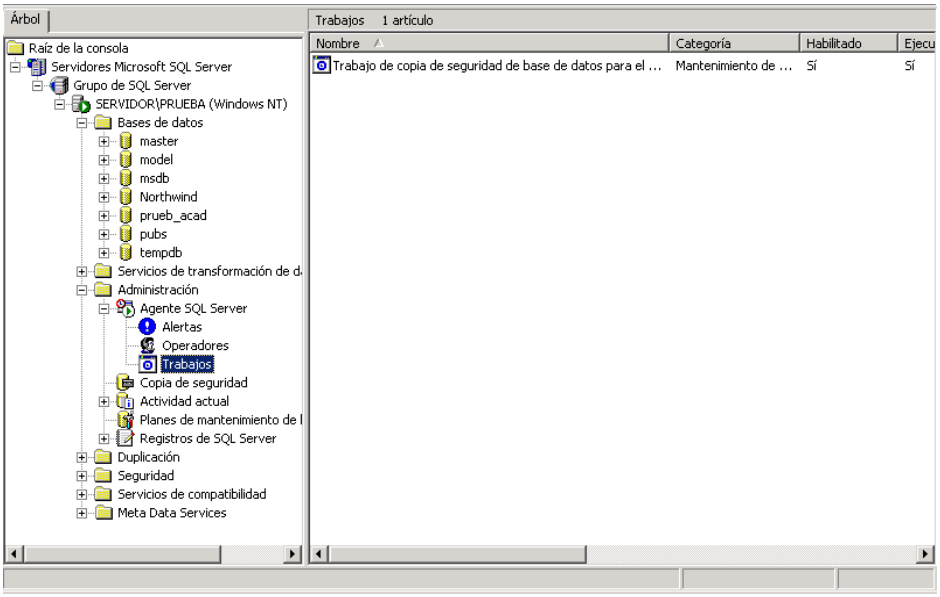


Figura 6.10. Ficha Opciones del Backup (2)

Como podemos ver en la figura, tenemos un único proceso en cola, el del backup, si damos clic derecho e iniciar el proceso, este realizara la copia de seguridad de la base de datos en el acto sin respetar la configuración de periodos realizado anteriormente.

Cabe recalcar que el usuario autorizado estará apto en la configuración de backups dependiendo de las necesidades que este sugiere.

CAPITULO 7

7. Administración de usuarios y accesos al sistema

7.1. Acceso a las pantallas

El usuario ADM_SYS por defecto tiene acceso a las pantallas de administración del sistema. Puede controlar el acceso de los demás usuarios a las diversas pantallas, agrupándolos por perfiles y determinando qué perfiles tienen accesos a todas o una parte de los formularios que integran este sistema.

Figura 7.1 Acceso al menú Administración de Usuarios

7.2. Mantenimiento de perfiles

Cuando por primera vez se ejecuta esta pantalla sale el perfil “S1” que es predeterminado como “Administrador del Sistema” tal como se muestra en la figura 7.2.

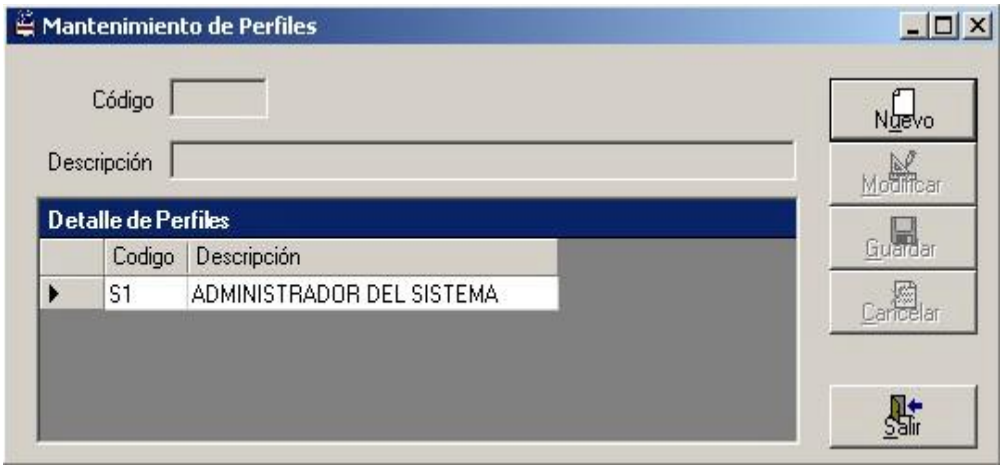


Figura 7.2 Mantenimiento de Perfiles-Pantalla inicial

Para ingresar nuevos perfiles, se presiona el botón “Nuevo”, se ingresa un nuevo código para este perfil y su correspondiente descripción.

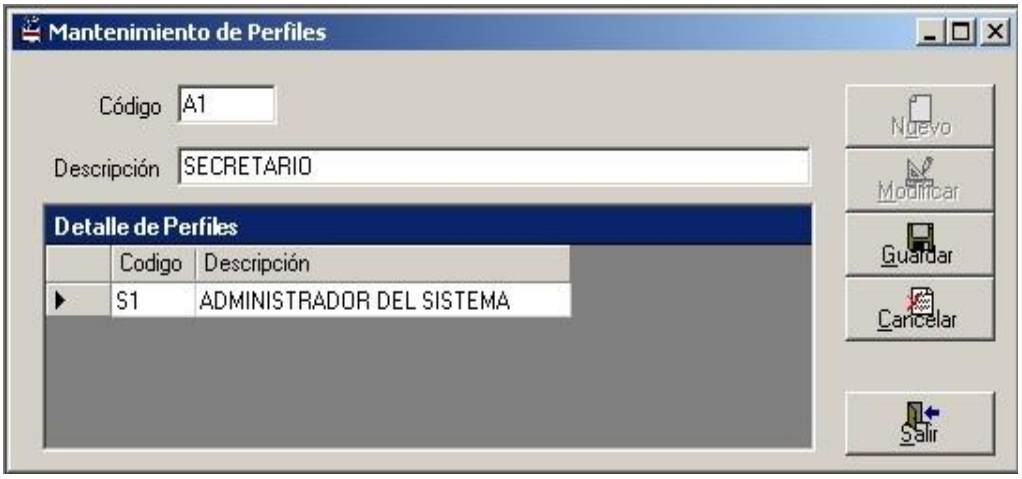


Figura 7.3 Mantenimiento de Perfiles-Nuevo perfil

En este ejemplo hemos ingresado tres perfiles en el sistema.

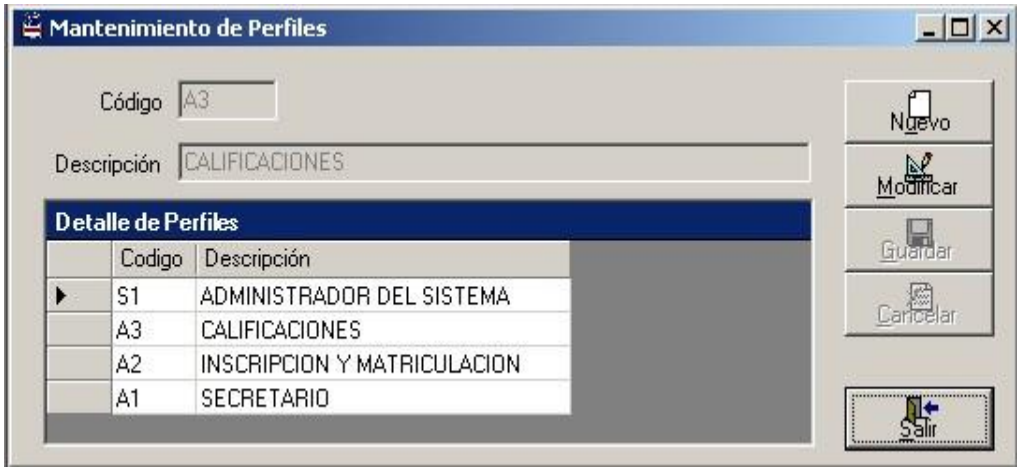


Figura 7.4 Mantenimiento de perfiles-Perfiles creados

7.3. Mantenimiento de pantallas

Esta pantalla sirve para la consulta de las demás pantallas que existen en el sistema, cabe destacar que cada una tiene un código asignado de forma secuencial (ver capítulo 1, en la sección 1.3 Descripción Técnica de las Pantallas)

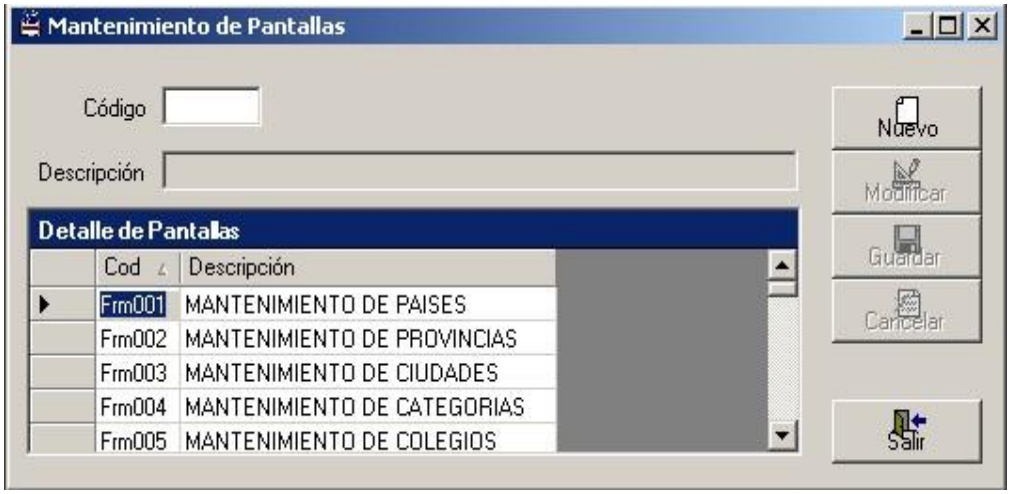


Figura 7.5 Mantenimiento de Pantallas

7.4. Mantenimiento de usuarios

Si se desea ingresar un usuario nuevo se da click sobre el botón "Nuevo", caso contrario, si se desea ver o modificar la información de un usuario existente se ingresa el nombre del mismo y se presiona la tecla Enter.

Cabe destacar que antes de la creación de un nuevo usuario, es importante haber ingresado los demás parámetros básicos del sistema, principalmente las subunidades, porque un usuario debe pertenecer a una subunidad antes de ingresar al sistema.

Figura 7.6 Creación de un nuevo usuario en el sistema

En la figura 7.7 vemos información del administrador del sistema, es el único que no pertenece a ninguna subunidad.

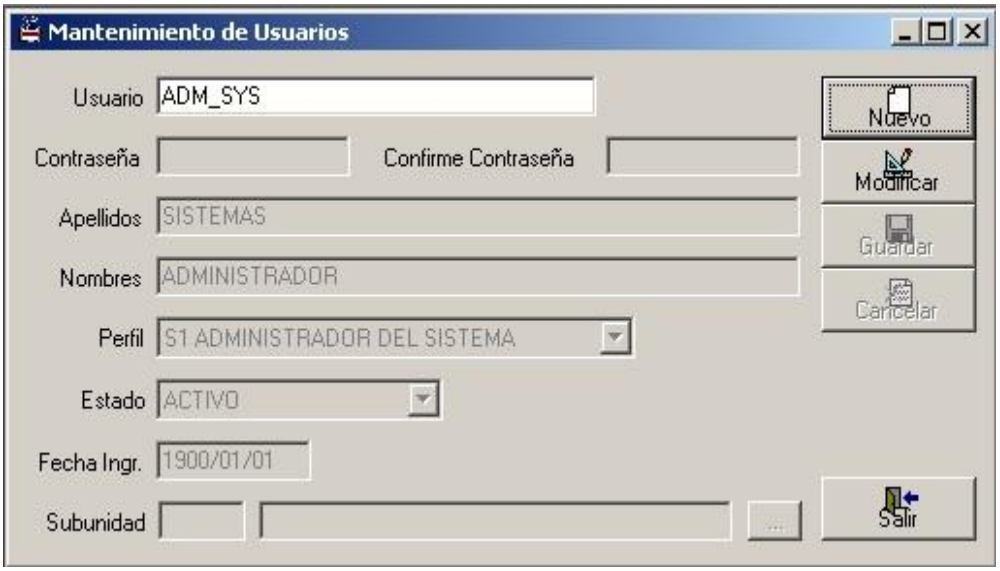


Figura 7.7 Consulta de un usuario existente

7.5. Mantenimiento de permisos

Los permisos sobre cada una de las pantallas son 4:

L – Sólo lectura

LI – Lectura e Impresión

LE – Lectura y Escritura

T – Lectura, Escritura e Impresión

Cabe recalcar que sólo ADM_SYS puede determinar los permisos de los usuarios a las demás pantallas, pero en caso de ser necesario podría también designarse a otro usuario dentro del perfil S1 (Administradores del sistema) y darle privilegios a ese perfil sobre las pantallas de Administración de usuarios.

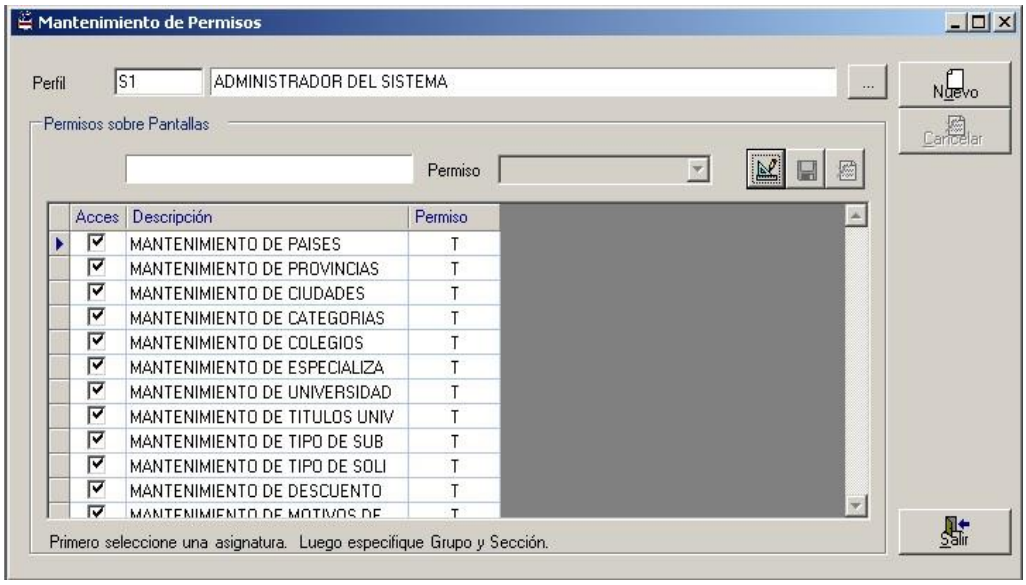


Figura 7.8 Mantenimiento de Permisos

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

Carrera de Ingeniería en Sistemas Computacionales

"Implementación de la Estructura del Sistema Académico con el Módulo de Matriculación para la Facultad de Ciencias Matemáticas y Físicas"

MANUAL DEL USUARIO

Autores:

AYUQUINA CARABAJO MARTHA ELIZABETH
ARELLANO GUZMAN GREGORIO ALEJANDRO
CRUZ OCHOA WILSON
ZAMBRANO MOLINA DARWIN FLORENCIO

GUAYAQUIL – ECUADOR

Año: 2004

INDICE GENERAL

INTRODUCCION 1

CAPITULO 1

1. ACCESO AL SISTEMA 2

1.1. Pantalla de acceso 2

1.2. Menú Principal 4

CAPITULO 2

2. PARAMETROS GENERALES 6

2.1. Mantenimiento de Países 6

2.2. Mantenimiento de Provincias 10

2.3. Mantenimiento de Ciudades 13

2.4. Mantenimiento de Categoría de Estudiantes 17

2.5. Mantenimiento de Colegios 19

2.6. Mantenimiento de Especializaciones 23

2.7. Mantenimiento de Universidades 24

2.8. Mantenimiento de Títulos Universitarios 29

2.9. Mantenimiento de Tipos de Subunidad 31

2.10. Mantenimiento de Tipos de Solicitudes 32

2.11. Mantenimiento de Descuento y Aranceles 34

2.12. Mantenimiento de Motivos de Cambio de Pénsum 39

2.13. Mantenimiento de Tipos de Proyectos 40

2.14. Mantenimiento de Bancos 42

2.15. Mantenimiento de Concepto de Cobros 45

2.16. Mantenimiento Fichas Docentes 47

2.17. Mantenimiento de la Ficha Estudiantil 55

2.18. Mantenimiento de Facultades 65

2.19. Mantenimiento de Subunidades 70

CAPITULO 3

3. PARAMETROS POR SUBUNIDAD 77

3.1. Mantenimiento de Áreas Académicas 77

3.2. Mantenimiento de Valores de la Matrícula Global 79

3. 3. Mantenimiento de Valores de la Matrícula por Asignatura 81

3. 4. Mantenimiento de Incremento de Matrículas 83

3. 5. Mantenimiento de Tipos de Cursos 84

3. 6. Mantenimiento de Tipos de Calificaciones 89

CAPITULO 4

4. CONTROL ACADEMICO 92

4.1. Mantenimiento de Asignatura 92

4.2. Mantenimiento de Pénsum 96

4.3. Mantenimiento de Prerrequisitos 99

CAPITULO 5

5. PERIODOS LECTIVOS 104

5.1. Creación de Períodos Lectivos 104

5.2. Cronograma de Entrega de Notas Parciales 110

CAPITULO 6

6. CONTROL DE HORARIOS 111

6.1. Distributivo de Asignaturas a Docentes 111

6.2. Horarios por Grupo 117

CAPITULO 7

7. INSCRIPCION Y MATRICULACION 123

7.1. Inscripción de Estudiante a Subunidad 123

7.2. Inscripción de Estudiante a una Asignatura 129

7.3. Orden de Pago 134

7.4. Cuotas 139

7.5. Cambio de Categoría 143

7.6. Cambio de Pénsum 144

7.7. Registro de Ordenes Canceladas 145

CAPITULO 8

8. REGISTRO DE CALIFICACIONES 148

8.1. Calificaciones Parciales 148

8.2. Promedio de Calificaciones 152

CAPITULO 9

9. REGISTRO DE SOLICITUDES 155

9.1. Solicitudes Especiales 155

9.2. Recalificación de Exámenes 158

CAPITULO 10

10. REPORTES DEL SISTEMA ACADEMICO 162

10.1. Informe de Estudiantes por grupo 162

10.2. Informe de Inscripciones por Estudiante 165

10.3. Consulta de Docentes por Subunidad 168

10.4. Reporte de Horarios por Grupos 170

10.5. Reporte de Horarios por Docentes 172

10.6. Resumen de Descuentos y Aranceles por Período 174

10.7. Reporte de Estudiantes Matriculados 175

10.8. Reporte de Estadísticas de Inscritos 176

10.9. Informe de Calificaciones por Asignatura 179

10.10. Resumen de Calificaciones por Estudiante 181

10.11. Detalle de Calificaciones Parciales 183

CAPITULO 11

11. SEGURIDAD

11.1. Cambio de Usuario 185

11.2. Cambio de Contraseña 186

CAPITULO 12

12. AYUDA 187

12.1. Ayuda del Sistema Académico 187

12.2. Ejemplos de la Ayuda del Sistema Académico 188

INDICE DE FIGURAS

Figura 1.1 Pantalla de Acceso al Sistema 2

Figura 1.2. Presentación del Sistema Académico 3

Figura 1.3. Menú Principal 4

Figura 2.1. Acceso a Mantenimiento de Países 6

Figura 2.2. Pantalla de Mantenimiento de Países 7

Figura 2.3. Ejemplo de Mantenimiento de Países 10

Figura 2.4. Pantalla de Mantenimiento de Provincias 11

Figura 2.5. Mantenimiento de Provincias 13

Figura 2.6. Pantalla de Mantenimiento de Ciudades 13

Figura 2.7.Consulta de Ciudades 14

Figura 2.8. Ejemplo de Consulta de Ciudades 15

Figura 2.9. Ejemplo de Modificar Ciudades 16

Figura 2.10. Pantalla de Mantenimiento de Categorías 17

Figura 2.11. Ejemplo de Mantenimiento de Categorías 18

Figura 2.12. Pantalla de Mantenimiento de Colegios 19

Figura 2.13. Ejemplo de Mantenimiento de Colegio-Paso1 20

Figura 2.14. Ejemplo de Mantenimiento de Colegios-Paso2 21

Figura 2.15. Ejemplo de Mantenimiento de Colegios-Paso3 21

Figura 2.16. Ejemplo de Consulta de Colegios 22

Figura 2.17. Pantalla de Mantenimiento de Especializaciones del Bachillerato 23

Figura 2.18. Ejemplo de Mantenimiento de Especializaciones 24

Figura 2.19. Pantalla de Mantenimiento de Universidades 25

Figura 2.20. Ejemplo de Mantenimiento de Universidades (1) 26

Figura 2.21.Pantalla de Mantenimiento de Universidades (2) 26

Figura 2.22. Ejemplo de Mantenimiento de Universidades (2) 27

Figura 2.23. Pantalla de Consulta de Universidades 28

Figura 2.24. Ejemplo de Consulta de Universidades 29

Figura 2.25. Pantalla de Mantenimiento de Títulos Universitarios 29

Figura 2.26. Ejemplo de Mantenimiento de Títulos Universitarios 30

Figura 2.27. Pantalla de Mantenimiento de Tipos de Subunidad 31

Figura 2.28. Ejemplo de Mantenimiento de Tipos de Subunidad 32

Figura 2.29. Pantalla de Mantenimiento de Tipo de Solicitudes 32

Figura 2.30. Ejemplo de Mantenimiento de Tipo de Solicitudes 33

Figura 2.31. Pantalla de Mantenimiento de Descuento y Aranceles 34

Figura 2.32. Pantalla de Consulta de Descuentos y Aranceles 35

Figura 2.33. Ejemplo de Consulta de Descuentos y Aranceles 36
Figura 2.34. Ejemplo de Motivo de Cambio de Pénsum 40
Figura 2.35. Pantalla de Tipo de Proyectos de Graduación 41
Figura 2.36. Ejemplo de Tipos de Proyectos de Graduación 42
Figura 2.37. Pantalla de Mantenimiento de Bancos 42
Figura 2.38. Ejemplo de Bancos 43
Figura 2.39. Consulta de Bancos 44
Figura 2.40. Ejemplo de Consulta de Bancos 45
Figura 2.41. Pantalla de Concepto de Cobros 46
Figura 2.42. Ejemplo de Conceptos de Cobros 47
Figura 2.43. Acceso de Mantenimiento de Fichas Docentes 47
Figura 2.44. Pantalla de la Ficha Docente (Datos Personales) 48
Figura 2.45. Ejemplo de la Ficha Docente (Datos Personales) 49
Figura 2.46. Pantalla de la Ficha Docente (Datos Generales) 49
Figura 2.47. Ejemplo de la Ficha Docente (Datos Generales) 50
Figura 2.48. Pantalla de la Ficha Docente (Domicilio) 51
Figura 2.49. Ejemplo de la Ficha Docente (Domicilio) 52
Figura 2.50. Pantalla de la Ficha Docente (Trabajo) 52
Figura 2.51. Ejemplo de la Ficha Docente (Trabajo) 53
Figura 2.52. Pantalla de la Ficha Docente (Estudios Realizados) 54
Figura 2.53. Ejemplo de la Ficha Docente (Estudios Realizados) 55
Figura 2.54. Pantalla de la Ficha Estudiantil 55

Figura 2.55. Ejemplo de Ficha Estudiantil- Datos Personales 57

Figura 2.56. Ficha Estudiantil-Datos Personales (2) 57

Figura 2.57.Ejemplo Ficha Estudiantil-Datos Personales (2) 58

Figura 2.58. Ficha Estudiantil- Datos Generales 59

Figura 2.59. Ejemplo de Ficha Estudiantil- Datos Generales 60

Figura 2.60. Ficha Estudiantil- Domicilio 61

Figura 2.61. Ejemplo de Ficha Estudiantil- Domicilio 62

Figura 2.62. Ficha Estudiantil- Trabajo 62

Figura 2.63. Ejemplo de Ficha Estudiantil- Trabajo 63

Figura 2.64. Ficha Estudiantil- Estudios 64

Figura 2.65. Ejemplo de Ficha Estudiantil- Estudios 65

Figura 2.66. Pantalla de Mantenimiento de Facultades 66

Figura 2.67. Ejemplo de Mantenimiento de Facultades 67

Figura 2.68. Pantalla de Consulta de Docentes 68

Figura 2.69. Ejemplo de Consulta de Docentes 69

Figura 2.70. Ejemplo de Consulta de Facultades 70

Figura 2.71. Pantalla de Mantenimiento (1) de Subunidades Académicas 71

Figura 2.72. Ejemplo de Mantenimiento (1) de Subunidades Académicas 72

Figura 2.73. Pantalla de Mantenimiento (2) de Subunidades Académicas 73

Figura 2.74. Ejemplo de Mantenimiento (2) de Subunidades Académicas 74
Figura 2.75. Ejemplo de Consulta de Facultades (Subunidad) 75
Figura 2.76. Ejemplo de Consulta de Subunidades Académicas 76
Figura 3.1. Pantalla de Áreas Académicas 77
Figura 3.2. Ejemplo de Áreas Académicas 79
Figura 3.3. Acceso a Valores de la Matrícula Global 79
Figura 3.4. Pantalla de Valores de la Matrícula Global 80
Figura 3.5. Ejemplo de Valores de la Matrícula Global 81
Figura 3.6. Pantalla de Valores de la Matrícula por Asignatura 81
Figura 3.7. Ejemplo de Valores de la Matrícula por Asignatura 82
Figura 3.8. Pantalla de Incremento de Matrículas 83
Figura 3.9. Ejemplo de Incremento de Matrículas 84
Figura 3.10. Pantalla de Tipos de Cursos 85
Figura 3.11. Ejemplo (1) de Tipos de Cursos 87
Figura 3.12. Ejemplo (2) de Tipos de Cursos 88
Figura 3.13. Consulta de Tipo de Cursos 88
Figura 3.14. Ejemplo de Consulta de Tipos de Cursos 89
Figura 3.15. Pantalla de Tipo de Calificaciones 90
Figura 3.16. Ejemplo de Tipo de Calificaciones 91
Figura 4.1. Acceso a Mantenimiento de Asignaturas 92
Figura 4.2. Pantalla de Mantenimiento de Asignaturas 93

Figura 4.3. Ejemplo de Asignaturas 95

Figura 4.4 Consulta de Asignaturas 95

Figura 4.5. Ejemplo de Consulta de Asignaturas 96

Figura 4.6. Pantalla de Mantenimiento de Pénsum 97

Figura 4.7. Ejemplo de Mantenimiento de Pénsum 98

Figura 4.8. Ejemplo de Consulta de Pénsum 99

Figura 4.9. Pantalla de Mantenimiento de Prerrequisitos 100

Figura 4.10. Ejemplo de Prerrequisito- Paso 1 100

Figura 4.11. Seleccionando un Prerrequisito-Paso 2 101

Figura 4.12. Ejemplo de Prerrequisito- Paso 3 102

Figura 4.13. Consulta de Prerrequisito 102

Figura 4.14. Ejemplo de Consulta de Prerrequisito 103

Figura 5.1. Acceso a Períodos Lectivos 104

Figura 5.2. Creación de Períodos Lectivos 105

Figura 5.3. Ejemplo de Creación de Períodos Lectivos- Paso 1 105

Figura 5.4. Ejemplo de Creación de Períodos Lectivos- Paso 2 106

Figura 5.5. Ejemplo de Creación de Períodos Lectivos- Paso 3 106

Figura 5.6. Ejemplo de Modificación de Períodos Lectivos 107

Figura 5.7. Ejemplo de Modificación de Períodos Lectivos- Mensaje 108

Figura 5.8. Objeto Calendario 108

Figura 5.9. Cambio de Meses en Objeto Calendario 109

Figura 5.10. Cambio de Año en Objeto Calendario 109

Figura 5.11. Cronograma de Entrega de Notas Parciales 110

Figura 5.12. Ejemplo de Cronograma de Entrega de Notas Parciales 110

Figura 6.1. Acceso a Control de Horarios 111

Figura 6.2. Período Lectivo- Paso 1 114

Figura 6.3. Docente- Paso 2 114

Figura 6.4. Asignaturas y Apertura de Grupos- Paso 3 115

Figura 6.5. Distributivo de Asignaturas a Docentes 115

Figura 6.6. Ejemplo de Eliminación un Registro 116

Figura 6.7. Horarios por Grupo-Paso1 117

Figura 6.8. Horarios por Grupo-Paso2 118

Figura 6.9. Horarios por Grupo-Paso 3 119

Figura 6.10. Modificar Hora-Clase (Paso 1) 120

Figura 6.11. Modificar Hora-Clase (Paso2) 121

Figura 6.12. Corrección de Hora-Clase (Paso 3) 122

Figura 6.13. Ejemplo de Horarios por Grupo 122

Figura 7.1. Acceso a Inscripción de Estudiante a Subunidad 123

Figura 7.2. Registro de Estudiantes en la Unidad Académica 125

Figura 7.3. Ejemplo del Registro de Estudiantes en la Unidad Académica 126

Figura 7.4. Consultando Registro de Estudiante en una 127

Subunidad

Figura 7.5. Modificando Registro de Estudiante en una Subunidad Académica 128

Figura 7.6. Inscripción de Asignaturas por Estudiante 129

Figura 7.7. Inscripción de Asignaturas por Estudiante-Paso 1 131

Figura 7.8. Inscripción de Asignaturas por Estudiante- Paso 2 131

Figura 7.9. Inscripción de Asignaturas por Estudiante- Paso 3 132

Figura 7.10. Inscripción de Asignaturas por Estudiante- Paso 4 132

Figura 7.11. Inscripción de Asignaturas por Estudiante- Paso 5 133

Figura 7.12. Orden de Pago 134

Figura 7.13. Orden de Pago- Paso 1 135

Figura 7.14. Orden de Pago- Paso 2 135

Figura 7.15. Orden de Pago- Paso 3 136

Figura 7.16. Orden de Pago- Paso 4 137

Figura 7.17. Orden de Pago- Paso 5 138

Figura 7.18. Cuotas- Paso 1 139

Figura 7.19. Cuotas- Paso 2 140

Figura 7.20. Cuotas- Paso 3 141

Figura 7.21. Cuotas- Paso 4 141

Figura 7.22. Cuotas- Paso 5 142

Figura 7.23. Cambio de Categoría 143

Figura 7.24. Ejemplo de Cambio de Categoría- Paso 1 144

Figura 7.25. Ejemplo de Cambio de categoría- Paso 2 144
Figura 7.26. Cambio de Pénsum 144
Figura 7.27. Ejemplo de Cambio de Pénsum 145
Figura 7.28. Registro de Ordenes Canceladas 146
Figura 7.29. Ejemplo de Registro de Ordenes Canceladas 147
Figura 8.1. Acceso a Registro de Calificaciones 148
Figura 8.2. Ingreso de Calificación Parcial 148
Figura 8.3. Ingreso de Calificación Parcial- Paso 1 149
Figura 8.4. Ingreso de Calificación Parcial- Paso 2 150
Figura 8.5. Ingreso de Calificación Parcial- Paso 3 150
Figura 8.6. Ingreso de Calificación Parcial- Paso 4 151
Figura 8.7. Ejemplo de Ingreso de Calificación Parcial 152
Figura 8.8. Cálculo de Promedios 153
Figura 8.9. Ejemplo de Cálculo de Promedios 153
Figura 9.1. Acceso a Solicitudes 155
Figura 9.2. Solicitudes Especiales 155
Figura 9.3. Solicitudes Especiales- Paso 1 156
Figura 9.4. Solicitudes Especiales- Paso 2 157
Figura 9.5. Solicitudes Especiales- Paso 3 157
Figura 9.6. Solicitudes Especiales- Paso 4 157
Figura 9.7. Recalificación 158
Figura 9.8. Recalificación- Paso 1 159

Figura 9.9. Recalificación- Paso2 160

Figura 9.10. Recalificación- Paso 3 160

Figura 9.11. Recalificación- Paso 4 161

Figura 10.1. Informe de Estudiantes por Grupo 162

Figura 10.2. Informe de Estudiantes por Grupo-Paso 1 163

Figura 10.3. Informe de Estudiantes por Grupo-Paso 2 164

Figura 10.4. Ejemplo del Informe de Estudiantes por Grupo 164

Figura 10.5. Informe de Inscripciones por Estudiante 165

Figura 10.6. Informe de Inscripciones por Estudiante- Paso 1 166

Figura 10.7. Informe de Inscripciones por Estudiante- Paso2 166

Figura 10.8. Ejemplo de Informe de Inscripciones por Estudiante 167

Figura 10.9. Informe de Inscripciones por Estudiante- Opción Nivel 167

Figura 10.10. Consulta de Docentes por Subunidad 168

Figura 10.11. Ejemplo del Listado de Docentes por Áreas Académicas 169

Figura 10.12. Acceso al Reporte de Horarios 170

Figura 10.13. Reporte de Horarios por Grupos 170

Figura 10.14. Reporte de Horarios por Grupos- Paso 1 171

Figura 10.15. Repote de Horarios por Grupos- Paso 2 171

Figura 10.16. Ejemplo deL Registro de Horario por Grupos 172

Figura 10.17. Reporte de Horarios por Docentes 172

Figura 10.18. Reporte de Horarios por Docentes- Paso 1 173
Figura 10.19. Ejemplo de Registro de Horarios por Docentes 173
Figura 10.20. Resumen de Descuentos y Aranceles 174
Figura 10.21. Informe de Descuentos y Aranceles 175
Figura 10.22. Estudiantes Matriculados 175
Figura 10.23. Listado de Estudiantes Matriculados 176
Figura 10.24. Estadísticas de Inscritos 176
Figura 10.25. Estadísticas de Estudiantes Inscritos por Nacionalidad 177
Figura 10.26. Estadísticas de Estudiantes Inscritos por Categoría de Colegio 178
Figura 10.27. Estadísticas de Estudiantes Inscritos por Edad 178
Figura 10.28. Estadísticas de Estudiantes Inscritos por Sexo 179
Figura 10.29. Informe de Calificaciones por Asignatura 179
Figura 10.30. Informe de Calificaciones por Asignatura- Paso1 180
Figura 10.31. Informe de Calificaciones por Asignatura- Paso 2 180
Figura 10.32. Listado de Calificaciones por Asignatura 181
Figura 10.33. Resumen de Calificaciones por Estudiante 181
Figura 10.34. Resumen de Calificaciones por Estudiante- Paso1 182
Figura 10.35. Resumen de Calificaciones por Estudiante- Paso2 182
Figura 10.36. Ejemplo de Resumen de Calificaciones por Estudiante 183

Figura 10.37. Pantalla de Informe de Calificaciones Parciales 183

Figura 10.38. Ejemplo de Detalle de Calificaciones por Parcial 184

Figura 11.1. Cambio de Usuario 185

Figura 11.2. Cambio de Contraseña 186

Figura 12.1. Acceso a la Ayuda del Sistema Académico 187

Figura 12.2. Ejemplo del uso de la Ayuda (1) 189

Figura 12.3. Ejemplo del uso de la Ayuda (2) 190

Figura 12.4. Ejemplo del uso de la Ayuda (3) 191

INTRODUCCION

El presente manual detalla la forma de manejar el Sistema en todas las opciones que dispone y controla el mismo; además, se especifica claramente la funcionalidad de los botones, checkboxes (casillas de verificación), objetos de listas y cómo acceder y salir en las diferentes opciones que brinda el menú principal.

El manual está realizado para usuarios que están familiarizados con el ambiente Windows, es decir, que sabe emplear correctamente el ratón, iconos y todo aquello que es peculiar en un ambiente gráfico de un sistema.

Este manual ha sido desarrollado con el fin de facilitar el manejo de las opciones del Sistema Académico para los diferentes usuarios que tendrán acceso al mismo.

CAPITULO 1

1. ACCESO AL SISTEMA

1.1. Pantalla de acceso

Cuando arranca el sistema aparece una pantalla de acceso, en la cual se escribirá su usuario y su contraseña. Dependiendo del cargo de ese usuario se van habilitar ciertos ítems en el menú, por ejemplo existe el cargo de matriculación, de consulta de notas, de ingreso de datos personales de estudiantes y docentes, de inscripción de estudiantes a las subunidades académicas, etc.

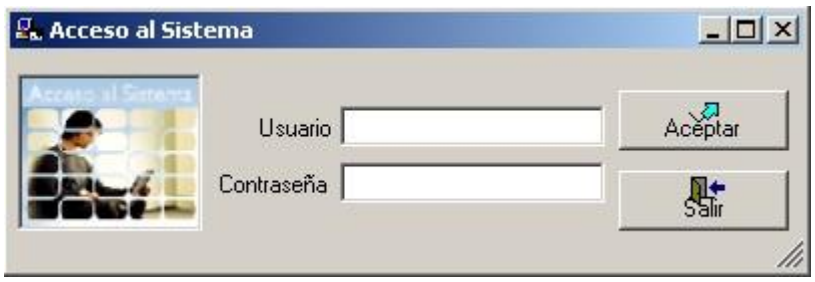


Figura 1.1 Pantalla de Acceso al Sistema

El usuario que tenga privilegios de administrador se encargará de llenar los parámetros básicos del sistema y otorgar permisos sobre el uso de las pantallas que se detallarán más adelante.

La forma de navegar dentro de un formulario se lo puede realizar ya sea con el botón TAB, con el botón ENTER o con el Mouse.

Luego de hacer un clic o enter sobre el botón Aceptar aparecerá en primera instancia la pantalla de presentación del Sistema Académico y luego aparece la pantalla del menú principal del sistema que tendrá como fondo una imagen de la facultad de Ciencias Matemáticas y Físicas.

Figura 1.2. Presentación del Sistema Académico

1.2. Menú Principal

Figura 1.3. Menú Principal

El menú principal constará de los siguientes ítems:

- Mantenimientos de parámetros y fichas personales
- Registro de asignaturas y pénsum
- Cargas horarias y horarios de grupos
- Inscripciones de estudiantes
- Matriculaciones
- Registro de notas y cálculo de promedios
- Registro de solicitudes de estudiantes
- Reportes

Hay que acotar que el administrador del sistema tendrá sólo acceso a las pantallas de mantenimiento de parámetros que van a servir para llenar las tablas tanto de estudiantes como de docentes.

En la barra de estado se observa el usuario con quien ingresó al sistema y la fecha actual. Adicionalmente se observará en el centro de la misma la descripción de la pantalla que se invoque y su nombre técnico.

CAPITULO 2

2. PARAMETROS GENERALES

2.1. Mantenimiento de Países

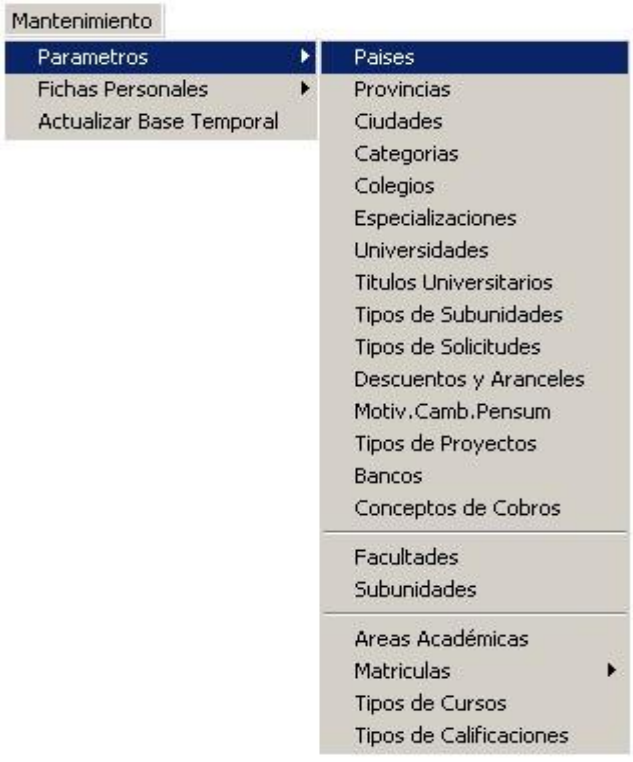


Figura 2.1. Acceso a Mantenimiento de Países

En este formulario el usuario realiza el mantenimiento y detalle de los países. En esta pantalla se observan tres cajas de texto y un grid de consulta.

En lo que se refiere a las cajas de texto, dependiendo de lo que uno solicite, aceptará sólo letras, sólo números o caracteres especiales. Por ejemplo en la caja de texto Código no puede aceptar por ningún motivo letras o símbolos, ya que por lo general los códigos son numéricos.

Con respecto al mantenimiento de países tenemos:

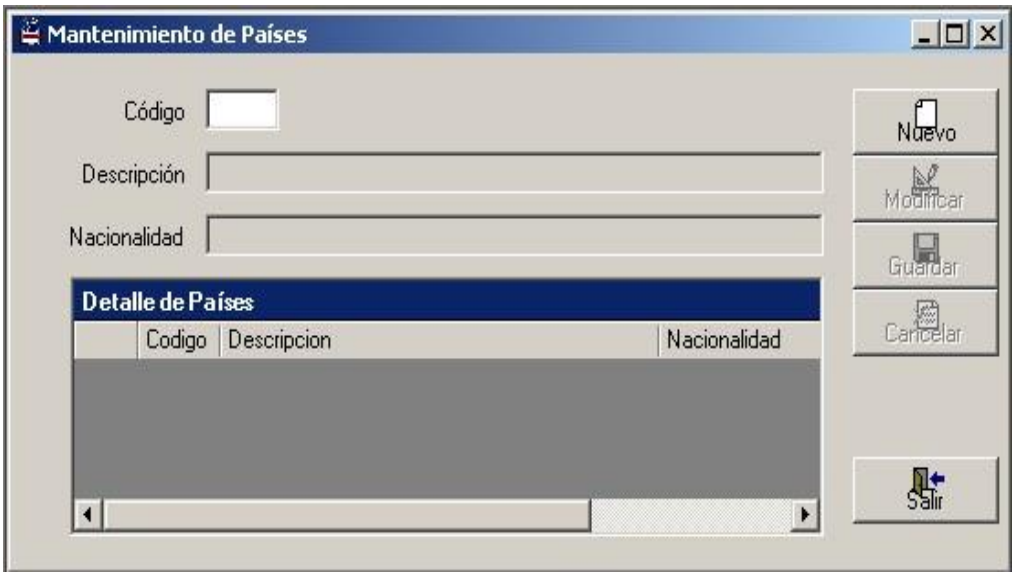


Figura 2.2. Pantalla de Mantenimiento de Países

- Una caja de texto llamada “Código” en la cual se digitará el código del país (se habilita para hacer la búsqueda de un país).
- Una caja de texto llamada “Descripción” en la cual se digitará el nombre del país.
- Una caja de texto llamada “Nacionalidad” en la cual se digitará la nacionalidad correspondiente de cada país.
- Un “Grid de consulta” donde aparece todos los países guardados en ese momento.

En la parte derecha del formulario existen 5 botones, de las cuales las 4 primeras son botones principales para el mantenimiento de este formulario y un botón al final “Salir”.

Los botones que son significativos para el formulario de mantenimiento de países, que van a estar por lo general también en la mayoría de las formas de mantenimiento, son las siguientes:

- Un botón “Nuevo” el cual se presiona cuando se quiere ingresar nuevos datos a las cajas de textos.
- Un botón “Modificar” el cual se presiona cuando se quiere cambiar datos a las cajas de textos.
- Un botón “Guardar” el cual se presiona cuando se quiere guardar los datos de la forma.

- Un botón “Cancelar” el cual se presiona cuando los datos de las cajas de texto están incorrectos.
- El botón “ Salir“ se presiona cuando el usuario desea salir de la forma.

Referente al mantenimiento, cuando hacemos clic al botón Nuevo se deshabilita la caja de texto Código, y al mismo tiempo se habilita tanto la caja de texto de Descripción y la de Nacionalidad, y a su vez se habilitan los botones Guardar y Cancelar, debido a que cuando se crea un nuevo registro tiene que guardarse en la base de datos.

Una vez puesta toda la información requerida hacemos clic sobre el botón Guardar o sencillamente damos Enter.

De ahí aparece en Detalle de Países (Grid de Consulta) toda la información creada, habilitándose los botones Nuevo y Modificar y los botones deshabilitados Guardar y Cancelar.

Al habilitarse el botón Nuevo significa que se deseará ingresar otro registro, de igual manera el botón Modificar si se olvidó de agregar algo en las dos cajas de texto: Descripción o Nacionalidad, o sino si ha habido algún error en escribir la información.

Figura 2.3. Ejemplo de Mantenimiento de Países

Con respecto a la caja de texto Código, tenemos que hacer una observación, éste sirve básicamente para la búsqueda de un cierto país si se sabe el código.

Otras observaciones que hay que considerar:

- En el detalle de consulta siempre va a salir la información en orden alfabético.
- Nunca se puede modificar un código, ya que se considera una clave importante en esta tabla de Países.

2.2. Mantenimiento de Provincias

En esta forma el usuario puede crear y modificar provincias.

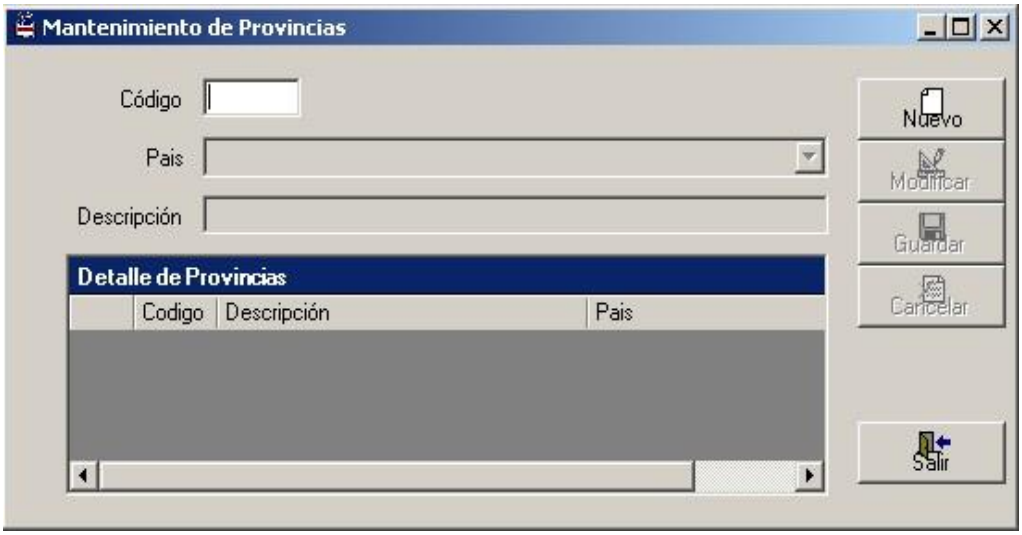


Figura 2.4. Pantalla de Mantenimiento de Provincias

En esta pantalla de provincias aparece:

- Una caja de texto llamada “Código” en la cual se digitará el código de la provincia (se habilita para hacer la búsqueda de una provincia).
- Un combo llamado “País” en la cual se seleccionará el nombre de los países ingresados.
- Una caja de texto llamada “Descripción” en la cual se digitará el nombre de la provincia del país.
- Un “Grid de Consulta” donde aparecerá todas las provincias ingresados en el sistema.

Cuando hacemos clic sobre el botón Nuevo se habilitará el combo “Países” donde saldrá una lista de los países ingresados

anteriormente en el sistema con sus respectivos códigos. Con respecto a la caja de texto “Descripción” también queda habilitado para poner el nombre de la provincia de donde proviene ya sea la Universidad, docentes o estudiantes.

También en este caso quedan habilitado los botones Guardar y Cancelar, ya que como se dijo anteriormente éstos deben ser guardados en una base de datos cuando se crea un nuevo registro.

Al poner toda la información que se requiere se da un Enter o un clic sobre el botón Guardar.

Inmediatamente saldrá en el Grid de Consulta toda la información que se haya creado en ese momento, y se habilitan tanto los botones Nuevo como Modificar en el caso de ingresar un registro o corregir alguna información errada respectivamente.

Figura 2.5. Mantenimiento de Provincias

2.3. Mantenimiento de Ciudades

En este formulario el usuario se efectúa el mantenimiento y la consulta de ciudades.

Con referente al mantenimiento tenemos:

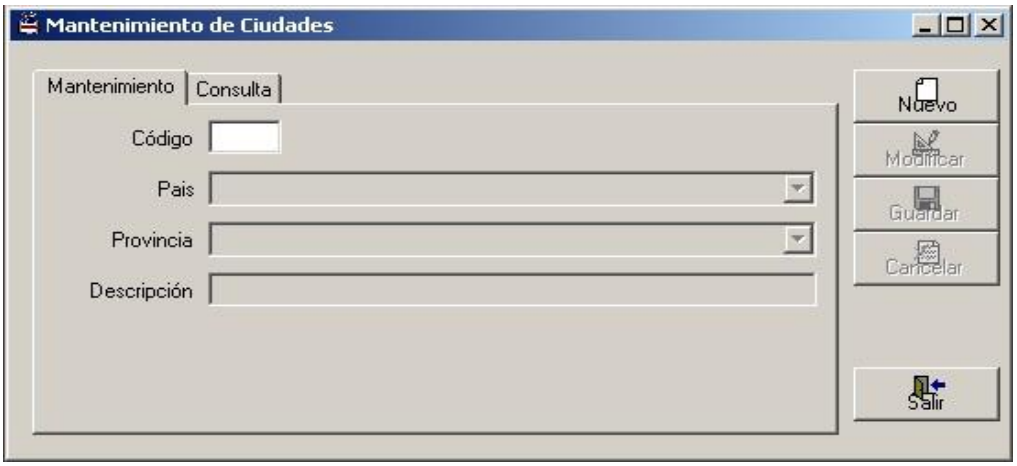


Figura 2.6. Pantalla de Mantenimiento de Ciudades

- Una caja de texto llamada “Código” en la cual se digitará el código de la ciudad (se habilita para la búsqueda de una ciudad).
- Un combo llamado “País” en la cual se seleccionará el nombre de los países ingresados.
- Un combo llamado “Provincia” en la cual se seleccionará el nombre de la provincia del país.
- Una caja de texto llamada “Descripción” donde se digitará el nombre de la ciudad.

En lo que se refiere a la consulta se tiene lo siguiente:

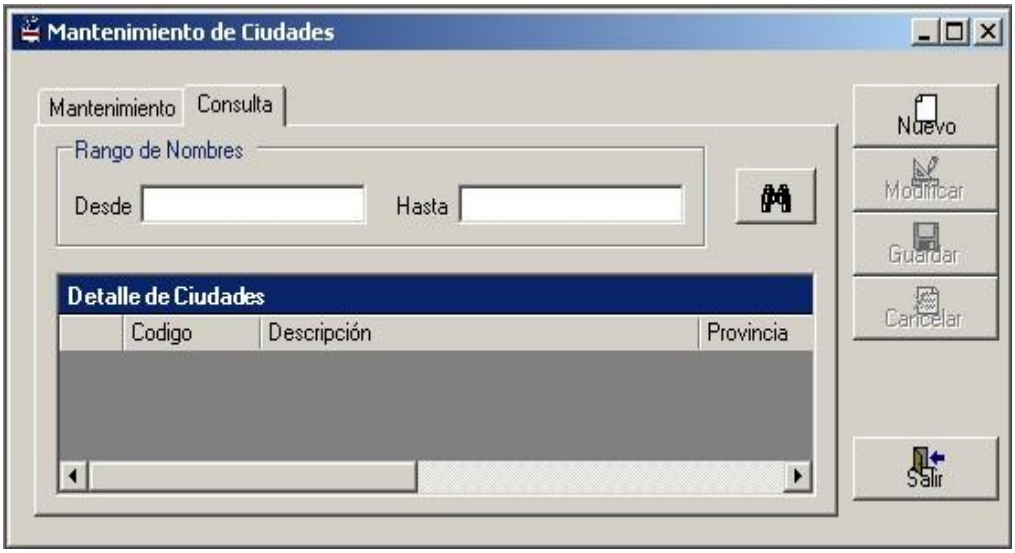


Figura 2.7.Consulta de Ciudades

- Un “Rango de Nombres” que contiene dos cajas de textos: uno de inicio y uno final para la búsqueda masiva de información deseada.
- Un “Grid de Consulta” donde aparecerá toda la información que se haya ingresado en el sistema o la información que se desea buscar cuando se presiona el botón. .

En las pantallas anteriores hemos dado ejemplos sobre cómo se hace el ingreso de un nuevo registro, en esta ocasión nos limitaremos a dar un ejemplo de cómo corregir o modificar alguna información errónea.

En esta pantalla de consulta de ciudades se observa que se ha repetido dos veces el nombre de una misma ciudad.

Figura 2.8. Ejemplo de Consulta de Ciudades

Para poder corregir dicha información se lo puede realizar a través de dos maneras:

- Se da un doble clic sobre el registro que se va a realizar cambios y automáticamente nos ubicamos en la ficha Mantenimiento, damos clic en el botón Modificar y hacemos la corrección respectiva.
- Se ve el código del registro que se quiere cambiar, nos dirigimos hacia mantenimiento y digitamos dicho código, presionamos Enter y a continuación se da un clic en el botón Modificar.

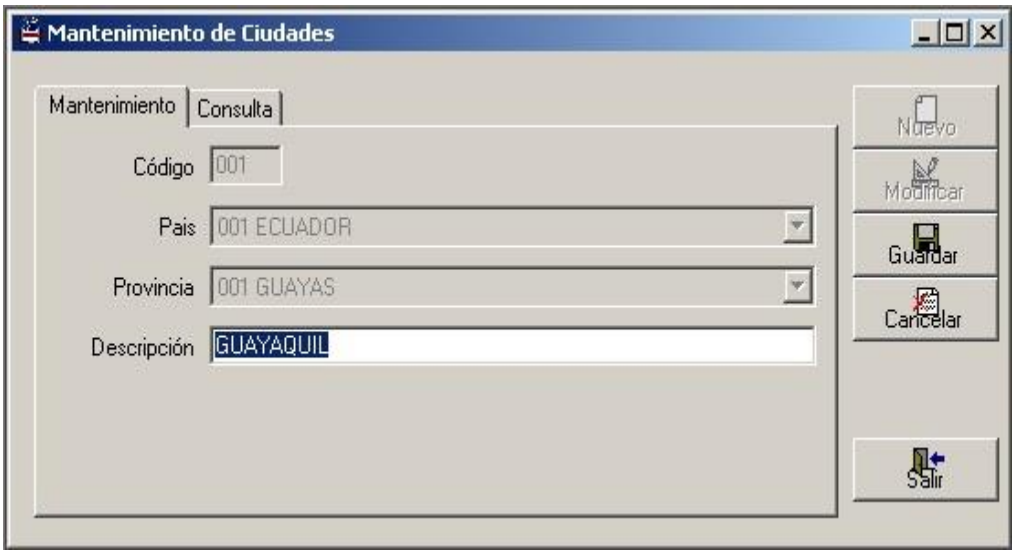


Figura 2.9. Ejemplo de Modificar Ciudades

2.4. Mantenimiento de Categorías de Estudiantes

En este formulario se tiene lo siguiente:

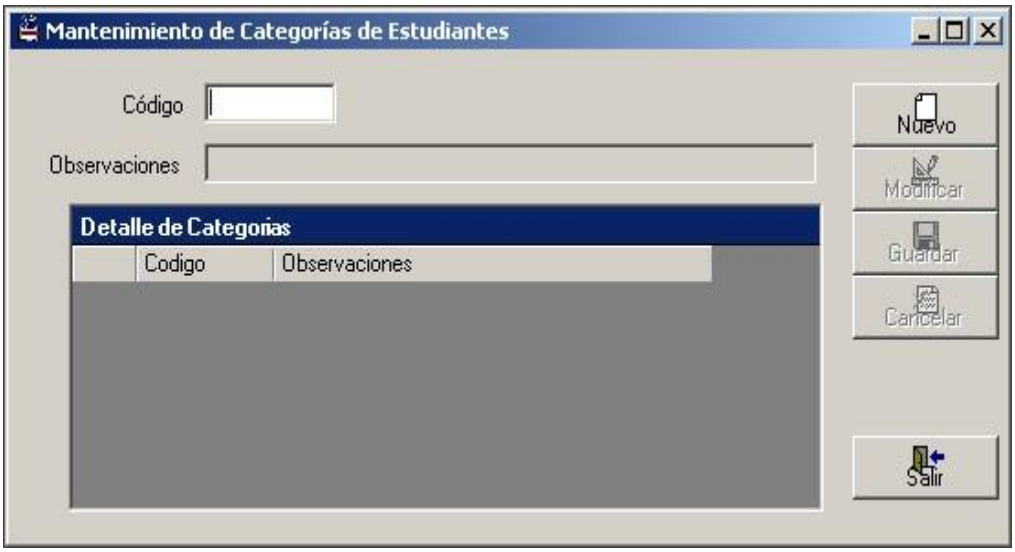


Figura 2.10. Pantalla de Mantenimiento de Categorías

- Una caja de texto llamada “Código” en la cual se digitará la categoría (se habilita para la búsqueda de una categoría).
- Una caja de texto llamada “Observaciones” la cual es opcional en la cual se digitará alguna observación respecto a esa categoría.
- Un “Grid de Consulta” en donde se mostrarán todos los datos de las categorías.

En el lado derecho de esta pantalla tenemos:

- Un botón “Nuevo” el cual se presiona cuando se quiere ingresar nuevas categorías.
- Un botón “Modificar” en la cual se presiona cuando se quiere cambiar alguna categoría.
- Un botón “Guardar” el cual se presiona para grabar las categorías ingresadas a la base de datos.
- Un botón “Cancelar” el cual se presiona cuando las categorías están incorrectos.
- Un botón “Salir” se presiona cuando el usuario desea salir de la forma.

A continuación daremos un pequeño ejemplo de dicha pantalla:

Figura 2.11. Ejemplo de Mantenimiento de Categorías

2.5. Mantenimiento de Colegios

En este formulario el usuario se encarga de realizar el mantenimiento y consulta de los colegios.

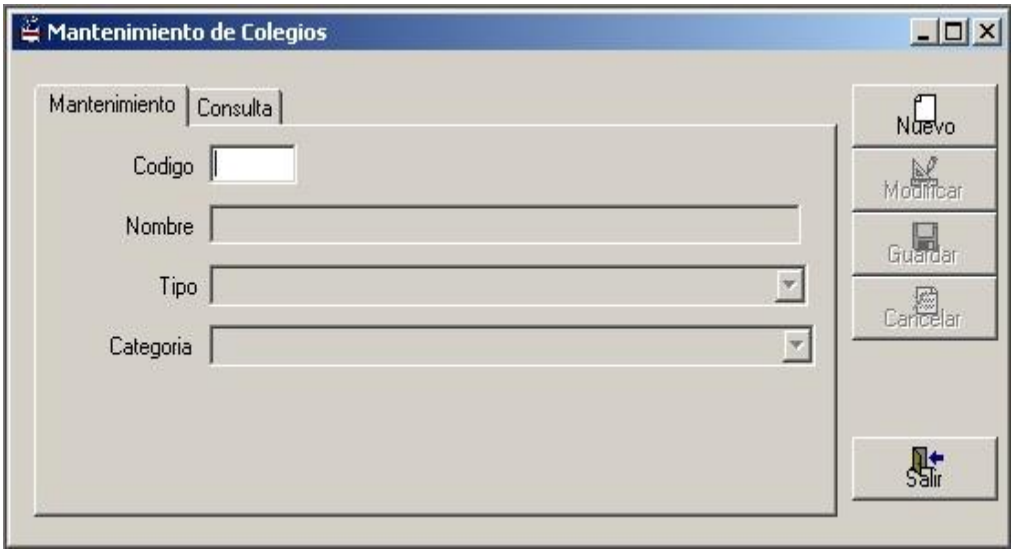


Figura 2.12. Pantalla de Mantenimiento de Colegios

Referente al mantenimiento se tiene:

- Una caja de texto llamada “Código” en la cual se digitará el código del colegio (se habilita para la búsqueda de un colegio).
- Una caja de texto llamada “Nombre” en la cual se digitará el nombre del colegio.

- Un combo llamado “Tipo” en la que se escogerá el tipo de colegio que corresponde ya sea Particular o Nacional en este caso.
- Un combo llamado “Categoría” en donde se escogerá la categoría del colegio ingresado en el sistema.

Como por ejemplo:

Figura 2.13. Ejemplo de Mantenimiento de Colegio-Paso1

Figura 2.14. Ejemplo de Mantenimiento de Colegios-Paso2

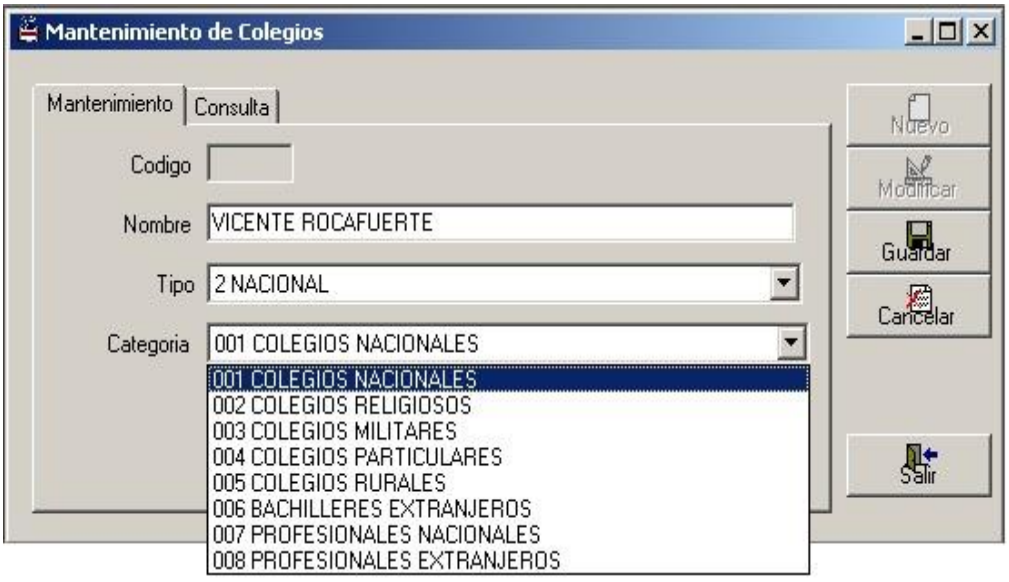


Figura 2.15. Ejemplo de Mantenimiento de Colegios-Paso3

Con respecto a consulta se tiene:

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de los colegios que se encuentran en el sistema.
- Un “Grid de Consulta” que sirve para la búsqueda de los detalles de los colegios, y esto se lo hace presionando el botón.
- Un botón “Salir” en la cual se presiona cuando el usuario desea salir de la forma.

Como por ejemplo:

Figura 2.16. Ejemplo de Consulta de Colegios

2.6. Mantenimiento de Especializaciones del Bachillerato

En dicho formulario el usuario se encargará de realizar el mantenimiento de las especializaciones del bachillerato.

En ella se tiene la siguiente información:

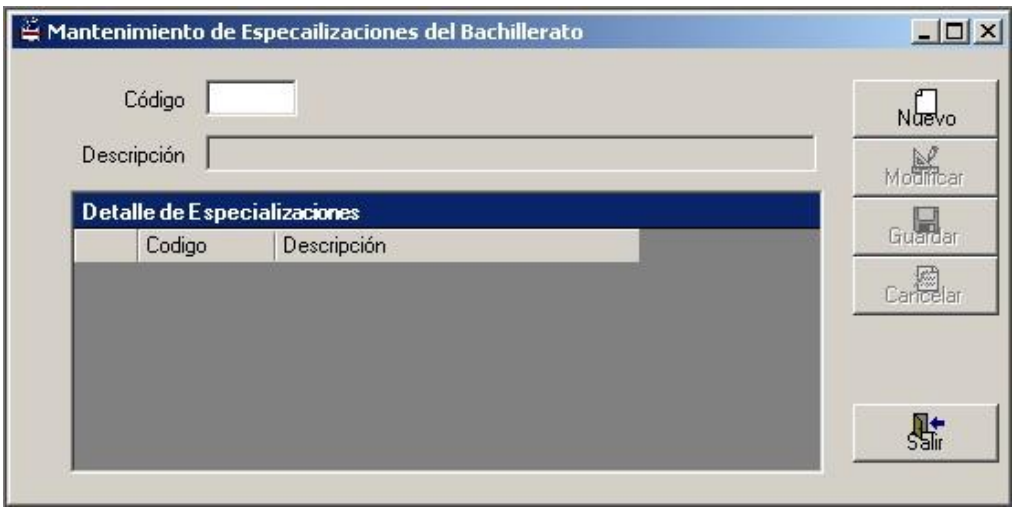


Figura 2.17. Pantalla de Mantenimiento de Especializaciones del Bachillerato

- Una caja de texto llamada “Código” en la cual se digitará el código de la especialización (se habilita para la búsqueda de alguna especialización).
- Una caja de texto llamada “Descripción” en la cual se digitará la especialización del bachillerato.

- Un “Grid de Consulta” donde se detallarán todos los datos que hayan ingresado en el sistema hasta ese momento.

Como por ejemplo:

Figura 2.18. Ejemplo de Mantenimiento de Especializaciones

2.7. Mantenimiento de Universidades

Mediante esta pantalla realizamos mantenimiento y consulta de las universidades.

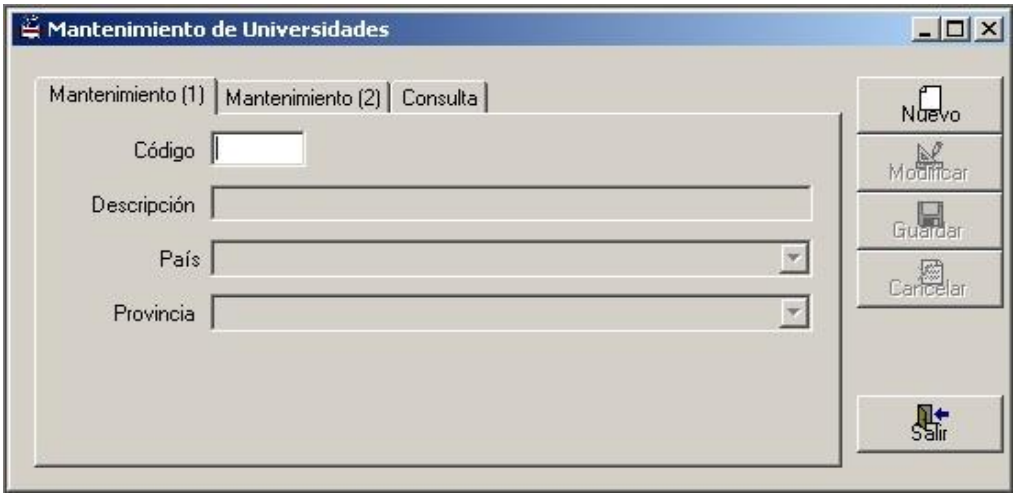


Figura 2.19. Pantalla de Mantenimiento de Universidades

Con respecto al mantenimiento (1) tenemos:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código de la universidad (se habilita para la búsqueda de alguna universidad en particular).
- Una caja de texto llamada “Descripción” en la cual se digitará el nombre de la universidad.
- Un combo llamado “País” en donde se seleccionará el país de donde proviene dicha universidad.
- Un combo llamado “Provincia” en donde se seleccionará la provincia oriunda de la universidad.

Ejemplo correspondiente al mantenimiento (1):

Figura 2.20. Ejemplo de Mantenimiento de Universidades (1)

Referente al mantenimiento (2) se tiene:

Figura 2.21.Pantalla de Mantenimiento de Universidades (2)

- Un combo llamado “Ciudad” en donde se seleccionará la ciudad que provenga dicha universidad.
- Una caja de texto llamada “Dirección” en la cual se digitará la dirección o lugar que esté ubicada la universidad.
- Una caja de texto llamada “Teléfono” en la cual se digitará el número telefónico de la institución.

Ejemplo correspondiente al mantenimiento (2):

Figura 2.22. Ejemplo de Mantenimiento de Universidades (2)

A lo que se refiere a consulta se tiene lo siguiente:

Figura 2.23. Pantalla de Consulta de Universidades

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de las universidades que se encuentran en el sistema.
- Un “Grid de Consulta” donde se detallarán todos los datos que hayan ingresado hasta ese momento en el sistema.

Tenemos como ejemplo la siguiente información con respecto a lo que se refiere a consulta:

Figura 2.24. Ejemplo de Consulta de Universidades

2.8. Mantenimiento de Títulos Universitarios

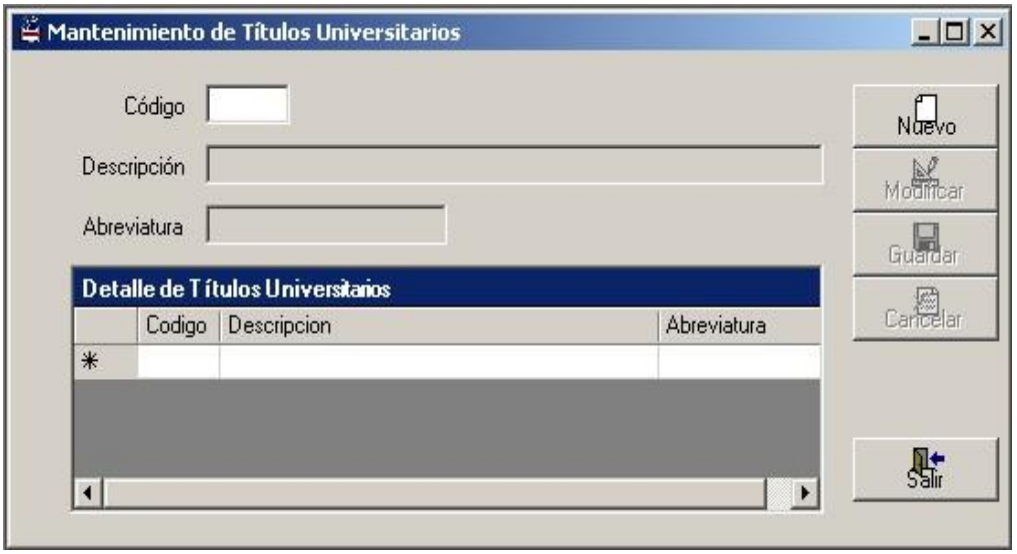


Figura 2.25. Pantalla de Mantenimiento de Títulos Universitarios

En dicha pantalla se tiene la siguiente información:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del título universitario (se habilita para la búsqueda de algún título universitario en particular).
- Una caja de texto llamada “Descripción” en la cual se digitará el nombre del título.
- Una caja de texto llamada “Abreviatura” en la cual se digitará la abreviatura del correspondiente título.
- Un “Grid de Consulta” en donde aparecerá toda la información que se haya acumulado en el sistema.

Por ejemplo tenemos:

Figura 2.26. Ejemplo de Mantenimiento de Títulos Universitarios

2.9. Mantenimiento de Tipos de Subunidad

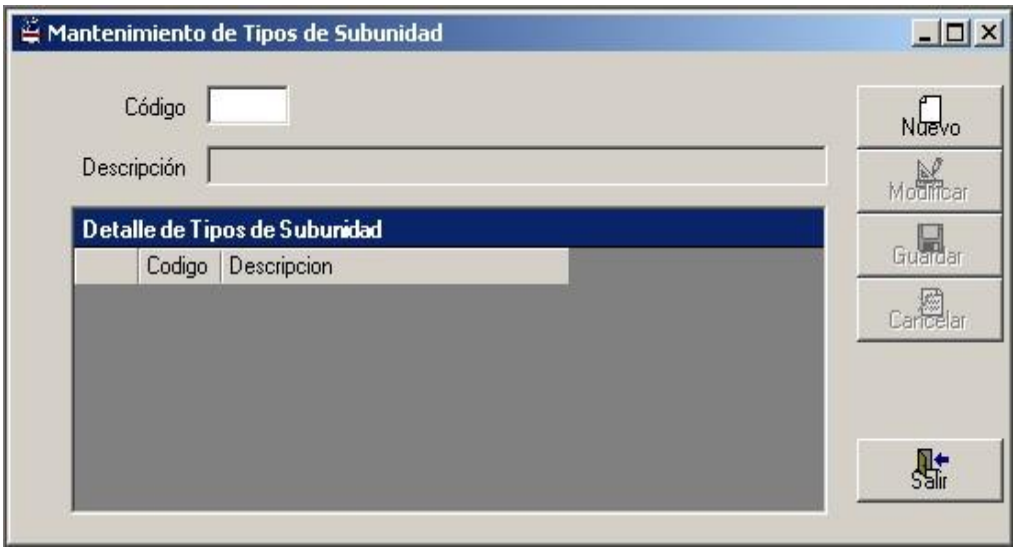


Figura 2.27. Pantalla de Mantenimiento de Tipos de Subunidad

En esta pantalla contiene lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del tipo de subunidad (se habilita para la búsqueda de algún tipo de subunidad).
- Una caja de texto llamada “Descripción” en la cual se digitará el tipo de Subunidad (por ejemplo: carrera o escuela).
- Un “Grid de Consulta” en donde aparecerá todos los datos ingresados en el sistema.

Como por ejemplo:

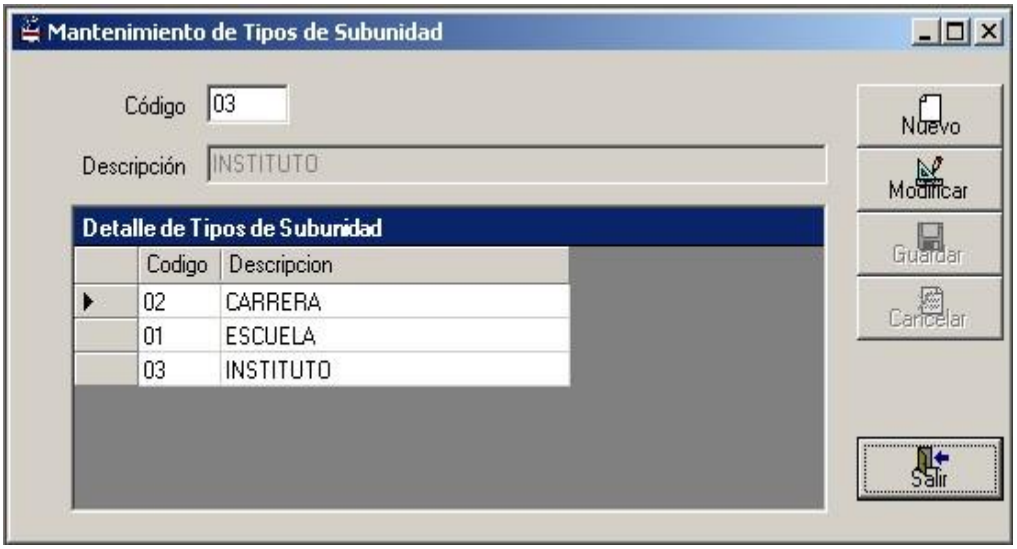


Figura 2.28. Ejemplo de Mantenimiento de Tipos de Subunidad

2.10. Mantenimiento de Tipo de Solicitudes

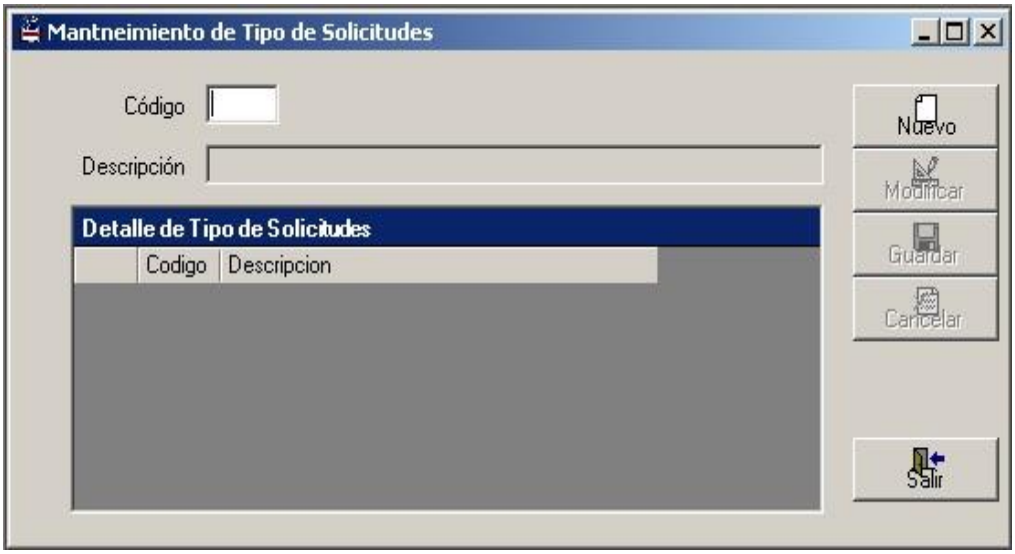


Figura 2.29. Pantalla de Mantenimiento de Tipo de Solicitudes

Se tiene lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del tipo de solicitud (se habilita para la búsqueda de algún tipo de solicitud).
- Una caja de texto llamada “Descripción” en la cual se digitará el tipo de solicitud.
- Un “Grid de Consulta” en donde aparecerá todos los datos ingresados en el sistema.

Como ejemplo tenemos:

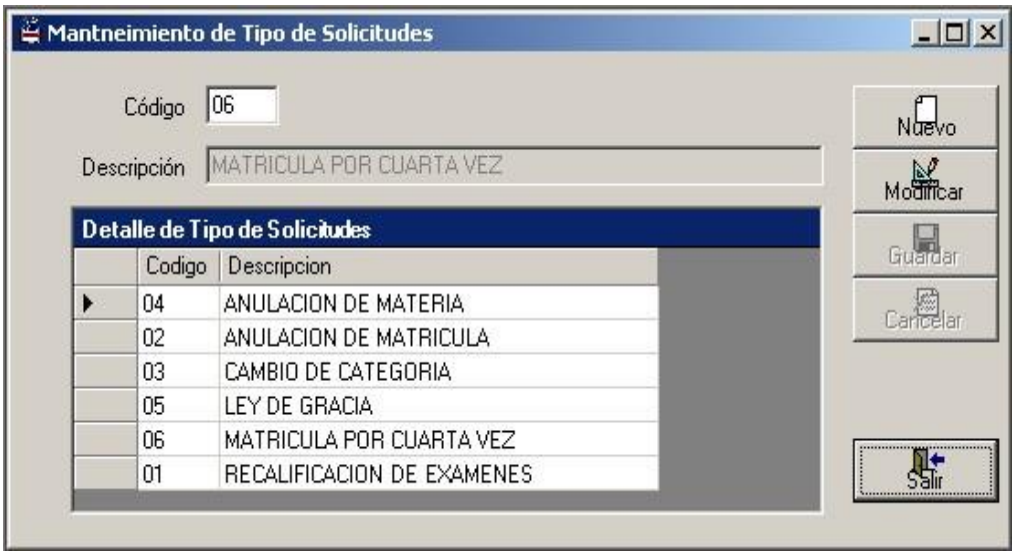


Figura 2.30. Ejemplo de Mantenimiento de Tipo de Solicitudes

2.11. Mantenimiento de Descuento y Aranceles

En este formulario el usuario se debe de encargar del mantenimiento y consulta de descuento y aranceles.

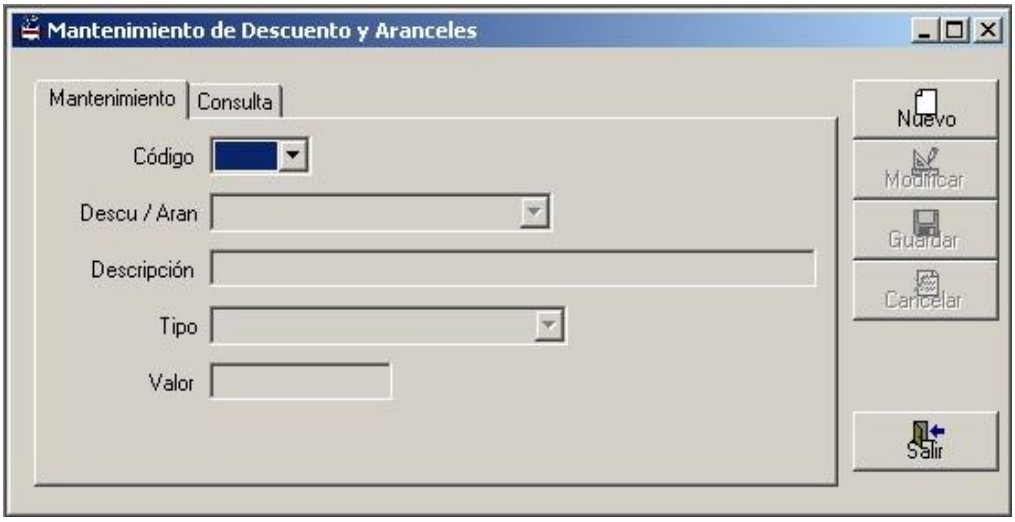


Figura 2.31. Pantalla de Mantenimiento de Descuento y Aranceles

En lo referente al mantenimiento se tiene:

- Un combo llamado “Código” donde debe de seleccionarse el respectivo código del tipo de descuento y arancel.
- Un combo llamado “Desc/Aranc” donde se seleccionará si el tipo escogido es descuento o arancel.
- Una caja de texto llamado “Descripción” en la cual se digitará la descripción del descuento o arancel.
- Un combo llamado “Tipo” donde se seleccionará si el tipo de descuento o arancel es de porcentaje o de monto fijo.

- Una caja de texto llamado “Valor” donde se va a grabar la cantidad del descuento o arancel.

Con respecto a la consulta se tiene:

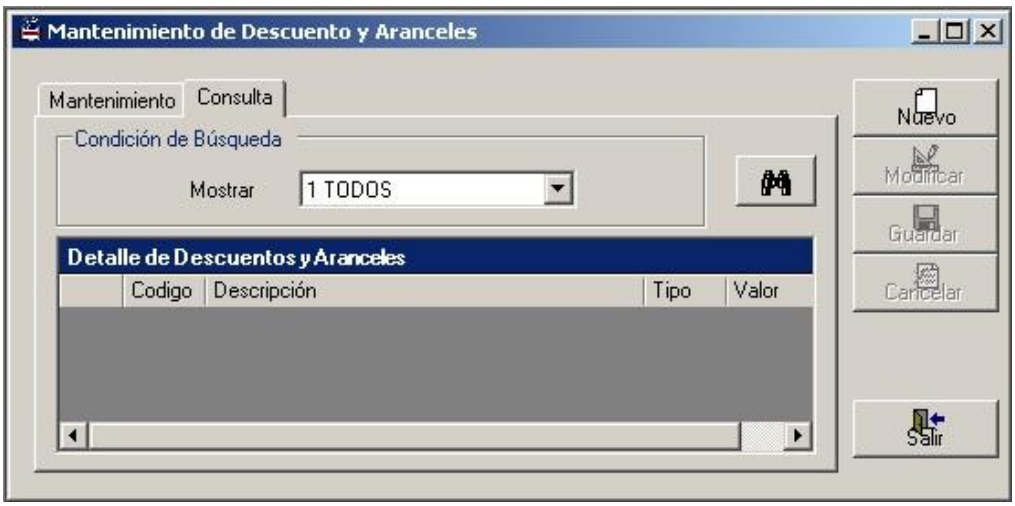


Figura 2.32. Pantalla de Consulta de Descuentos y Aranceles

- Un combo llamado “Muestra” en la cual se selecciona si es descuento, arancel o ambos.
- Un “Grid de Consulta” donde aparecerá toda la información que se haya ingresado en el sistema o la información que se desea buscar cuando se presiona el botón.

Como por ejemplo:

Figura 2.33. Ejemplo de Consulta de Descuentos y Aranceles

En este ejemplo se muestra un listado de algunos descuentos y aranceles con su debido código, descripción, tipo y valor.

A continuación se detalla un listado de Descuentos y Aranceles (Documento facilitado por la facultad del periodo lectivo 2002-2003)

Desc./Aran.	**Descripción**	**Tipo**	**Valor**
Descuento	Promedio del año anterior en 1era matrícula fue mayor de 9.5	Porcentaje	100
Descuento	Promedio del año anterior en 1era matrícula de 9 a 9.49	Porcentaje	75
Descuento	Promedio del año anterior en 1era matrícula de 8.5 a 8.99	Porcentaje	50

Descuento	Promedio del año anterior en 1era matrícula de 8 a 8.49	Porcentaje	25
Descuento	Servidor Universitario	Porcentaje	25
Descuento	Hijo del Servidor Universitario	Porcentaje	10
Descuento	90% de asistencias del año anterior	Porcentaje	20
Descuento	Entre 85 y 89.99% de asistencia del año anterior	Porcentaje	15
Descuento	Discapacidades	Porcentaje	100
Descuento	Deportista	Porcentaje	5
Descuento	Deport. representante de unidad académica	Porcentaje	10
Descuento	Campeones o vicecampeones universitarios	Porcentaje	50
Descuento	Estudiantes que rinden 1era y 2da llamada y aprueban	Porcentaje	5
Descuento	Pobreza crítica confirmada	Porcentaje	80
Arancel	Especies Valoradas	Monto fijo	0.80
Arancel	Derecho de copia para estudiante	Monto fijo	1
Arancel	Derecho de copia para egresados o profesionales	Monto fijo	4
Arancel	Derecho de duplicado del título profesional o grado académico	Monto fijo	7

Arancel	Derecho de pase a otra universidad	Monto fijo	44
Arancel	Derecho de pase a otra unidad académica	Monto fijo	7
Arancel	Certificado de salud	Monto fijo	5.50
Arancel	Certificado de egresado	Monto fijo	5.50
Arancel	Derecho de examen de recuperación por materia	Monto fijo	2.50
Arancel	Derecho de solicitud de matrícula a crédito	Monto fijo	1
Arancel	Revalidación, homologación o equiparación de título profesional o grado académico (nacionales)	Monto fijo	220
Arancel	Revalidación, homologación o equiparación de título profesional o grado académico (extranjeros)	Monto fijo	440
Arancel	Tasa de emisión y liquidación del impuesto dos por mil	Monto fijo	3
Arancel	Derecho de participación en concurso para profesores universitarios	Monto fijo	16
Arancel	Derecho de certificado final para auxiliar de enfermería	Monto fijo	16
Arancel	Título de pregrado (Nacionales)	Monto fijo	68
Arancel	Otro, segundo o posterior título de pregrado (Nacionales)	Monto fijo	83

Arancel	Diplomado y especialidad (Nacionales)	Monto fijo	109
Arancel	Maestría (Nacionales)	Monto fijo	320
Arancel	Doctorado (Nacionales)	Monto fijo	660
Arancel	Título de pregrado (Extranjeros)	Monto fijo	412
Arancel	Otro, segundo o posterior título de pregrado (Extranjeros)	Monto fijo	462
Arancel	Diplomado y especialidad (Extranjeros)	Monto fijo	660
Arancel	Maestría (Extranjeros)	Monto fijo	770

2.12. Mantenimiento de Motivos de Cambio de Pénsum

Este formulario se crea , debido a los inconvenientes que presentan los alumnos en el periodo lectivo, debido a que no ven las asignaturas en forma regular o por pérdida de año.

En esta pantalla tenemos:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del motivo de cambio de pénsum (se habilita para la búsqueda de los pénsums).
- Una caja de texto llamada “Descripción” en la cual se digitará la descripción del motivo de cambio de pénsum.

- Un “Grid de Consulta” en donde aparecerá todos los datos ingresados al sistema.

Como por ejemplo:

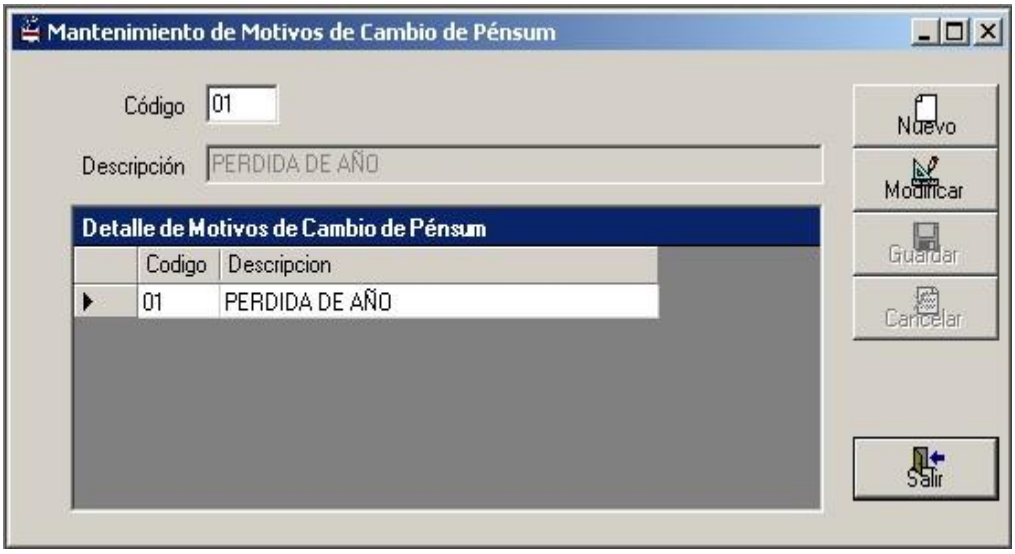


Figura 2.34. Ejemplo de Motivo de Cambio de Pénsum

2.13. Mantenimiento de Tipos de Proyectos

Este formulario el usuario se encarga del mantenimiento de los tipos de proyectos de graduación.

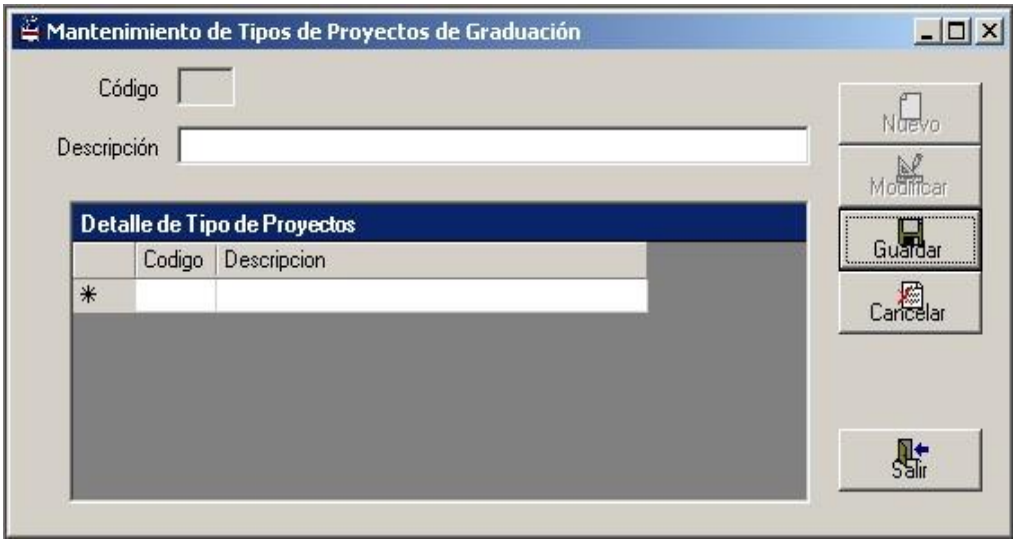


Figura 2.35. Pantalla de Tipo de Proyectos de Graduación

En esta pantalla aparecerá lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del tipo de proyecto (se habilita para la búsqueda del respectivo tipo de proyecto).
- Una caja de texto llamada “Descripción” en la cual se digitará el tipo de proyecto de graduación que el estudiante haya escogido.
- Un “Grid de Consulta” en donde se detallarán todos los datos ingresados al sistema.

Como por ejemplo:

Figura. 2.36. Ejemplo de Tipos de Proyectos de Graduación

2.14. Mantenimiento de Bancos

En este formulario el usuario debe de encargarse del mantenimiento de los bancos.

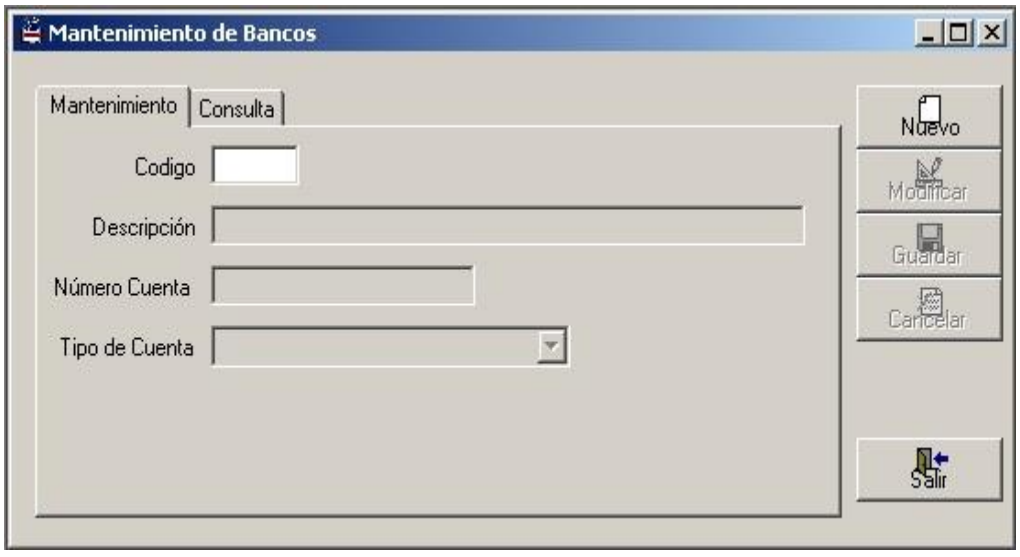


Figura 2.37. Pantalla de Mantenimiento de Bancos

En lo que respecta a mantenimiento tenemos:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del banco (se habilita para la búsqueda del banco).
- Una caja de texto llamada “Descripción” en la cual se digitará la descripción del banco.
- Una caja de texto llamada” Numero Cuenta” en la cual se digitará en respectivo número de cuenta del banco.
- Un combo llamado “Tipo de Cuenta “ en donde se selecciona el tipo de cuenta del banco, ya sea ahorro o crédito.

Como por ejemplo:

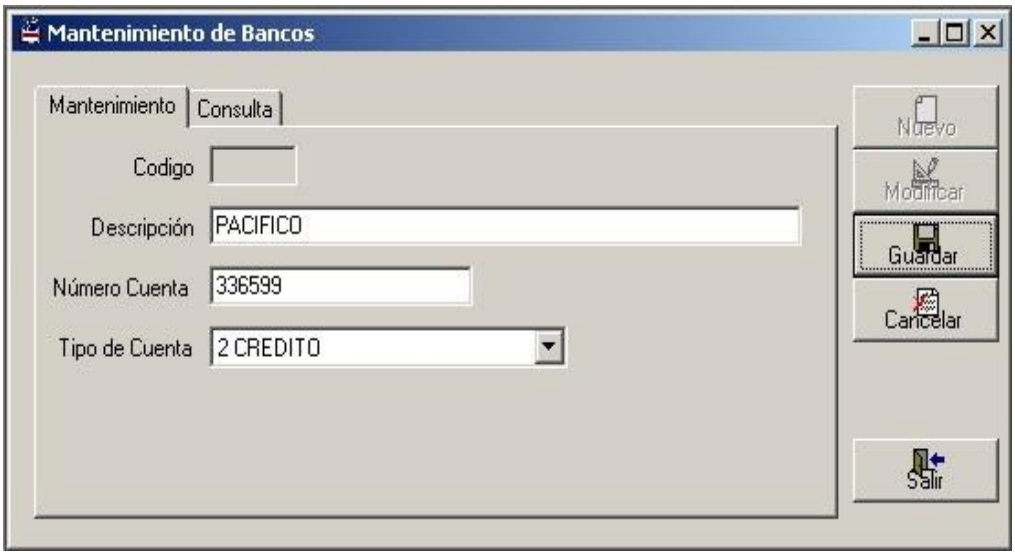


Figura 2.38. Ejemplo de Bancos

Con lo que se refiere a consulta de bancos se tiene:

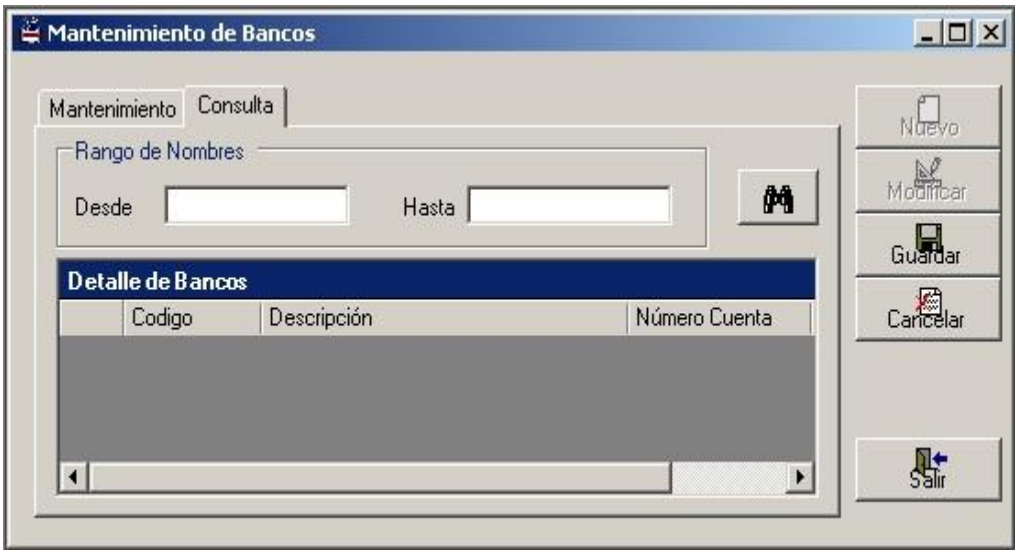


Figura 2.39. Consulta de Bancos

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de los bancos que ingresaron al sistema.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos que corresponden a los bancos, y esto se lo hace presionando el botón .

Como por ejemplo:

Figura 2.40. Ejemplo de Consulta de Bancos

2.15. Mantenimiento de Concepto de Cobros

En dicha pantalla el usuario se encarga del mantenimiento de los conceptos de cobros.

Ya entrando en la pantalla tenemos:

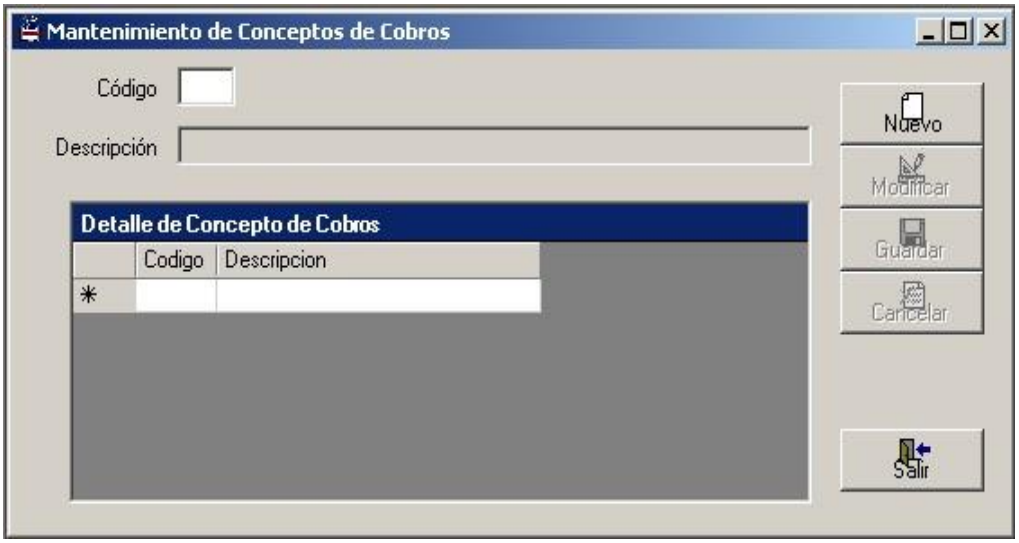


Figura 2.41. Pantalla de Concepto de Cobros

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código por concepto de cobros (se habilita para la búsqueda de algún concepto de cobros).
- Una caja de texto llamada “Descripción” en la cual se digitará un concepto de cobros.
- Un “Grid de Consulta” en donde se detallarán todos los datos ingresados al sistema.

Como por ejemplo:

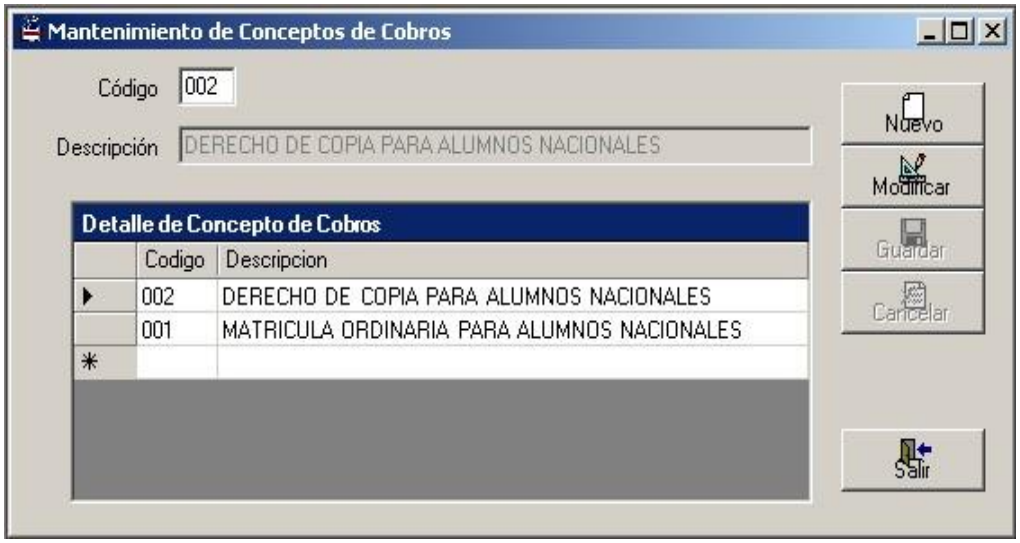


Figura 2.42. Ejemplo de Conceptos de Cobros

2.16. Mantenimiento Fichas Docentes

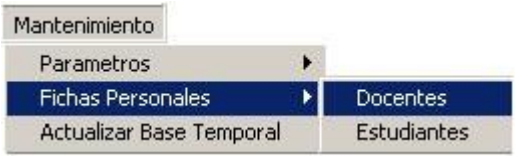


Figura 2.43. Acceso de Mantenimiento de Fichas Docentes

Este formulario se encargará de llenar todos los datos personales, datos generales, domicilio, trabajo y estudios realizados del docente.

En la pantalla de mantenimiento de la ficha del docente correspondiente a los datos personales se tendrá lo siguiente:

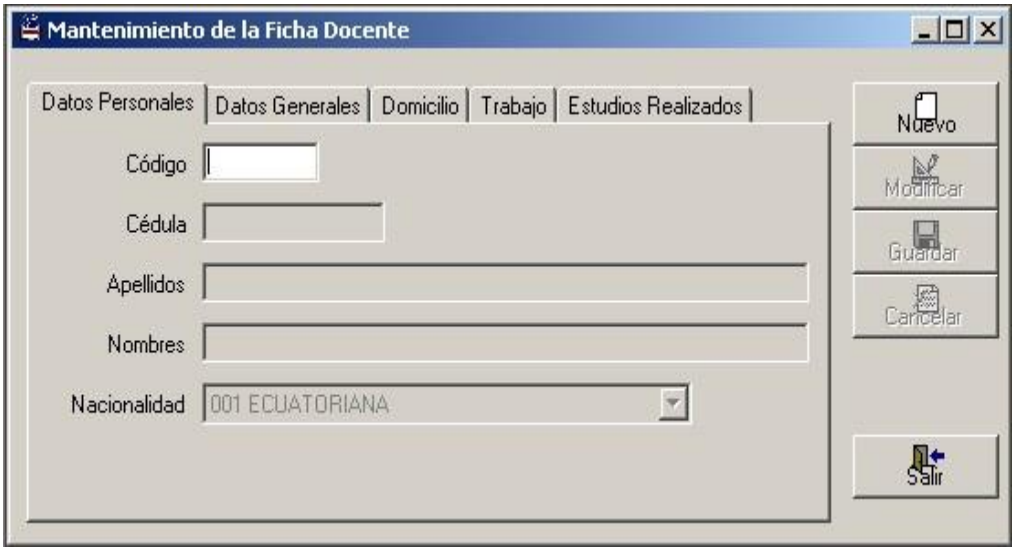


Figura 2.44. Pantalla de la Ficha Docente (Datos Personales)

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del docente (se habilita para la búsqueda de algún docente en particular).
- Una caja de texto llamada “Cédula” en la cual se digitará el correspondiente numero de cédula del docente.
- Una caja de texto llamada “Apellidos” en la cual se digitará los apellidos del docente.
- Una caja de texto llamada “Nombres” en la cual se digitará los nombres del docente.
- Un combo llamado “Nacionalidad” en la que se escoge su procedencia de nacimiento.

Como por ejemplo:

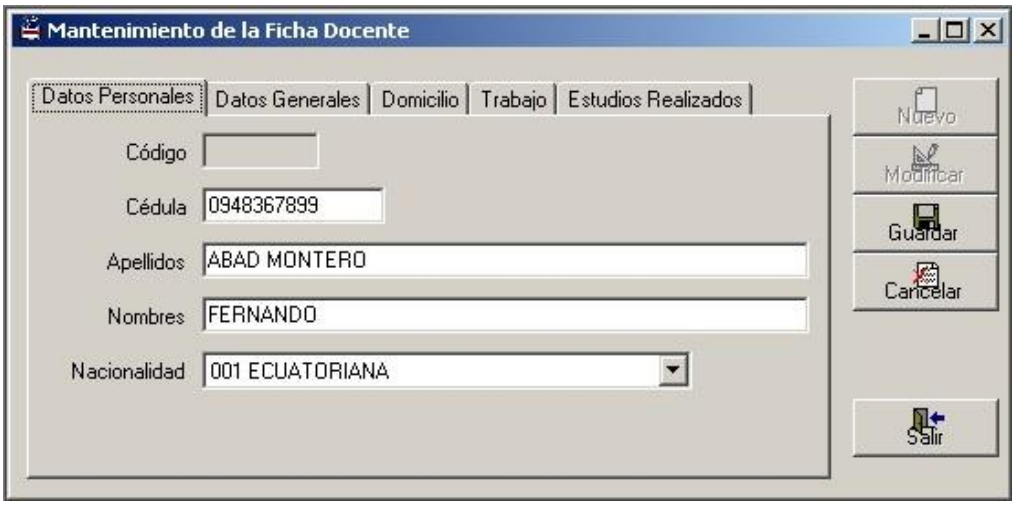


Figura 2.45. Ejemplo de la Ficha Docente (Datos Personales)

Con respecto a los datos generales tenemos:

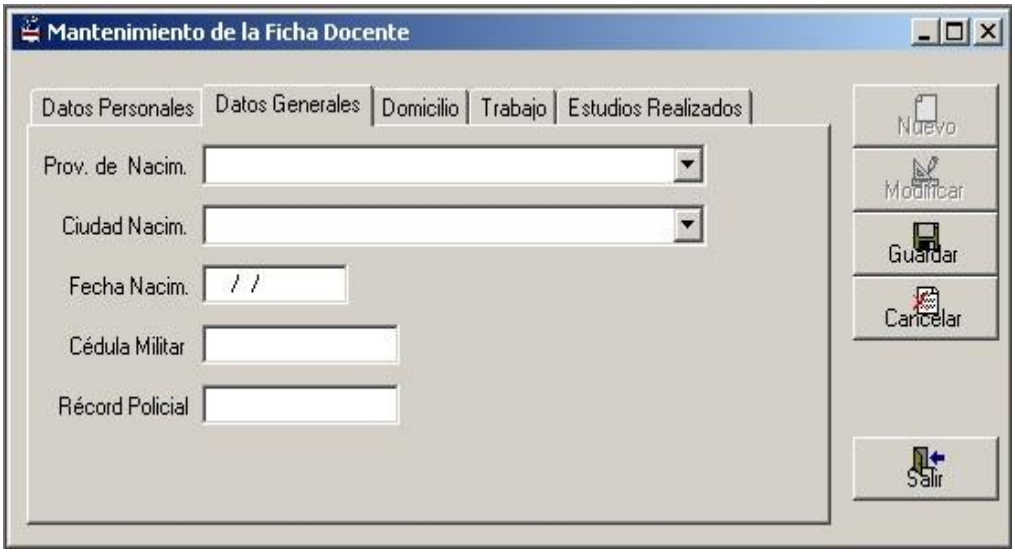


Figura 2.46. Pantalla de la Ficha Docente (Datos Generales)

- Un combo “Prov. de Nacim.” en la que se escoge la provincia donde nació.
- Un combo”Ciudad de Nacim.” en la que se selecciona la ciudad donde nació.
- Una caja de texto “Fecha Nacim.” en la cual se registra el siguiente formato: 4 dígitos del año, 2 dígitos del mes y 2 dígitos del día.
- Una caja de texto “Cédula Militar” en la cual se digitará el número de la libreta militar.
- Una caja de texto “Récord Policial” en la cual se digitará el número del récord policial.

Como por ejemplo:

Figura 2.47. Ejemplo de la Ficha Docente (Datos Generales)

Referente a domicilio se tiene lo siguiente:

Figura 2.48. Pantalla de la Ficha Docente (Domicilio)

- Un combo llamado “Provincia” en la cual se selecciona la provincia donde reside.
- Un combo llamado “Ciudad” en la cual se selecciona la ciudad donde reside.
- Una caja de texto llamada “Dirección” en la cual se digita el lugar donde reside el docente.
- Dos cajas textos llamada “Teléfono 1” y “Teléfono 2” en donde se digitarán los números telefónicos donde se lo puede ubicar al docente.

Como por ejemplo:

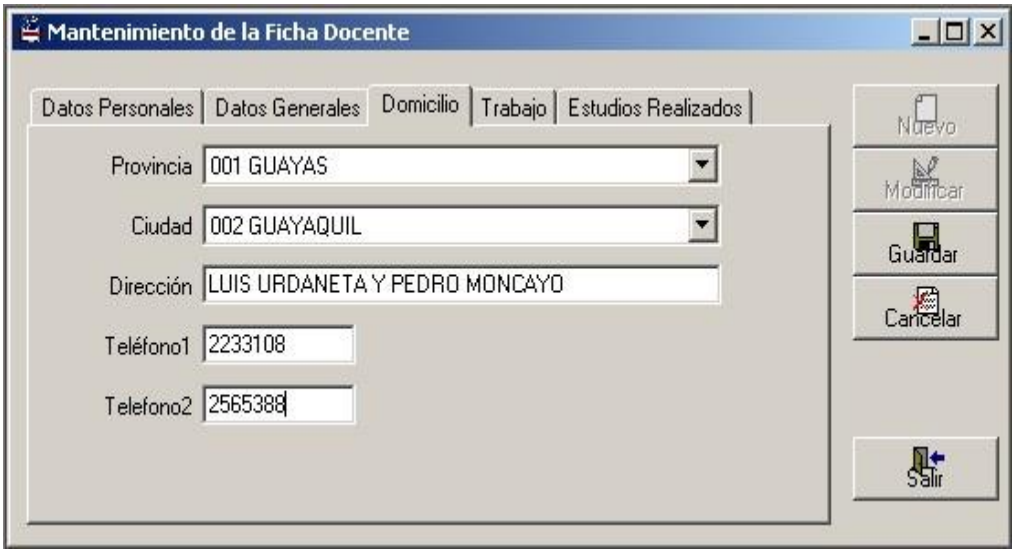


Figura 2.49. Ejemplo de la Ficha Docente (Domicilio)

En lo que concierne a trabajo se tiene:

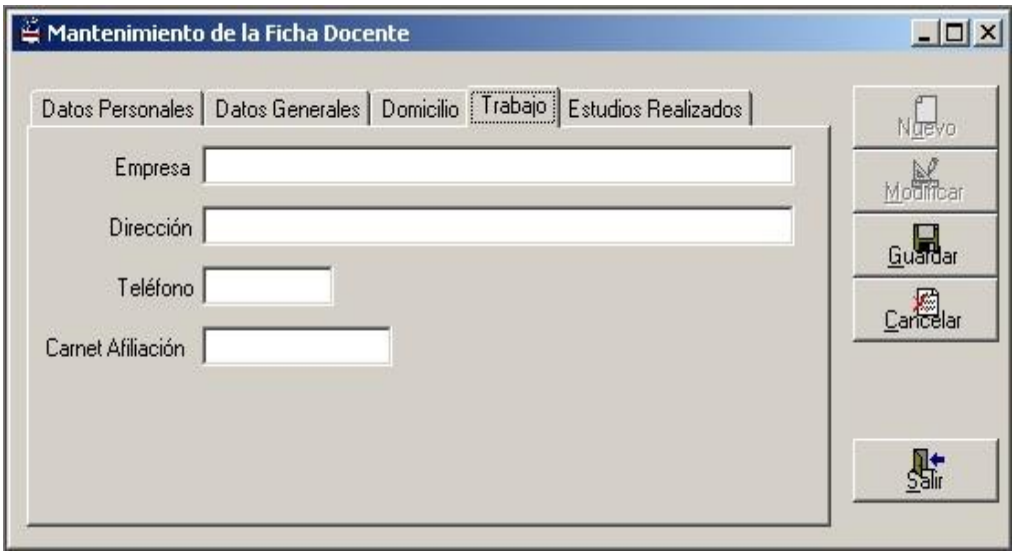


Figura 2.50. Pantalla de la Ficha Docente (Trabajo)

- Una caja de texto llamada “Empresa” en la cual se digitará el nombre de la empresa donde trabaja el docente.
- Una caja de texto llamada “Dirección” en la cual se digitará la dirección donde trabaja el docente.
- Una caja de texto llamada “Teléfono” en la cual se digitará el número telefónico de la empresa donde trabaja el docente.
- Una caja de texto llamada “Carnet Afiliación” en la cual se digitará el número del seguro social del docente.

Como por ejemplo:

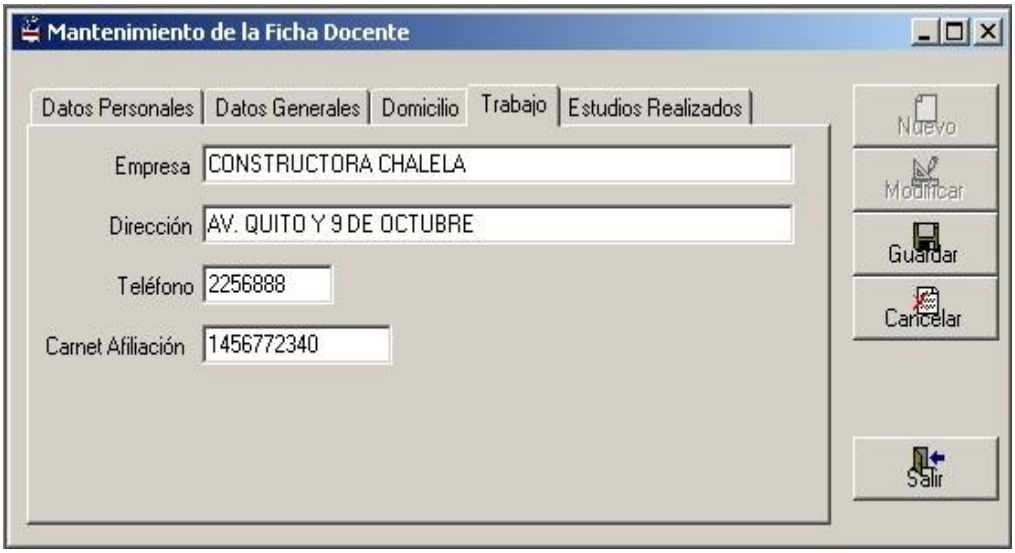


Figura 2.51. Ejemplo de la Ficha Docente (Trabajo)

A lo que respecta a estudios realizados:

Figura 2.52. Pantalla de la Ficha Docente (Estudios Realizados)

- Un combo llamado “Título Univ.” en donde se escogerá el título del docente.
- Un combo llamado “Universidad” en donde se escogerá el nombre de la universidad de la cual obtuvo el título.
- Un combo llamado “Logros Obtenidos” en donde se digitará los logros obtenidos en su vida profesional.

Como por ejemplo:

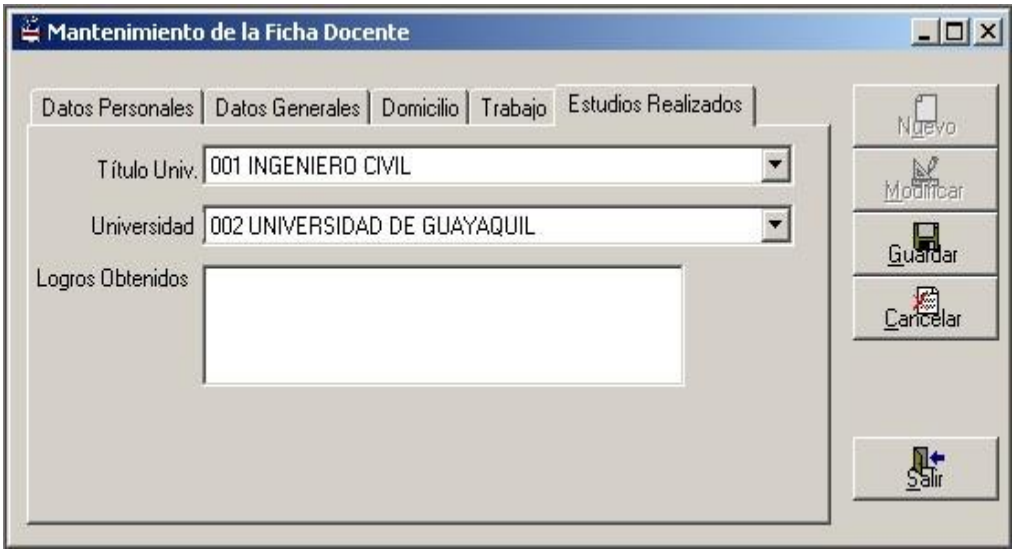


Figura 2.53. Ejemplo de la Ficha Docente (Estudios Realizados)

2.17. Mantenimiento de la Ficha Estudiantil

Este formulario se encargará de llenar todos los datos personales, datos generales, domicilio, trabajo del estudiante.

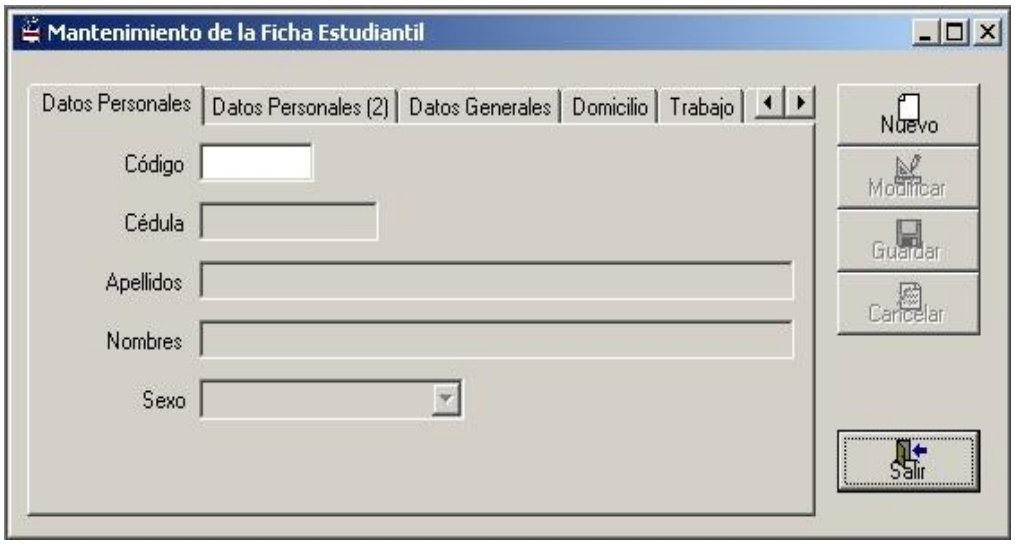


Figura 2.54. Pantalla de la Ficha Estudiantil

Con respecto a los datos personales (1) se tiene:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del estudiante (se habilita para la búsqueda de algún estudiante).
- Una caja de texto llamada “Cédula” en la cual se digitará el correspondiente numero de cédula del estudiante.
- Una caja de texto llamada “Apellidos” en la cual se digitará los apellidos del estudiante.
- Una caja de texto llamada “Nombres” en la cual se digitará los nombres del estudiante.
- Un combo llamada “Sexo” en la cual se escogerá el sexo del estudiante.

Como por ejemplo:

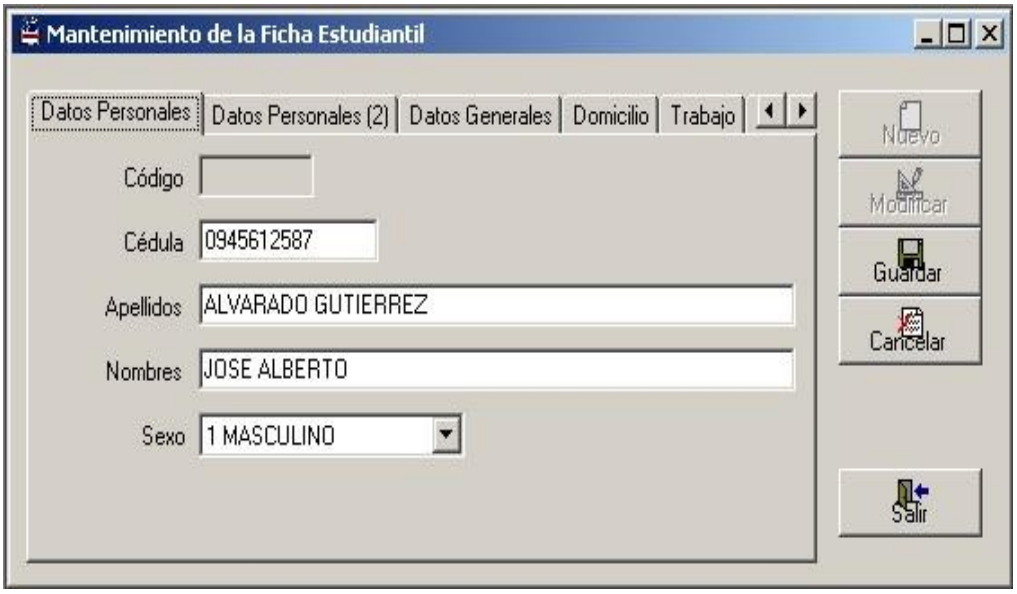


Figura 2.55. Ejemplo de Ficha Estudiantil- Datos Personales

En lo que respecta a datos personales (2) tenemos:

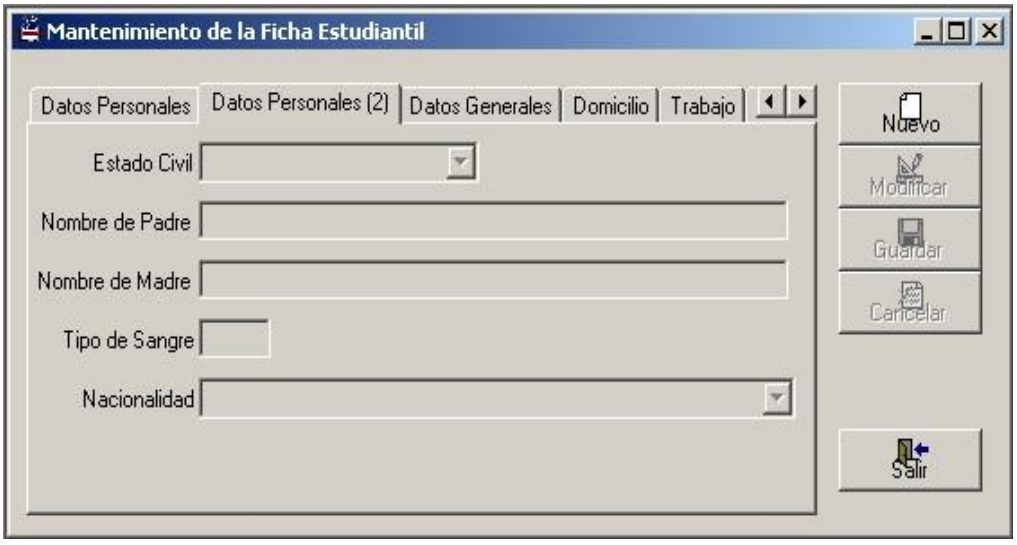


Figura 2.56. Ficha Estudiantil-Datos Personales (2)

- Un combo llamado “Estado Civil” en la que se escoge entre: Casado, Divorciado o Viudo del estudiante.
- Una caja de texto llamado “ Nombre del Padre” en la cual se digitará sus nombres y apellidos.
- Una caja de texto llamada “Nombre de la Madre” en cual se digitará los nombres y apellidos.
- Una caja de texto llamada “Tipo de Sangre” en la cual se digitará el tipo de sangre del estudiante.
- Un combo llamada “Nacionalidad” en la cual se escoge su procedencia de nacimiento.

Como por ejemplo:

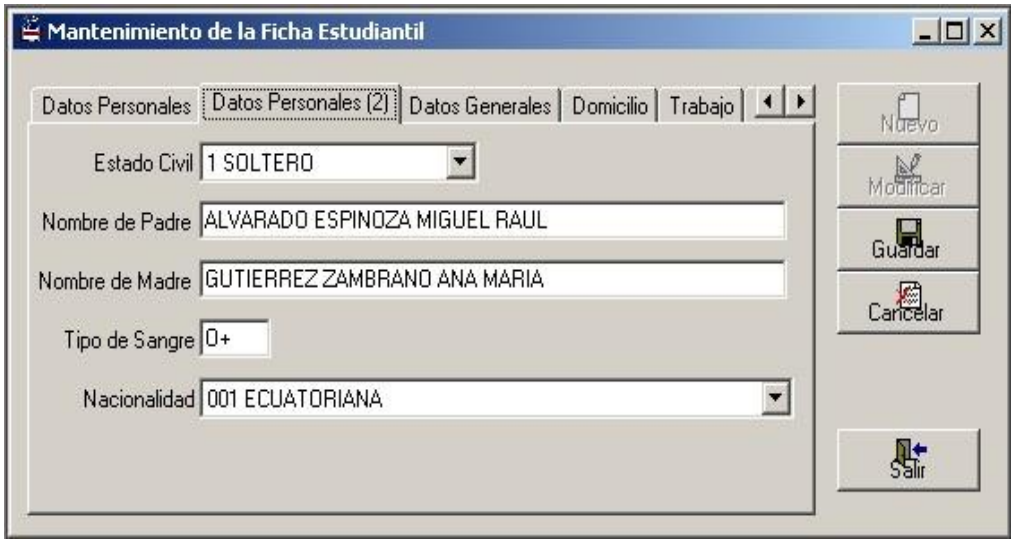


Figura 2.57.Ejemplo Ficha Estudiantil-Datos Personales (2)

Referente a datos generales tenemos:

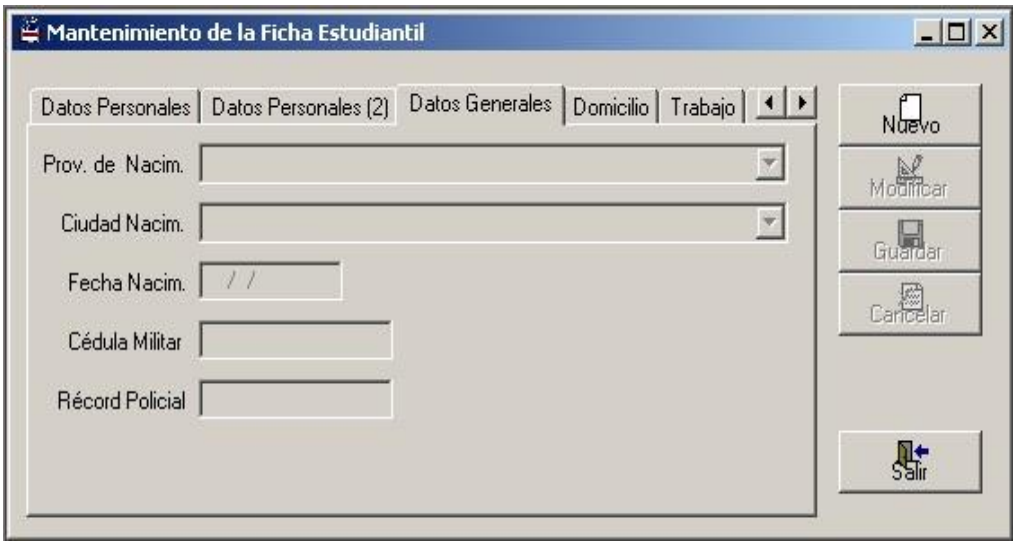


Figura 2.58. Ficha Estudiantil- Datos Generales

- Un combo “Prov. de Nacim.” en la que se escoge la provincia donde nació.
- Un combo”Ciudad de Nacim.” en la que se selecciona la ciudad donde nació.
- Una caja de texto “Fecha Nacim.” en la cual se registra el siguiente formato: 4 dígitos del año, 2 dígitos del mes y 2 dígitos del día.
- Una caja de texto “Cédula Militar” en la cual se digitará el número de la libreta militar.

- Una caja de texto “Récord Policial” en la cual se digitará el número del récord policial.

Como por ejemplo:

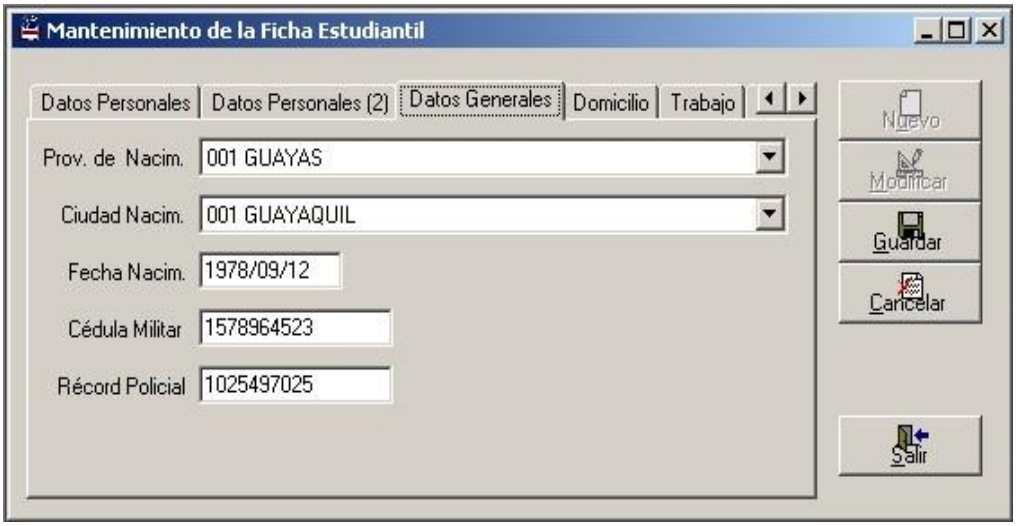


Figura 2.59. Ejemplo de Ficha Estudiantil- Datos Generales

A lo que concierne a domicilio tenemos:

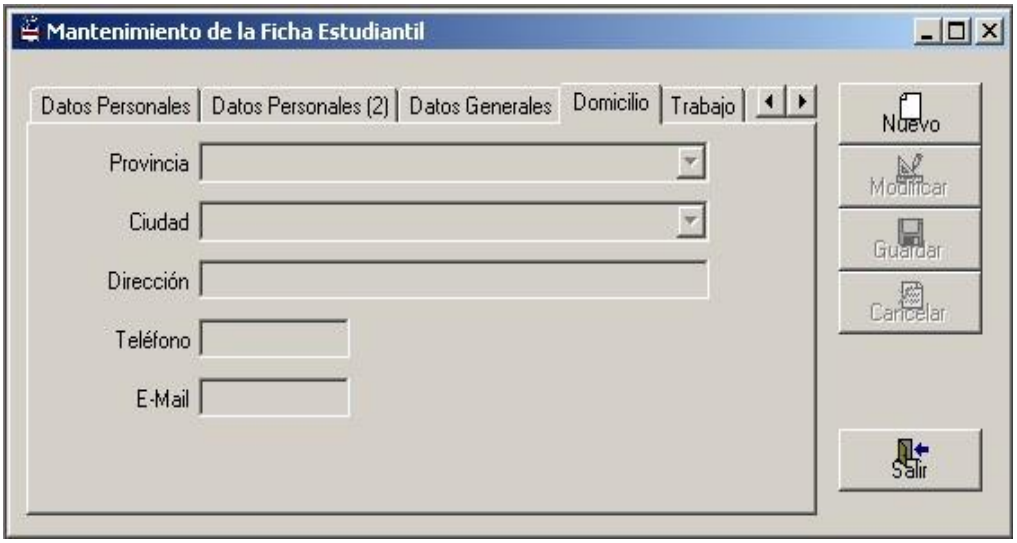


Figura 2.60. Ficha Estudiantil- Domicilio

- Un combo llamado “Provincia” en la cual se selecciona la provincia donde reside.
- Un combo llamado “Ciudad” en la cual se selecciona la ciudad donde reside.
- Una caja de texto llamada “Dirección” en la cual se digita el lugar donde reside el estudiante.
- Una caja de texto llamada “Teléfono” en donde se digitará el número telefónico donde se lo puede ubicar al estudiante a su domicilio.
- Una caja de texto llamada “E-Mail” donde se digitará el correo electrónico del estudiante.

Como por ejemplo:

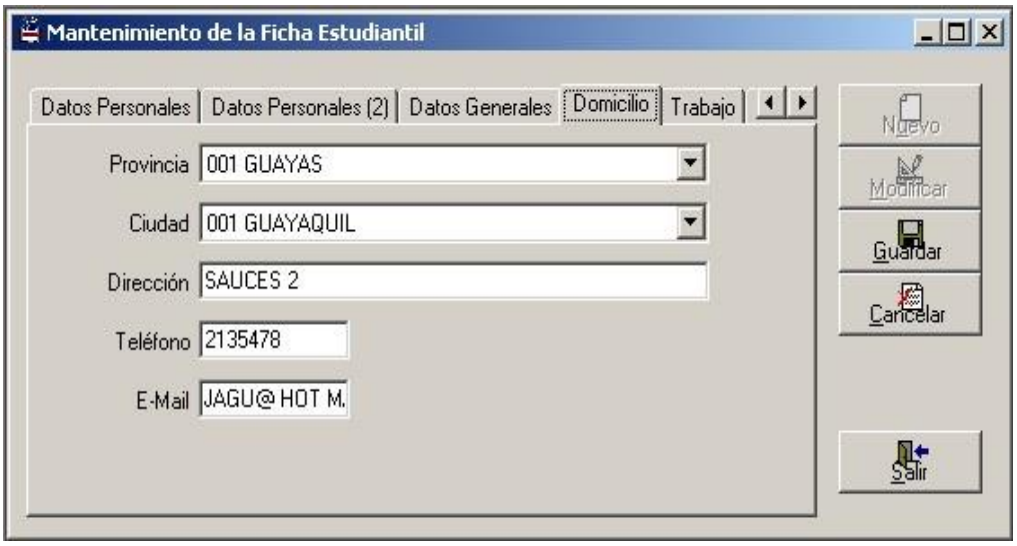


Figura 2.61. Ejemplo de Ficha Estudiantil- Domicilio

Con respecto a trabajo se tiene:

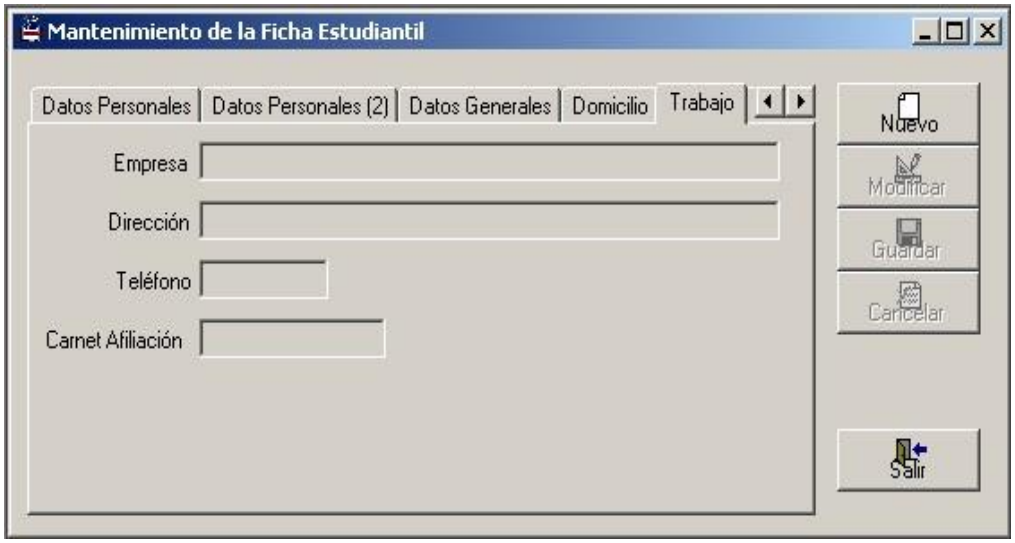


Figura 2.62. Ficha Estudiantil- Trabajo

- Una caja de texto llamada “Empresa” en la cual se digitará el nombre de la empresa donde trabaja el estudiante.
- Una caja de texto llamada “Dirección” en la cual se digitará la dirección donde trabaja el estudiante.
- Una caja de texto llamada “Teléfono” en la cual se digitará el número telefónico de la empresa donde trabaja el estudiante.
- Una caja de texto llamada “Carnet Afiliación” en la cual se digitará el número del seguro social del estudiante.

Como por ejemplo:

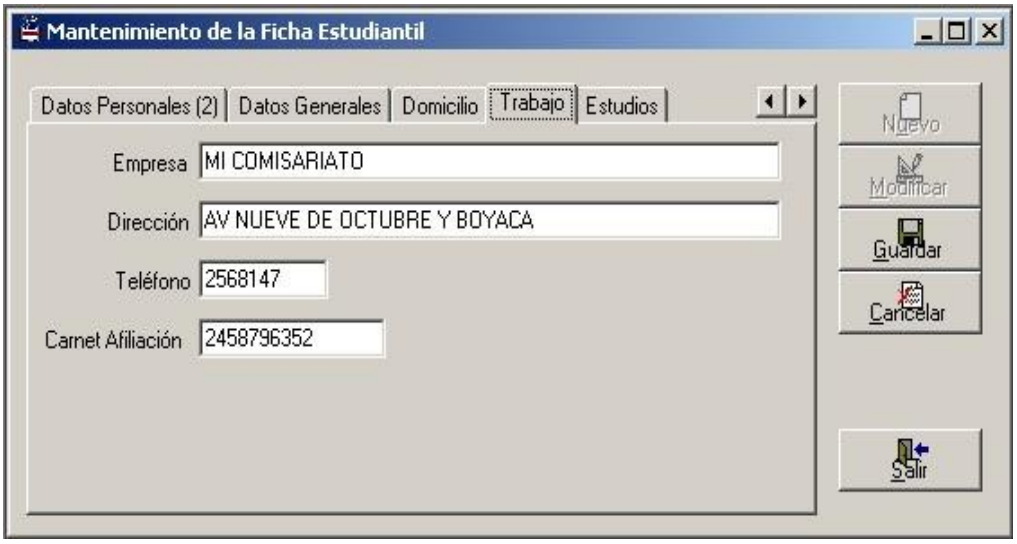


Figura 2.63. Ejemplo de Ficha Estudiantil- Trabajo

En lo que concierne a estudios se tiene:

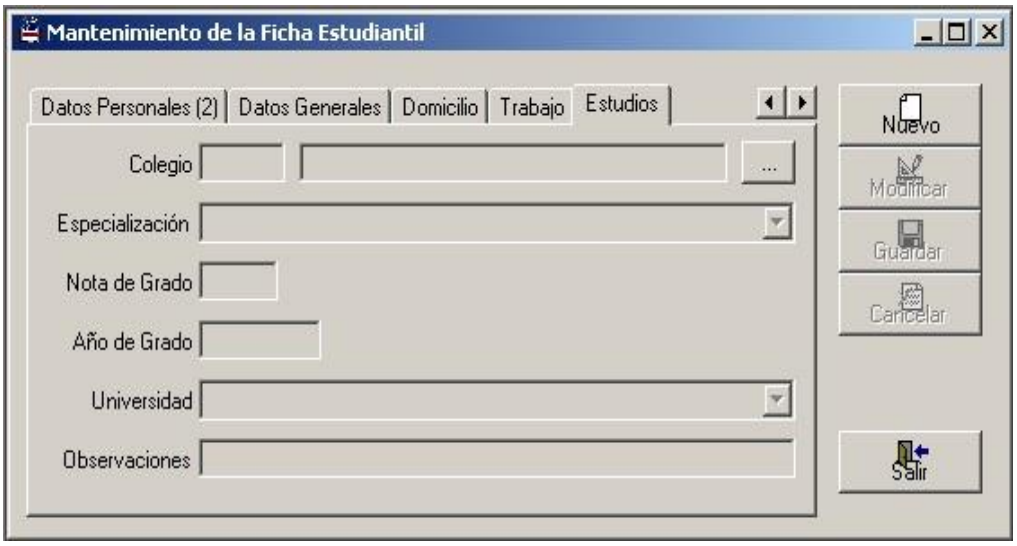


Figura 2.64. Ficha Estudiantil- Estudios

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del colegio (se habilita para la búsqueda de algún colegio en particular). Al presiona el botón que se encuentra en la parte derecha del mismo aparecerá un listado de los nombres de los colegios que fueron ingresados al sistema.
- Un combo llamado “Especialización” en la que se escogerá la especialización del estudiante.
- Una caja de texto llamada “Nota de Grado” en la cual se digitará la nota con que se graduó el estudiante.

- Una caja de texto llamada “Año de Grado” en la cual se digitará el año en que se graduó el estudiante.
- Un combo llamado “Universidad” en donde se escogerá el nombre de la universidad con su respectivo código.
- Una caja de texto llamada “Observaciones” en la cual se digitará alguna observación del estudiante.

Como por ejemplo:

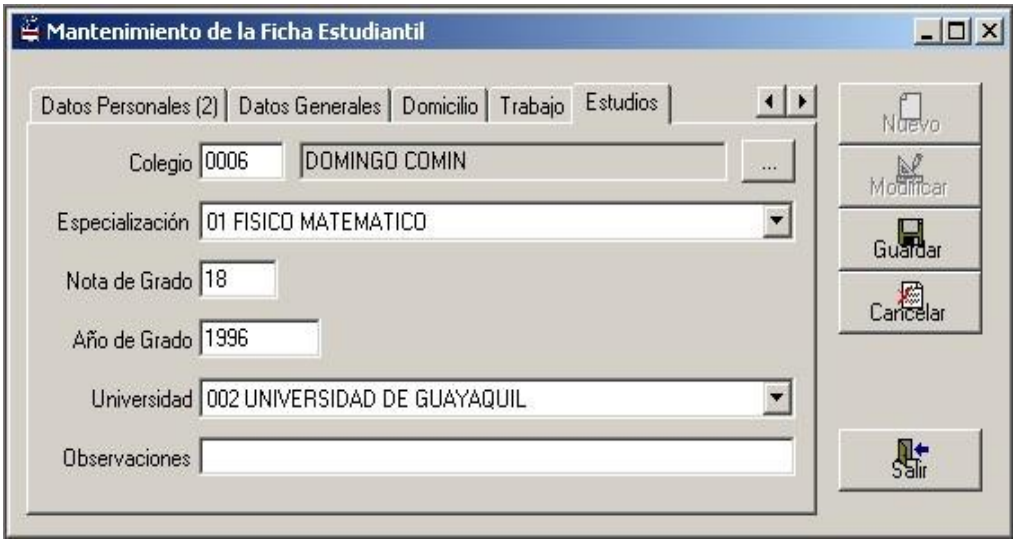


Figura 2.65. Ejemplo de Ficha Estudiantil- Estudios

2.18. Mantenimiento de Facultades

En dicho formulario el usuario se encarga del mantenimiento y consulta de las facultades.

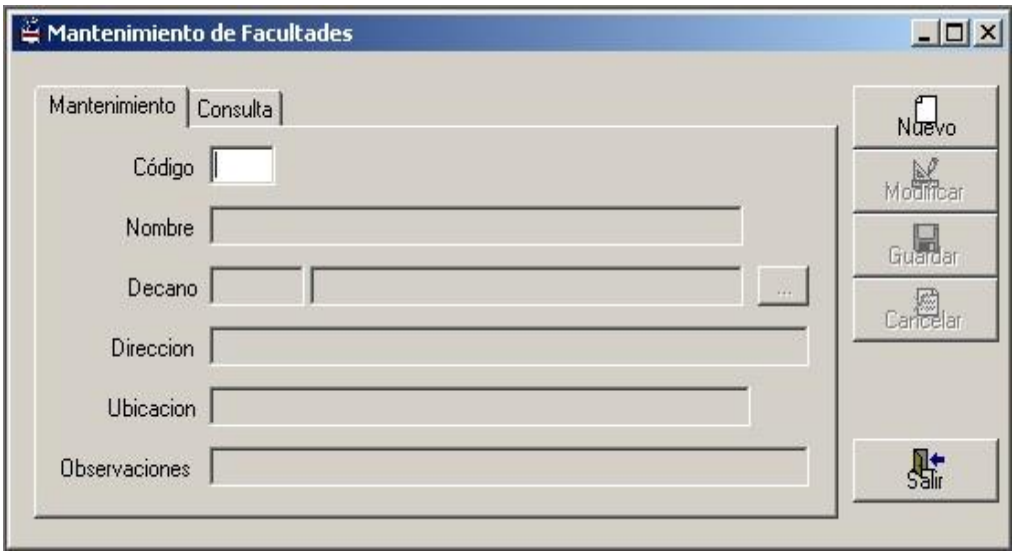


Figura 2.66. Pantalla de Mantenimiento de Facultades

Esta pantalla se tendrá lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código de la facultad (se habilita para la búsqueda de alguna facultad en particular).
- Una caja de texto llamada “Nombre” en la cual se digitará el correspondiente nombre de la facultad.
- Dos cajas de texto, una llamada “Decano” en la cual se digitará el código del decano, y otra caja de texto (lado derecho) donde aparecerá el nombre del decano de acuerdo al código que se haya ingresado.

- Una caja de texto llamada “Dirección” en la cual se digitará la dirección de la facultad.
- Una caja de texto llamada “Ubicación” en la cual se digitará el lugar donde se encuentra ubicado la facultad.
- Una caja de texto llamada “Observaciones” en la cual se detallará alguna observación de la facultad.

Como por ejemplo:

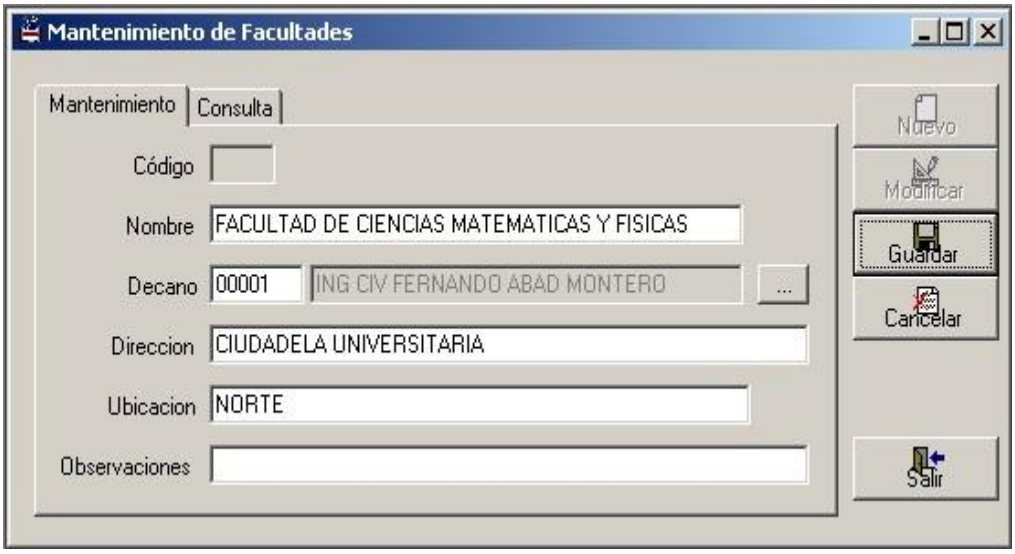


Figura 2.67. Ejemplo de Mantenimiento de Facultades

Con respecto a la pantalla de consulta de docentes tenemos:

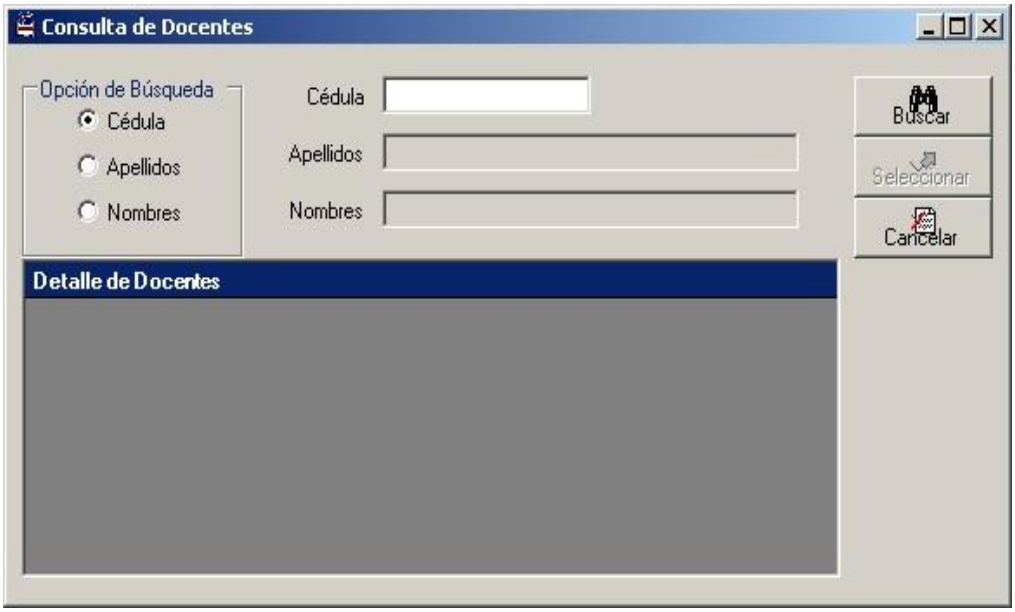


Figura 2.68. Pantalla de Consulta de Docentes

- Una tabla de “Opciones de búsqueda” en la que se desea buscar por: ”Cédula”, “Apellidos” o “Nombres” que correspondan al docente y se llenarán las cajas de texto dependiendo de la opción escogida.
- Una caja de texto llamada “Cédula” en la cual se digitará el número de cédula del docente.
- Una caja de texto llamada “Apellidos” en la cual se digitará los apellidos del docente.
- Una caja de texto llamada “Nombres” en la cual se digitará los nombres del docente.

- Un botón llamado “Buscar” con el cual se activa la búsqueda y se muestra en el grid.
- Un botón llamado “Seleccionar” en la cual se presiona cuando el usuario selecciona al docente.
- Un botón llamado “Cancelar” el cual se presiona cuando se quiere salir de la forma.

Como por ejemplo:

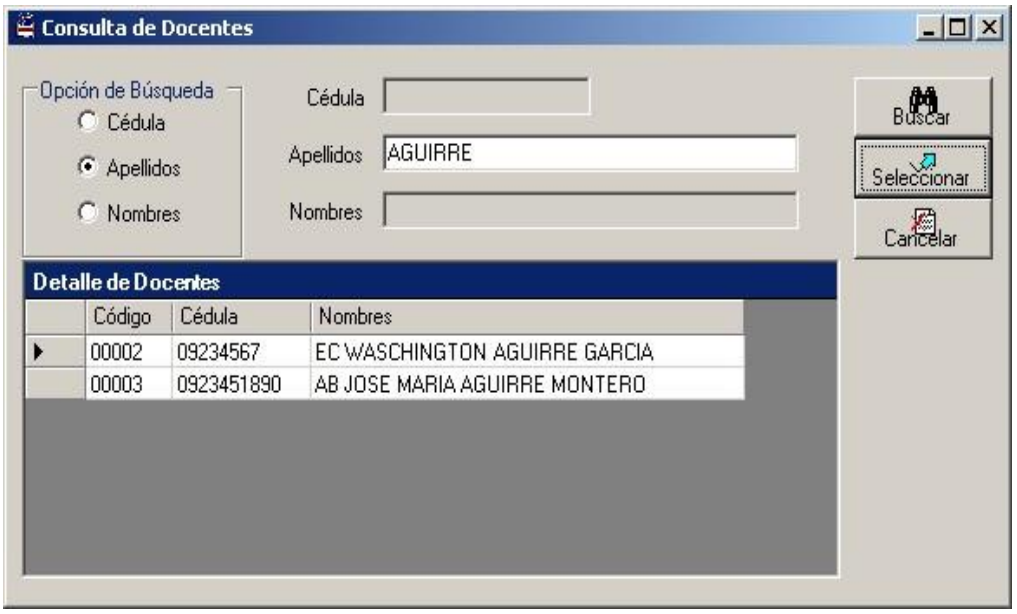


Figura 2.69. Ejemplo de Consulta de Docentes

En lo que concierne a consulta de la facultad se tiene lo siguiente:

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de las facultades que ingresaron en el sistema.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos que corresponden a las facultades, y esto se lo hace presionando el botón .

Como por ejemplo:

Figura 2.70. Ejemplo de Consulta de Facultades

2.19. Mantenimiento de Subunidades

En este formulario el usuario realiza el mantenimiento y consulta de las subunidades académicas. Se llaman subunidades académicas a

toda escuela, carrera o instituto que se ha creado dentro de una facultad.

Con lo concerniente al mantenimiento (1) tenemos:

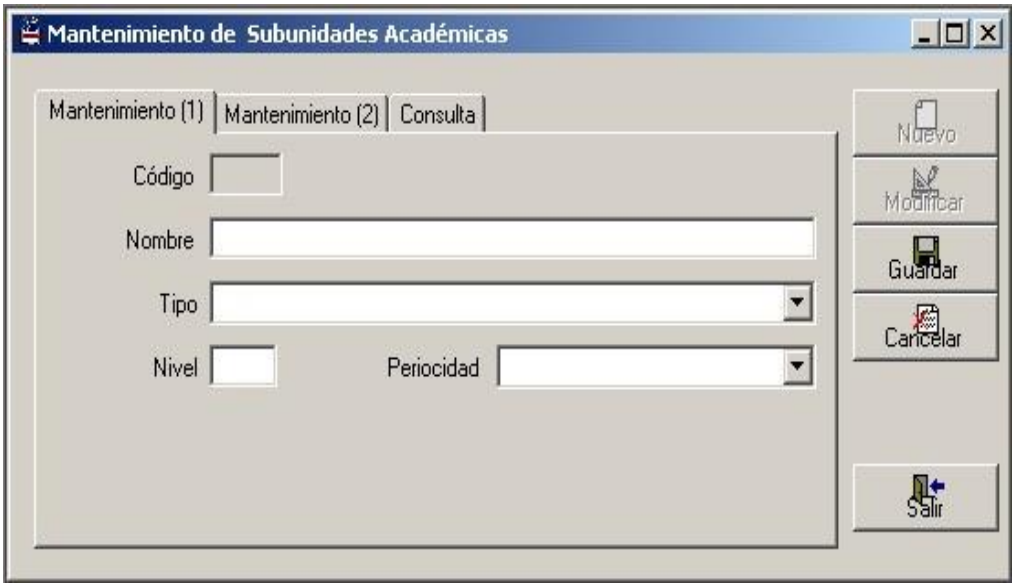


Figura 2.71. Pantalla de Mantenimiento (1) de Subunidades Académicas

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código de la subunidad académica (se habilita para la búsqueda de alguna subunidad específica).
- Una caja de texto llamada “Nombre” en la cual se digitará en nombre de la subunidad académica.

- Un combo llamado “Tipo” en la cual se escogerá el tipo de subunidad que corresponda, es decir, ya sea escuela, carrera o instituto.
- Una caja de texto llamada “Nivel” en la cual se digitará el nivel en que se haya la subunidad.
- Un combo llamado “Periodicidad” en la cual se escogerá los periodos de la subunidad académica.

Como por ejemplo:

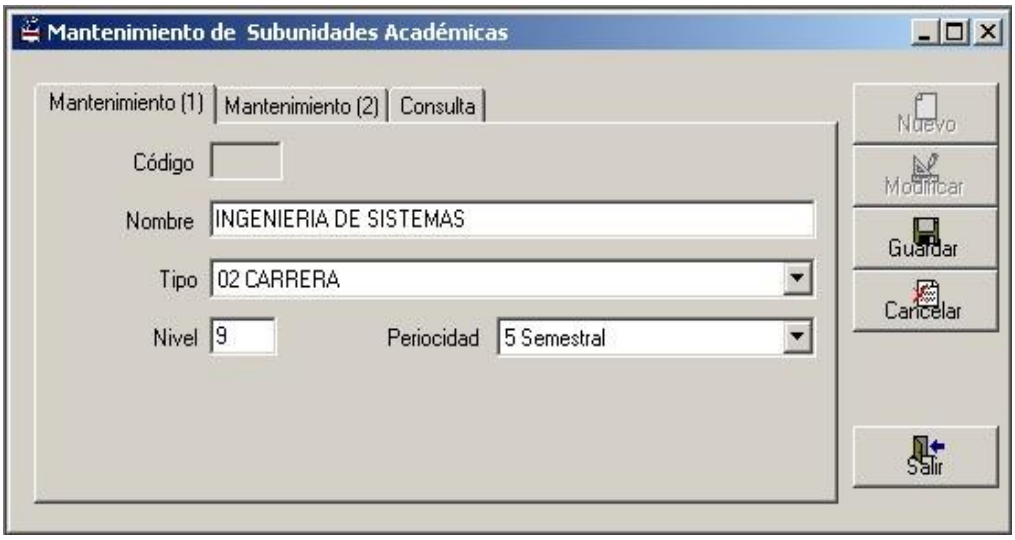


Figura 2.72. Ejemplo de Mantenimiento (1) de Subunidades Académicas

En lo que respecta al mantenimiento (2) tenemos:

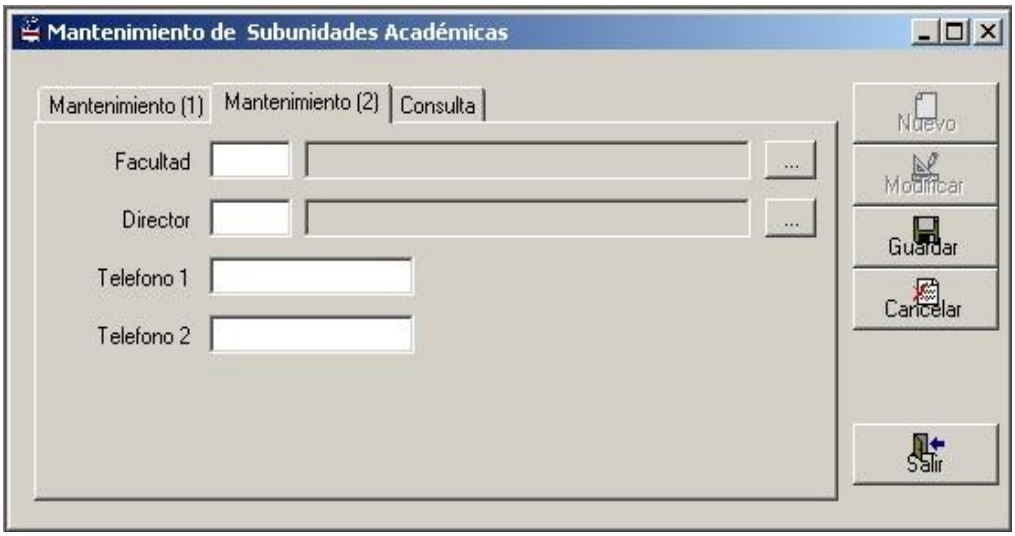


Figura 2.73. Pantalla de Mantenimiento (2) de Subunidades Académicas

- Dos cajas de textos, una llamada “Facultad” en la cual se digitará el código de la facultad que se asigne, y otra caja de texto (lado derecho) donde aparecerá el nombre de la facultad de acuerdo al código que se haya ingresado.
- Dos cajas de textos, una llamada “Director” en la cual se digitará el código del docente director de la subunidad, y otra caja de texto (lado derecho) donde aparecerá el nombre del director de acuerdo al código asignado.

- Dos cajas de textos llamada “Teléfono 1” y “Teléfono 2” en la cual se digitarán los teléfonos con la cual se puedan comunicar con la subunidad.

Como por ejemplo:

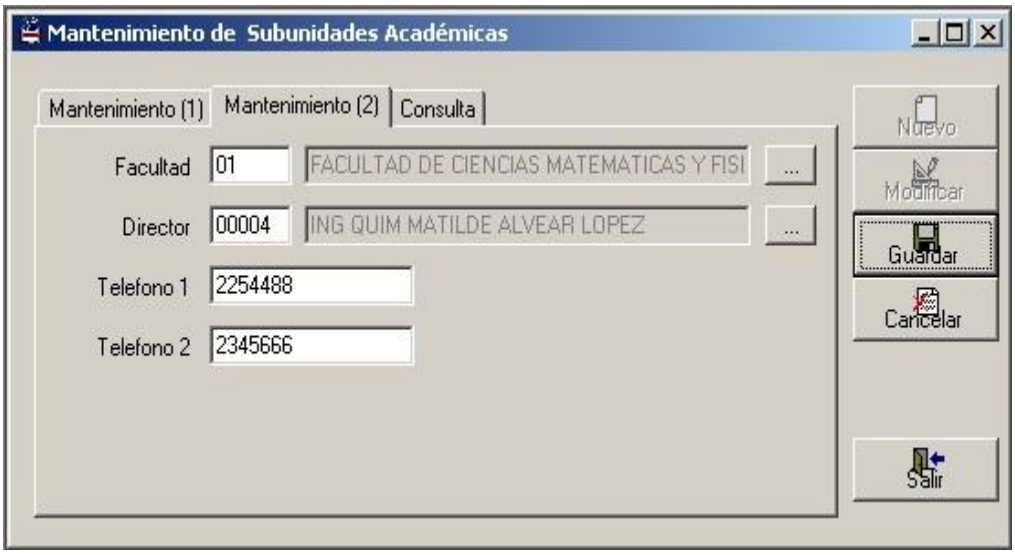


Figura 2.74. Ejemplo de Mantenimiento (2) de Subunidades Académicas

Cuando se activa el botón derecho que corresponde a la fila del “Facultad” aparecerá lo siguiente:

- Un “Grid de Consulta” en donde aparecerán los nombres de las facultades que se hayan ingresado al sistema.

- Un botón “Seleccionar” en donde el usuario selecciona el nombre de la facultad.
- Un botón “Cancelar” que se presiona cuando el usuario desee salir de la forma.

Como por ejemplo:

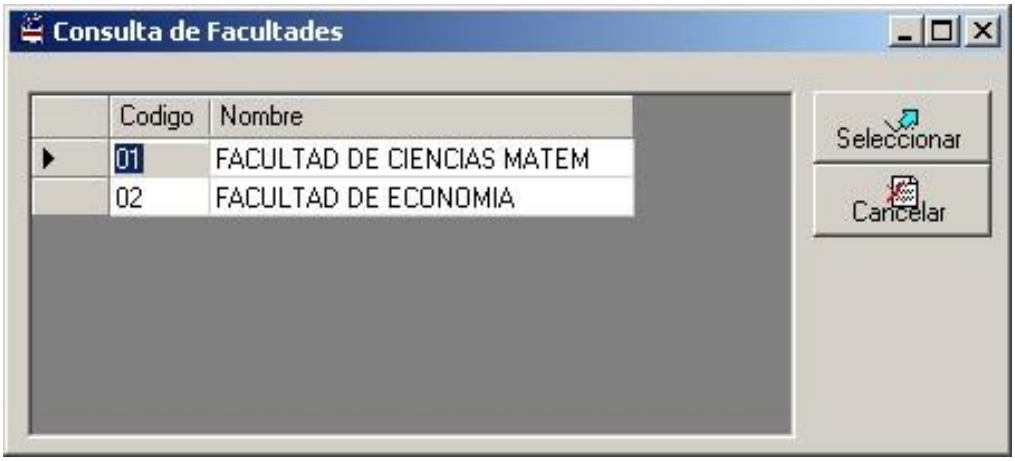


Figura 2.75. Ejemplo de Consulta de Facultades (Subunidad)

Cuando se presiona el botón derecho que corresponde a la fila de “Director” aparecerá una tabla de “Opciones de búsqueda” en la que se desea buscar por: ”Cédula”, “Apellidos” o “Nombres” que correspondan al docente y se llenarán las cajas de texto dependiendo de la opción escogida.

A lo que se refiere a consulta de subunidades académicas:

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitar el rango de la subunidades que exista en la universidad.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos que corresponda a la subunidad, se lo realiza presionando el botón .

Como por ejemplo:

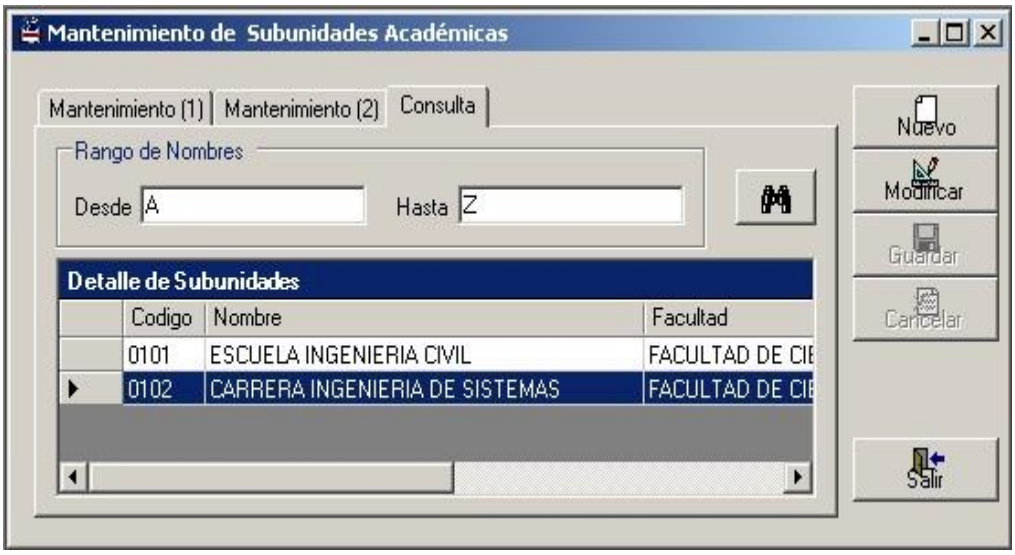


Figura 2.76. Ejemplo de Consulta de Subunidades Académicas

CAPITULO 3

3. PARAMETROS POR SUBUNIDAD

3.1. Mantenimiento de Áreas Académicas

Este formulario le permite al usuario realizar el mantenimiento de las áreas académicas.

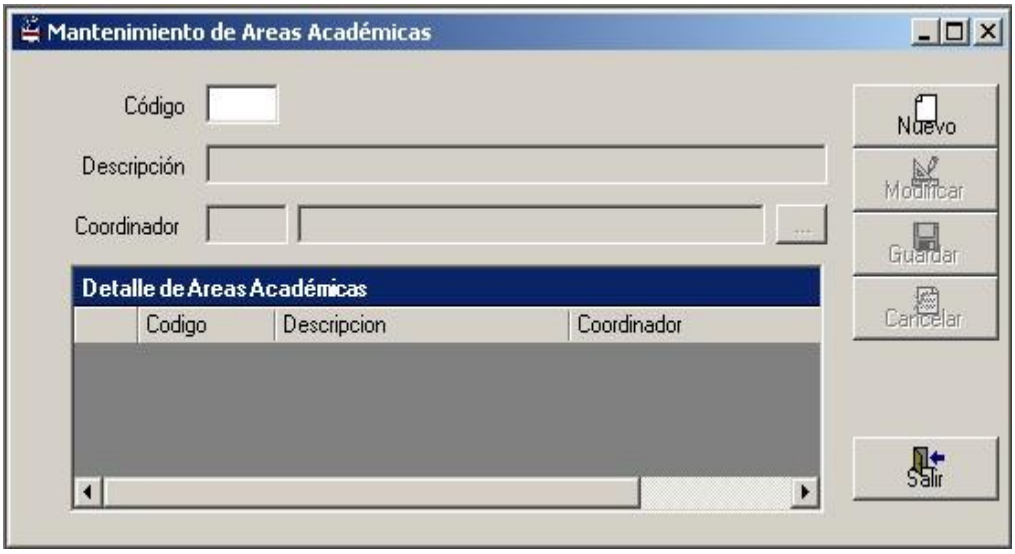


Figura 3.1. Pantalla de Áreas Académicas

En esta pantalla se muestra lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del área académica (se habilita para la búsqueda del área académica).
- Una caja de texto llamada “Descripción” en la cual se digitará la descripción del área académica.
- Dos cajas de textos llamada “Coordinador”, en la cual en la primera se digitará el código del coordinador del área académica, mostrándose en la segunda caja de texto el nombre del coordinador de acuerdo al código que se haya ingresado. Al presionar el botón que se ubica en parte derecha de la fila aparecerá un tabla de “Opciones de Búsqueda” en la que se desea buscar ya sea por ”Cédula”, “Apellidos” o “Nombres” que correspondan al docente y se llenarán las cajas de texto dependiendo de la opción escogida.

Como por ejemplo:

Figura 3.2. Ejemplo de Áreas Académicas

3.2. Mantenimiento de Valores de la Matrícula Global

En este formulario se detallan los nombres de las categorías y su correspondiente valor a pagar.

Figura 3.3. Acceso a Valores de la Matrícula Global

Ya entrando a la pantalla del valor de la matricula global se muestra lo siguiente:

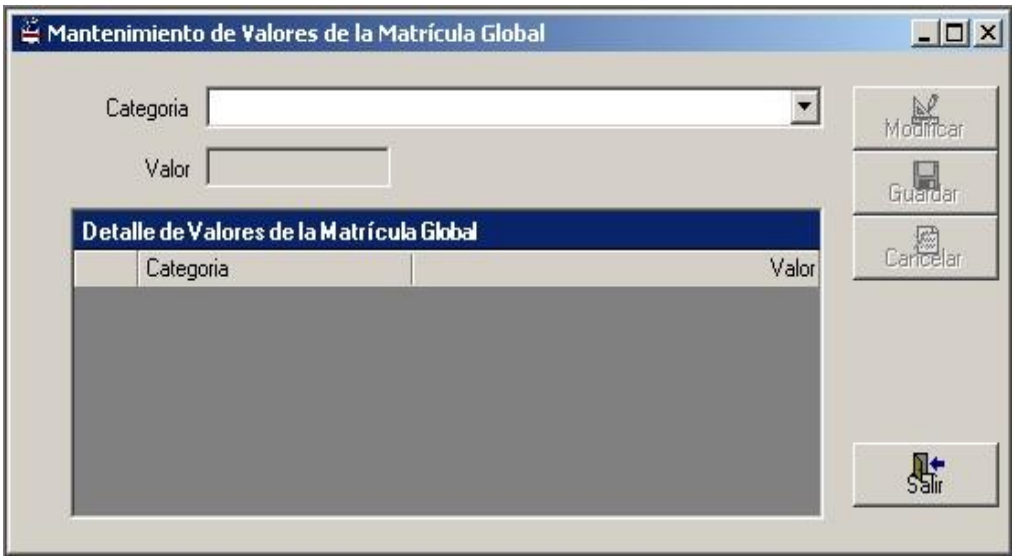


Figura 3.4. Pantalla de Valores de la Matrícula Global

- Un combo llamado “Categoría” en donde se seleccionará una categoría existente.
- Una caja de texto llamada “Valor” en la cual se digitará el valor a pagar por el estudiante.
- Un “Grid de Consulta” en donde aparecerá todos los datos ingresados al sistema.

Como por ejemplo:

Figura 3.5. Ejemplo de Valores de la Matrícula Global

3. 3. Mantenimiento de Valores de la Matrícula por Asignatura

Este formulario el usuario hará el respectivo mantenimiento de los valores de la matrícula por asignatura dependiendo de la categoría.

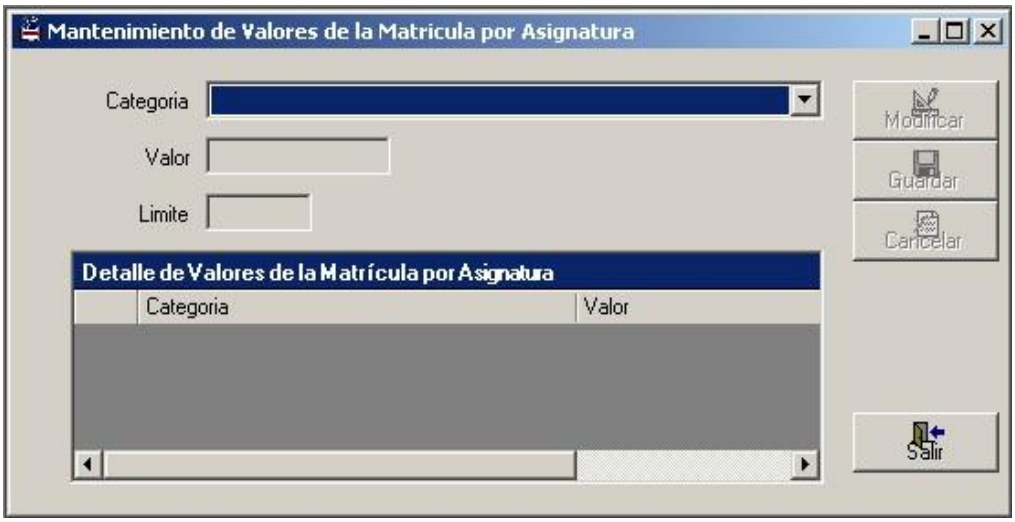


Figura 3.6. Pantalla de Valores de la Matrícula por Asignatura

Esta pantalla presenta los siguientes puntos:

- Un combo llamado “Categoría” en donde se seleccionará una categoría existente.
- Una caja de texto llamada “Valor” en la cual se digitará el valor de la matrícula.
- Una caja de texto llamada “Límite” en la cual se digitará el número máximo de materias que se considera para este tipo de matrícula.
- Un “Grid de Consulta” en la que mostrará todos lo valores de la matrícula por asignatura.

Como por ejemplo:

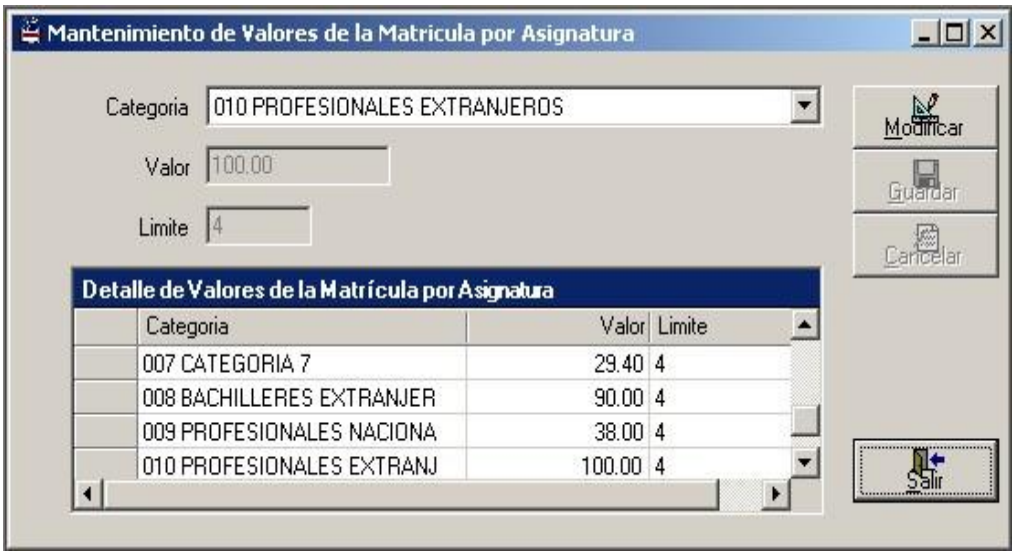


Figura 3.7. Ejemplo de Valores de la Matrícula por Asignatura

3. 4. Mantenimiento de Incremento de Matrículas

En este formulario el usuario realizará el mantenimiento de incremento de las matrículas.

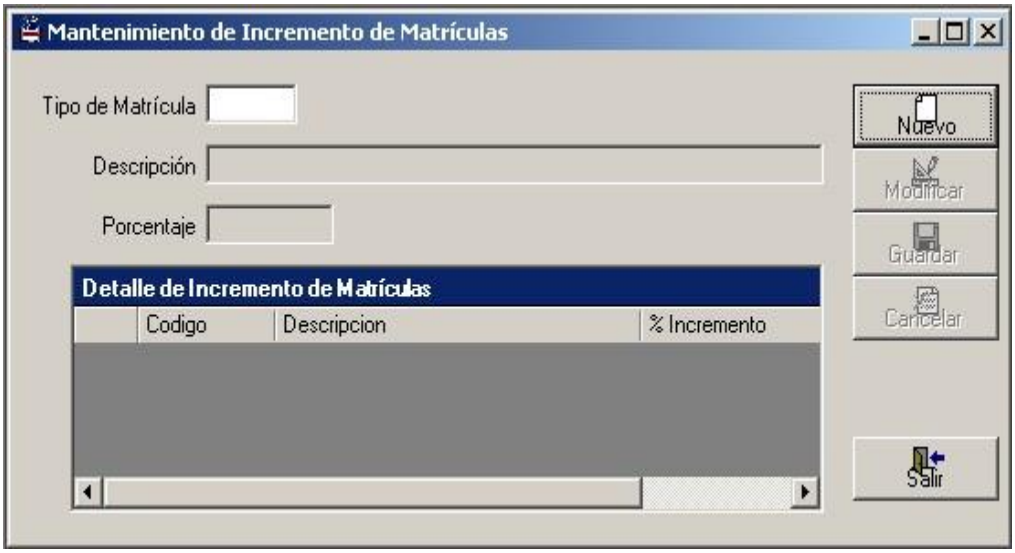


Figura 3.8. Pantalla de Incremento de Matrículas

En esta pantalla se tendrá lo siguiente:

- Una caja de texto llamada “Tipo de Matrícula” en donde se digitará el tipo de matrícula que corresponda.
- Una caja de texto llamada “Descripción” el cual se digitará la razón del incremento (primera matrícula, segunda matrícula, etc).
- Una caja de texto llamada “Porcentaje” en la cual se digitará el porcentaje de incremento que se le hará al estudiante.

- Un “Grid de Consulta” en donde se detallarán todos los incrementos de las matrículas por haber.

Como por ejemplo:

Figura 3.9. Ejemplo de Incremento de Matrículas

3. 5. Mantenimiento de Tipos de Cursos

Aquí en este formulario el usuario deberá de efectuar el respectivo mantenimiento y consulta de los tipos de cursos.

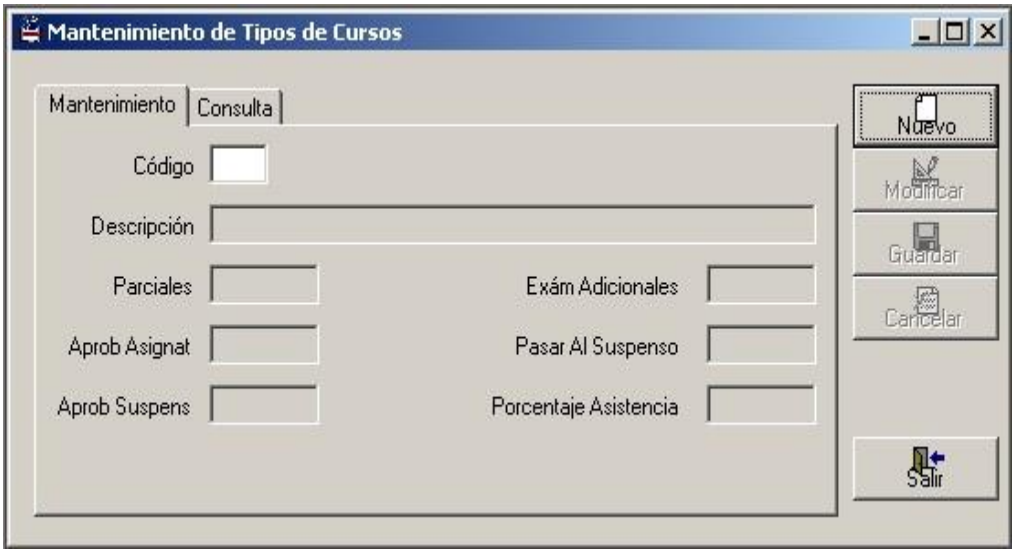


Figura 3.10. Pantalla de Tipos de Cursos

Con lo referente a mantenimiento tenemos:

- Una caja de texto llamada “Código” en la cual se digitará el código del área académica (se habilita para la búsqueda del área académica).
- Una caja de texto llamada “Descripción” en la cual se digitará el tipo de curso, los cuales pueden ser : seminarios de arrastre, cursos de invierno, cursos normales, preuniversitarios, actividades extracurriculares, etc.
- Una caja de texto llamada “Parciales” en la cual se digitará el número de parciales que tiene el alumno en la unidad académica.

- Una caja de texto llamada “Aprob. Asignat” en la cual se digitará la mínima nota que el estudiante necesita para aprobar la asignatura.
- Una caja de texto llamada “Aprob. Suspens” en la cual se digitará la mínima nota que el estudiante necesita para aprobar el suspenso.
- Una caja de texto llamada “Exám. Adicionales” en la cual se digitará la cantidad de exámenes adicionales que el estudiante puede rendir para aprobar la asignatura.
- Una caja de texto llamada” Pasar al Suspenso” en la cual se digitará la mínima nota que el estudiante necesita para pasar al suspenso.
- Una caja de texto llamada “Porcentaje Asistencia” en la cual se digitará el porcentaje mínimo que debe tener el estudiante para rendir las pruebas parciales.

Como por ejemplo, supongamos que dentro de la subunidad donde usted trabaja, normalmente en cada semestre hay dos parciales y un examen de recuperación. En los dos parciales se tiene que reunir como mínimo 14 puntos para aprobar cada asignatura; caso contrario, con un mínimo de 10 se puede pasar al examen de recuperación. La mínima nota para aprobar dicho examen es 7. Respecto al mínimo

porcentaje de asistencias aceptado es del 75%. Eso es lo que se ingresa en la pantalla siguiente:

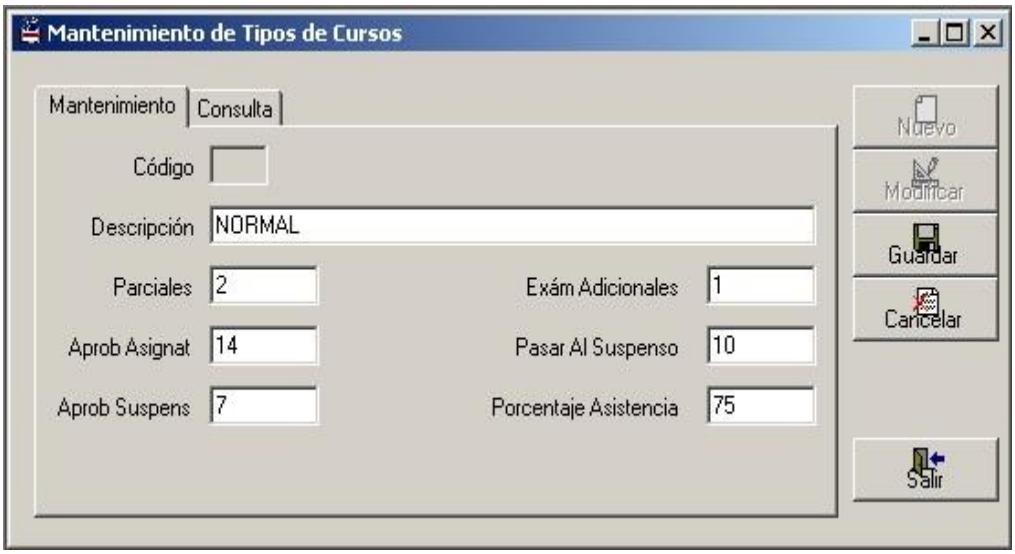


Figura 3.11. Ejemplo (1) de Tipos de Cursos

Así mismo, supongamos que existan seminarios de arrastre los cuales tienen sus propias reglas: tienen un solo parcial y un examen de recuperación, que lo mínimo para aprobar la asignatura sea 7 y el mínimo para pasar al suspenso sea 5. Estos parámetros se ingresarán de la siguiente manera:

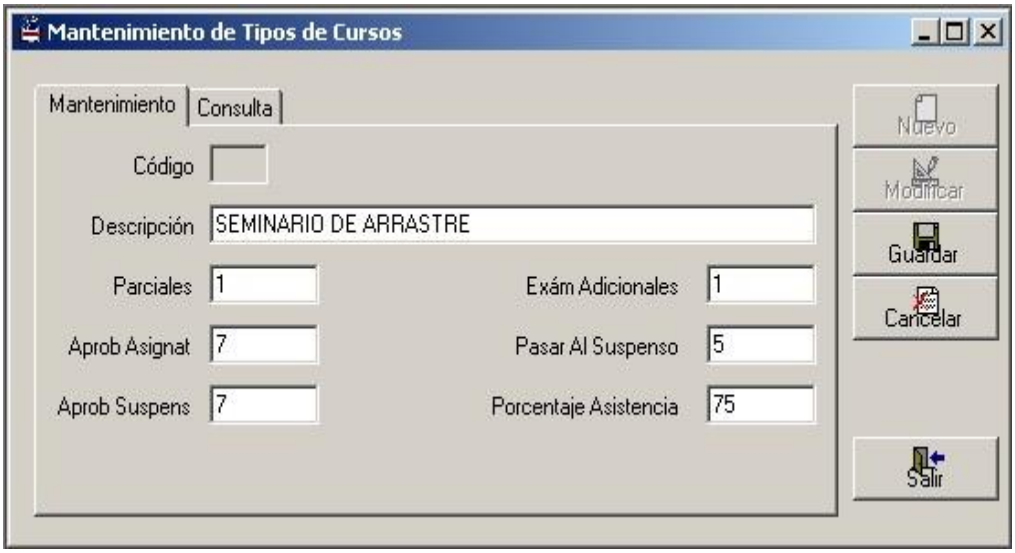


Figura 3.12. Ejemplo (2) de Tipos de Cursos

A lo que se refiere a consulta se tiene:

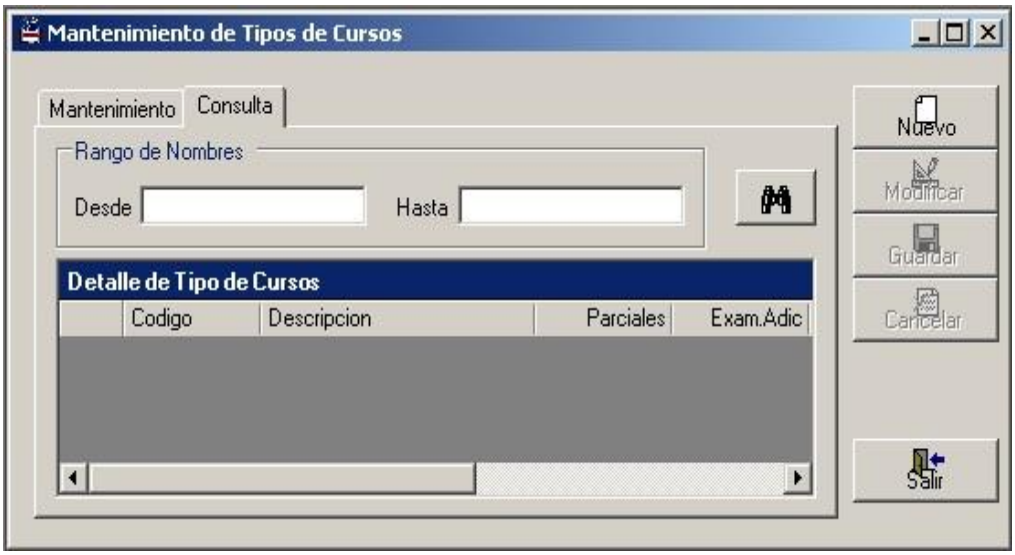


Figura 3.13. Consulta de Tipo de Cursos

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de los tipos de cursos que ingresaron al sistema.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos que corresponden a los tipos de cursos, y esto se lo hace presionando el botón .

Como por ejemplo:

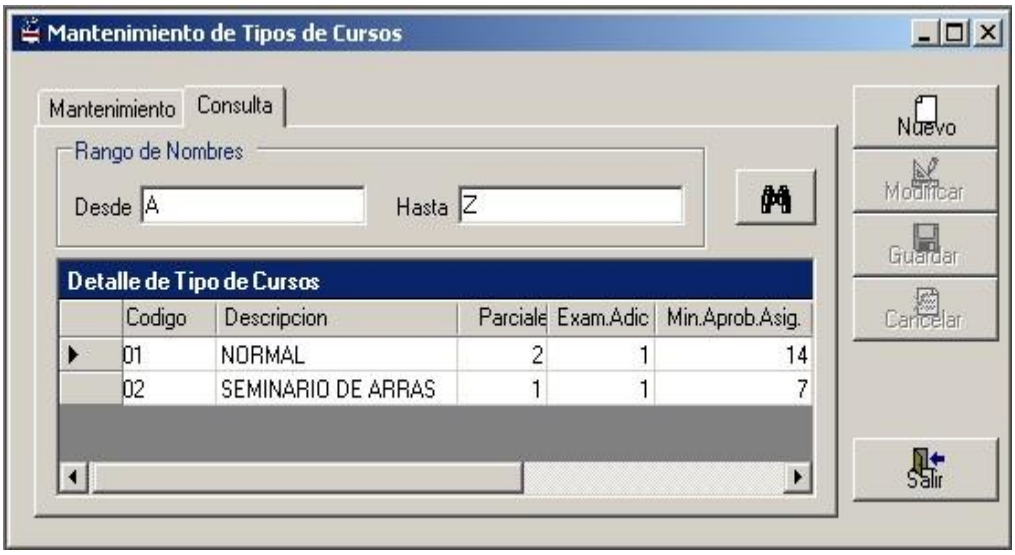


Figura 3.14. Ejemplo de Consulta de Tipos de Cursos

3. 6. Mantenimiento de Tipos de Calificaciones

Este formulario el usuario debe de realizar el respectivo mantenimiento de los tipos de calificaciones que hay en cada

subunidad, es decir, primer parcial, segundo parcial, tercer parcial, suspenso, mejoramiento, etc.

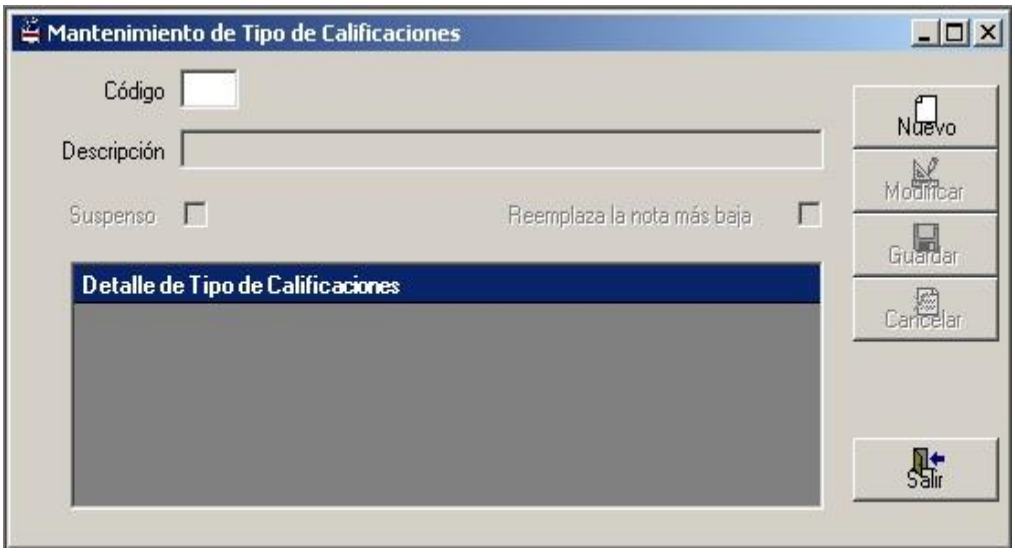


Figura 3.15. Pantalla de Tipo de Calificaciones

En esta pantalla tenemos lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el correspondiente código del tipo de calificación (se habilita para la búsqueda del respectivo tipo de calificación).
- Una caja de texto llamada “Descripción” en la cual se digitará el tipo de calificación .
- Una casilla de verificación llamada “Suspenso” que indica si esa calificación corresponde al examen de suspenso.

- Otra casilla de verificación llamada “Reemplaza la nota más baja” que indica que esa calificación sea de mejoramiento entre todas las notas parciales.

Como por ejemplo:

Figura 3.16. Ejemplo de Tipo de Calificaciones

CAPITULO 4

4. CONTROL ACADEMICO

4.1. Mantenimiento de Asignatura

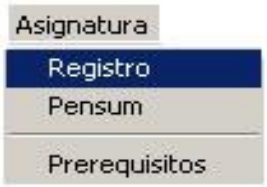


Figura 4.1. Acceso a Mantenimiento de Asignaturas

En esta pantalla el usuario se encarga de realizar el mantenimiento y consulta de asignaturas.

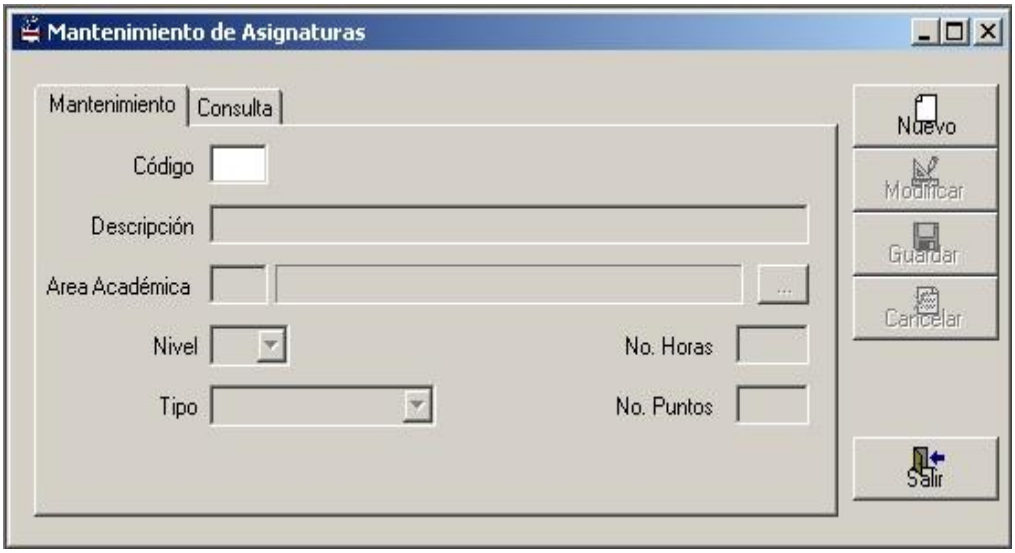


Figura 4.2. Pantalla de Mantenimiento de Asignaturas

Se debe tomar en cuenta que antes de ingresar una asignatura se debe ingresar por lo menos un área académica.

En esta pantalla se tiene lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el código de la asignatura (se habilita para la búsqueda de alguna asignatura específica).
- Una caja de texto llamada “Descripción” en la cual se digitará la descripción de dicha asignatura.
- Una caja de texto llamada “Area Académica” en la cual se digitará el código del área académica. En su lado derecho al presionar el botón aparecerá otra caja de texto en la cual se

mostrará la descripción del área académica según el código que se haya asignado.

- Un combo llamado “Nivel” en donde se escogerá el nivel al que pertenece la asignatura.
- Un combo llamado “Tipo” en la que se escogerá si la materia es obligatoria o selectiva.
- Una caja de texto llamada “No. Horas” en la cual se digitará el número de horas dentro de una semana.
- Una caja de texto llamada “No. Puntos” en la cual se digitará la cantidad de puntos que el estudiante va ganando cuando aprueba la materia. Hay carreras que tienen como requisito para egresar una cantidad determinada de puntos entre materias obligatorias y selectivas.

Como por ejemplo:

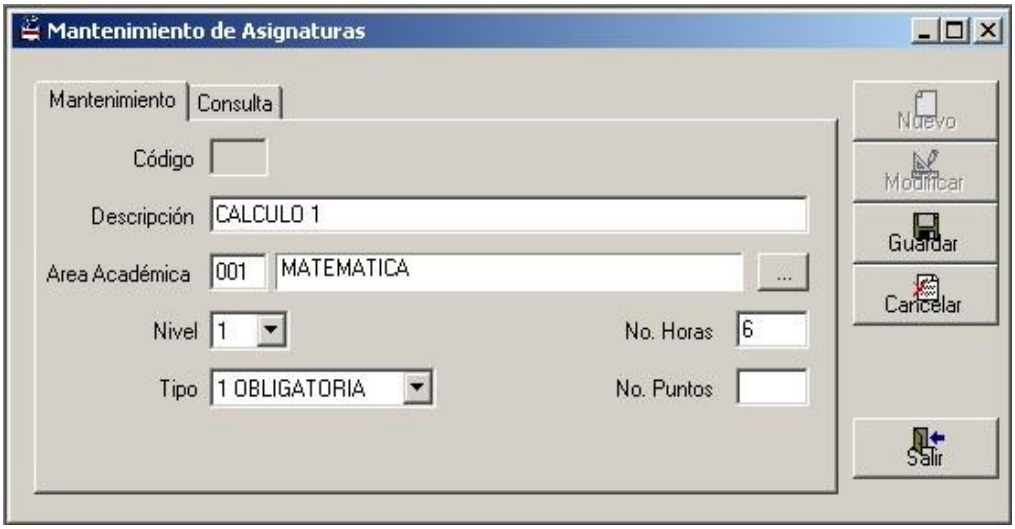


Figura 4.3. Ejemplo de Asignaturas

Con respecto a consulta tenemos:

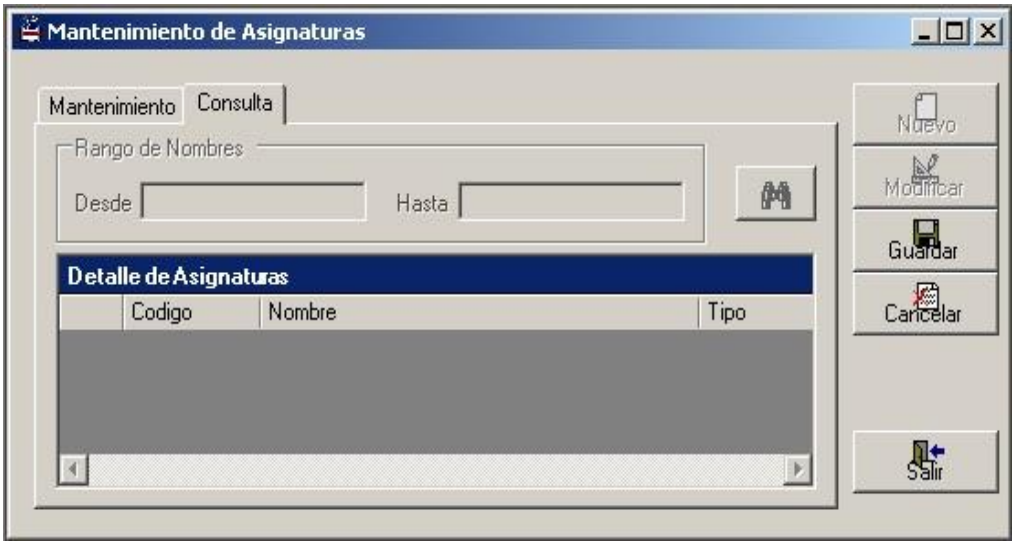


Figura 4.4 Consulta de Asignaturas

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de nombres de las asignaturas de la subunidad que ingresaron al sistema.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos que corresponden a la signaturas de la subunidad, y esto se lo hace presionando el botón .

Como por ejemplo:

Figura 4.5. Ejemplo de Consulta de Asignaturas

4.2. Mantenimiento de Pénsum

Este formulario el usuario realiza el mantenimiento y consulta de pénsum.

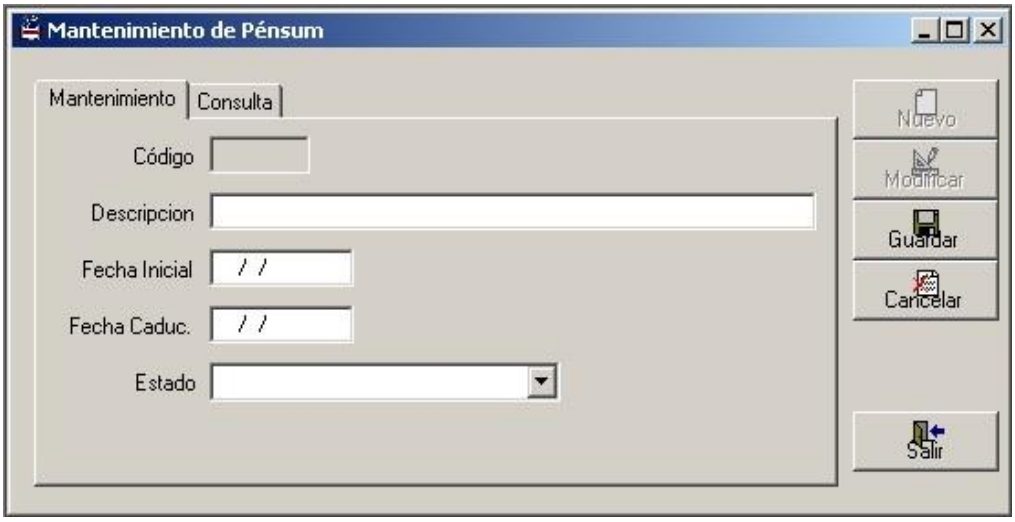


Figura 4.6. Pantalla de Mantenimiento de Pénsum

En dicha pantalla se tiene lo siguiente:

- Una caja de texto llamada “Código” en la cual se digitará el código del pénsum (se habilita para la búsqueda del pénsum).
- Una caja de texto llamada “Descripción” en la cual se digitará la descripción del pénsum.
- Una caja de texto llamada “Fecha Inicial” en la cual se digitará la fecha en que se puso en vigencia el pénsum.
- Una caja de texto llamada “Fecha Caduc.” En la cual se digitará la fecha en que feneció dicho pénsum.
- Un combo llamado “Estado” en la cual se escogerá el estado actual del pénsum, estas pueden ser: vigente, inactivo y espera.

Como por ejemplo:

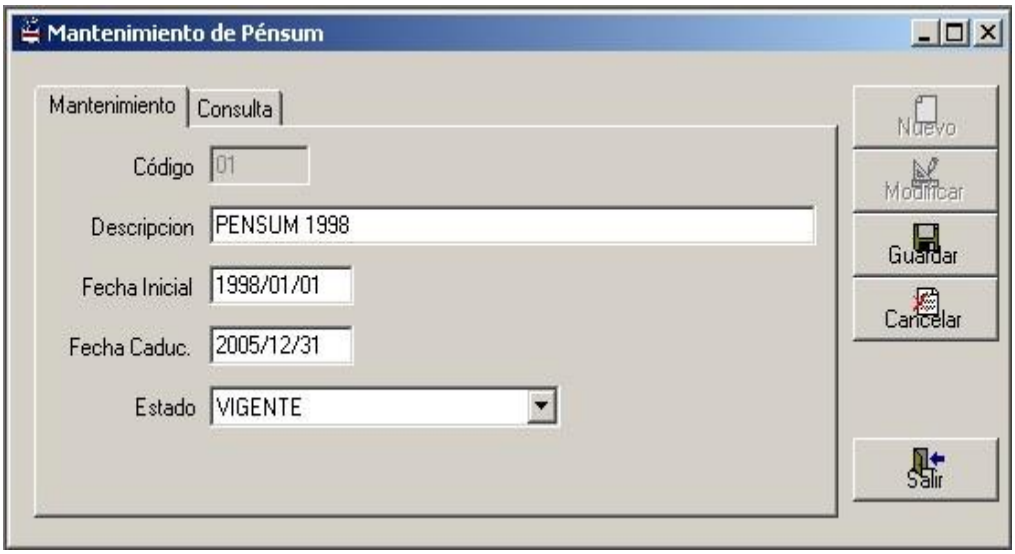


Figura 4.7. Ejemplo de Mantenimiento de Pénsum

A lo que concierne a consulta de pénsum se tiene lo siguiente:

- Dos cajas de texto llamadas “Desde” y “Hasta” en la cual se digitará el rango de fechas de inicio del pénsum.
- Un “Grid de Consulta” que sirve para la búsqueda de los datos correspondiente al pénsum, y esto se lo hace presionando el botón .

Como por ejemplo:

Figura 4.8. Ejemplo de Consulta de Pénsum

Cabe destacar que cuando se crea dentro del sistema el primer pénsum, se deberá ingresar a continuación todos los prerrequisitos para cada asignatura. No se recomienda crear un nuevo pénsum sin haber ingresado los prerrequisitos del pénsum anterior.

4.3. Mantenimiento de Prerrequisitos

En este formulario el usuario realiza el mantenimiento y consulta de los prerrequisitos por cada asignatura, de acuerdo a un pénsum creado.

En dicha pantalla tenemos lo siguiente:

Figura 4.9. Pantalla de Mantenimiento de Prerrequisitos

- En “Asignatura” se va a digitar el código de la materia de la que se señalarán los prerrequisitos, en el siguiente ejemplo, deseamos ingresar los prerrequisitos de Calculo 2:

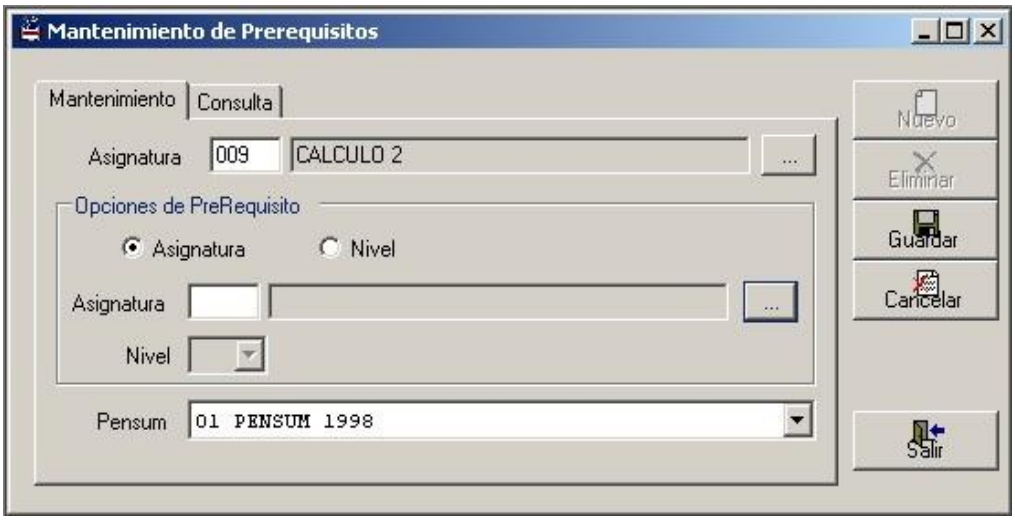


Figura 4.10. Ejemplo de Prerrequisito- Paso 1

- En el cuadro llamado “Opciones de Prerrequisito” se va a indicar el requisito de la asignatura ingresada. Dependiendo del pénsum los requisitos para ver cada materia es haber aprobado o bien un nivel (curso) o una o más asignaturas. En el ejemplo anterior, para ver Cálculo 2 se requiere haber aprobado Cálculo 1 y Matemáticas Discretas. Ingresamos el primer requisito que es Cálculo 1, para esto seleccionamos dentro de “Opciones de Prerrequisito” el ítem “Asignatura” y a continuación escribimos el código del requisito. En caso de no conocer el código de una materia, se puede acceder a la pantalla de Consulta de Asignaturas presionando el botón que está a la derecha.

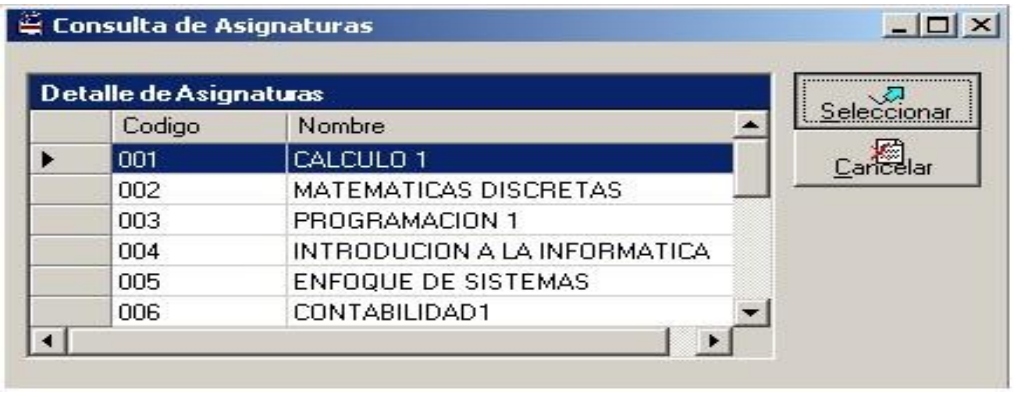


Figura 4.11. Seleccionando un Prerrequisito-Paso 2

- Un combo llamado “Pénsum” el cual se seleccionará el pénsum creado al cual corresponde el prerrequisito ingresado.

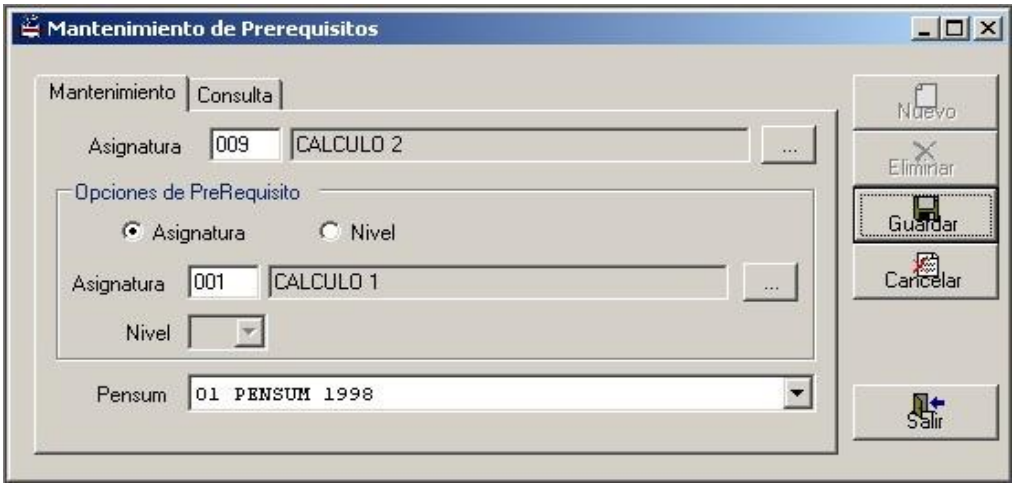


Figura 4.12. Ejemplo de Prerrequisito- Paso 3

A lo que respecta a consulta se tiene:

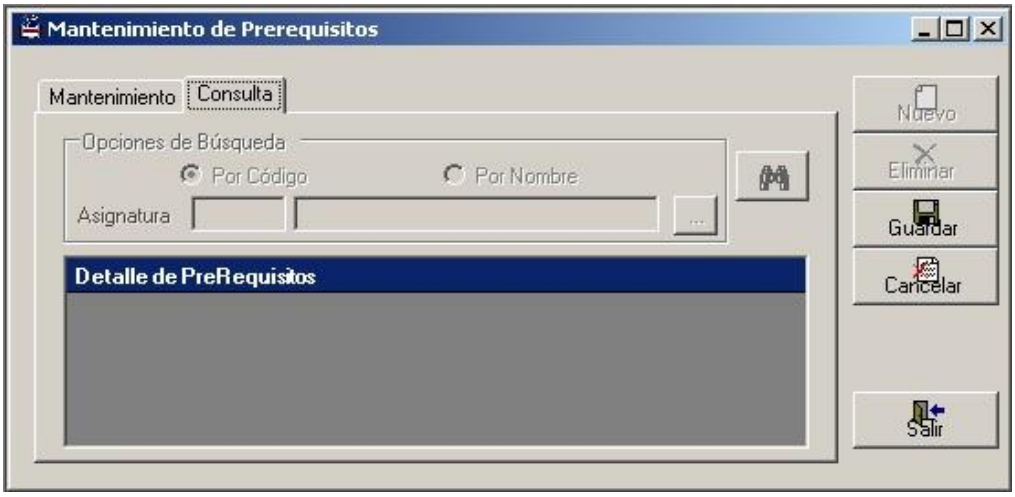


Figura 4.13. Consulta de Prerrequisito

- Una cuadro llamado “Opciones de Búsqueda” en donde se desea especificar el prerrequisito ya sea por el código o por el

nombre de la asignatura. Si acceso por ***código*** se activará la caja de texto "Asignatura" donde se digitará el código de la materia que se va a cursar, si acceso por ***nombre*** se activará la otra caja de texto en donde se pondrá el nombre de la materia a cursar.

- Un "Grid de Consulta" donde se detallarán todos los prerrequisitos de la materia ingresada, y esto se lo realiza presionando el botón .

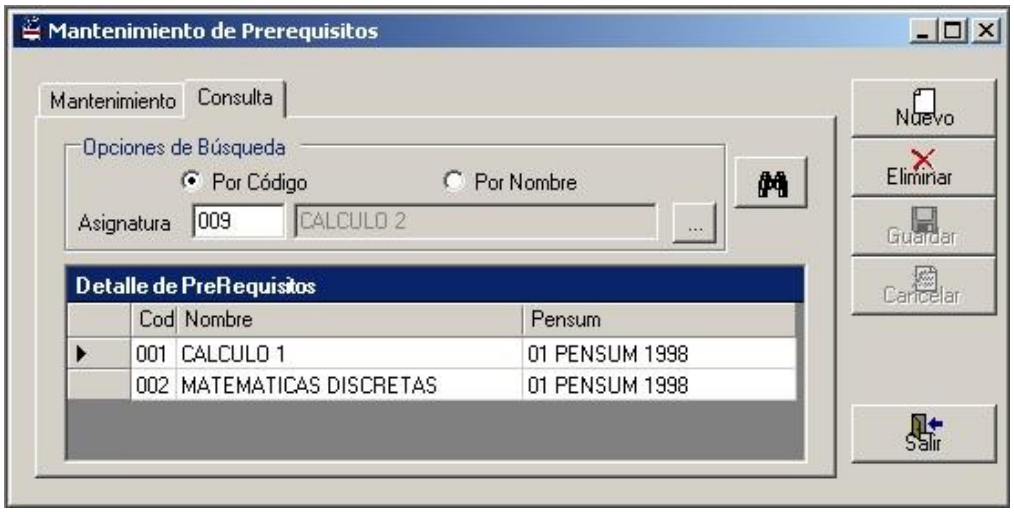


Figura 4.14. Ejemplo de Consulta de Prerrequisito

Cabe destacar que no se puede crear ni eliminar prerrequisitos dentro de un pénsum que está en vigencia (utilizado actualmente por un estudiante).

CAPITULO 5

5. PERÍODOS LECTIVOS

5.1. Creación de Períodos Lectivos

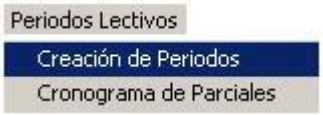


Figura 5.1. Acceso a Períodos Lectivos

Esta pantalla sirve para fijar el inicio y el fin de cada período lectivo. Existen cuatro estados para cada período:

- Espera: Cuando el período lectivo aún no se ha abierto.
- Inicio de Matriculación: Cuando se distribuye las asignaturas a los docentes, se crean los horarios y se inicia la inscripción de los estudiantes.
- Cierre de Matriculación: Cuando ya no se aceptan más inscripciones y todos los grupos ya están abiertos.
- Cierre de Período: Cuando finaliza el período lectivo.

Un período lectivo se encuentra definido por tres campos: el año, ciclo, y el tipo de curso.

Figura 5.2. Creación de Períodos Lectivos

Cuando la combinación de años, ciclo y tipo de curso forman un período ya creado se habilita el botón modificar y los datos son cargados a las cajas de textos, caso contrario se habilita el botón nuevo.

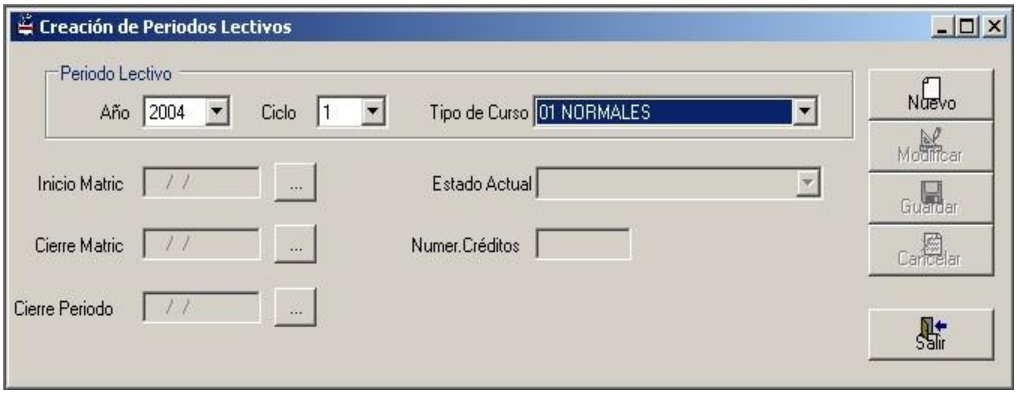


Figura 5.3. Ejemplo de Creación de Períodos Lectivos- Paso 1

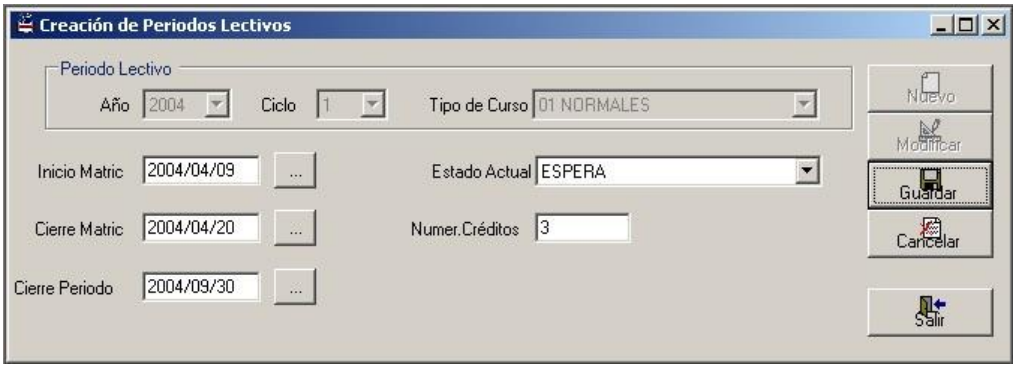


Figura 5.4. Ejemplo de Creación de Períodos Lectivos- Paso 2

Se recomienda hacer un cronograma de actividades con fechas tentativas antes de iniciar un período lectivo para poder guardar toda esa información en esta pantalla y en la de Entrega de Notas Parciales. Así mismo se recomienda, estar pendiente de las fechas que se ingresen porque va a ser muy importante cuando el usuario tenga que cambiar el estado del período lectivo. Las fechas a ingresar son: fecha de inicio de matriculación, fecha de cierre de matriculación y fecha de cierre de período.

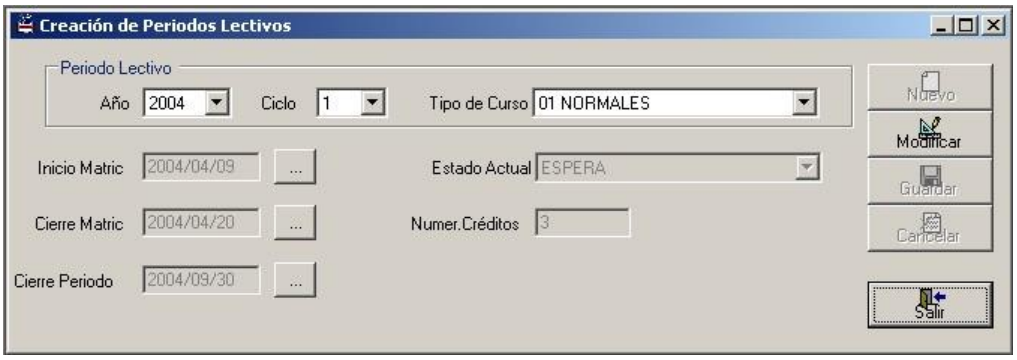


Figura 5.5. Ejemplo de Creación de Períodos Lectivos- Paso 3

El sistema no va a permitir que cambie unas fechas ya ingresadas en un período lectivo existente por otra fecha menor. Tampoco va a permitir modificar una fecha guardada si la fecha actual es mayor que ella. Los estados son controlados por medio de las fechas digitadas.

En la siguiente pantalla al presionar el botón “Modificar” se observan que las tres cajas de textos de las fechas de inicio y cierre de matriculación, así como el cierre de período se encuentran deshabilitadas por las razones anteriormente explicadas.

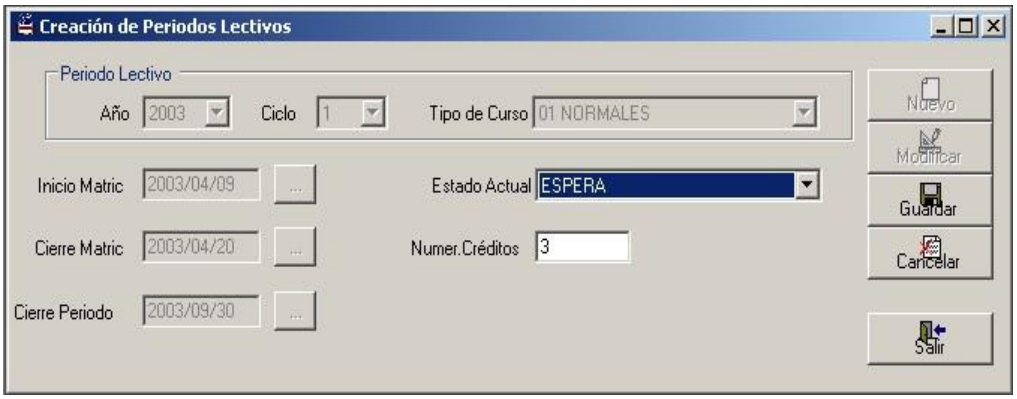


Figura 5.6. Ejemplo de Modificación de Períodos Lectivos

En la siguiente figura se muestra un ejemplo donde se ha querido modificar el estado de un período lectivo a uno que no corresponde dentro de las fechas anteriormente guardadas; la fecha de cierre de ese período es 2003/09/30, por lo tanto, el estado actual no debería ser Espera.

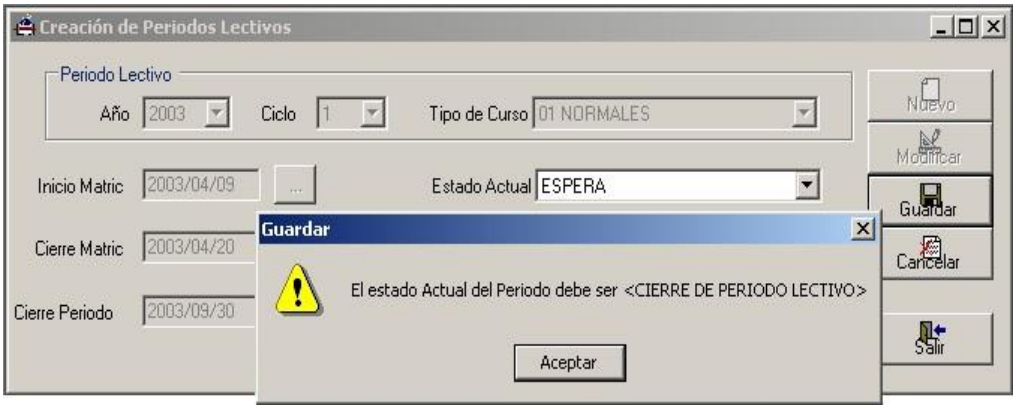


Figura 5.7. Ejemplo de Modificación de Períodos Lectivos- Mensaje

Objeto Calendario

Cuando se presiona el botón que está a la derecha de cada caja de texto donde se ingresan las fechas, aparece un objeto llamado Calendario.

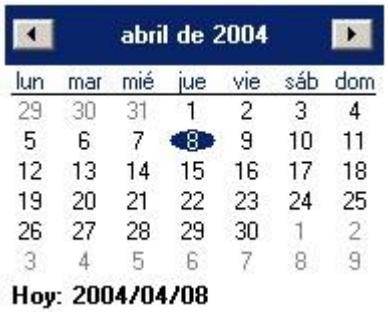


Figura 5.8. Objeto Calendario

Para cambiar de fechas dentro del mismo mes con el Mouse me ubico justamente en el día deseado. Para cambiar de meses, se tiene que presionar en las flechas hacia delante o hacia atrás, o sino damos clic

en el mes y saldrá una lista donde podremos seleccionar el mes deseado.

Figura 5.9. Cambio de Meses en Objeto Calendario

Para el cambio de año se procede de la siguiente manera como se muestra en la figura:

Figura 5.10. Cambio de Año en Objeto Calendario

5.2. Cronograma de Entrega de Notas Parciales

En esta pantalla se ingresarán las fechas tentativas en las que un docente podría entregar calificaciones por cada parcial.

Figura 5.11. Cronograma de Entrega de Notas Parciales

Por cada período lectivo se deben ingresar las fechas tope de entrega de notas de todos los parciales.

Figura 5.12. Ejemplo de Cronograma de Entrega de Notas Parciales

CAPITULO 6

6. CONTROL DE HORARIOS

6.1. Distributivo de Asignaturas a Docentes

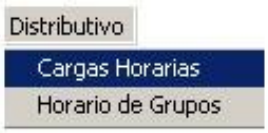


Figura 6.1. Acceso a Control de Horarios

La pantalla del Distributivo de Asignaturas a Docentes consiste en especificar las asignaturas, los grupos y el número de estudiantes que se les asigne a cada docente.

Entrando ya en la pantalla para procesar la información realizamos lo siguiente:

- Se digitará el código del docente asignado o se hará la respectiva búsqueda presionando el botón derecho.

- Se actualizará la información correspondiente al “Período Lectivo”, es decir, a lo que respecta al año, ciclo y tipo de curso. Cabe destacar que en esta pantalla sólo se considerarán los períodos lectivos en estado “Inicio de Matriculación”
- Finalmente, se le asignará para cada docente las respectivas asignaturas , grupos y el número de estudiantes que estará a cargo durante dicho período.

A continuación daremos un ejemplo del manejo de dicha pantalla:

Supongamos que dentro del período lectivo actual, para los dos primeros semestres se han abierto los siguientes grupos:

Primer Semestre	***Segundo Semestre***
S1A	S2A
S1B	S2B
S1C	S2C
S1D	S2D

Vamos a considerar el ingreso de la carga horaria del docente que se detalla a continuación:

Ing. José Huayamabe	
Asignaturas	**Grupos**
Cálculo 1	S1A, S1B
Matemáticas Discretas	S1C, S1D
Cálculo 2	S2A, S2D

El manejo de esta pantalla será de la siguiente manera:

- Seleccionamos un período lectivo abierto.
- Damos clic en el botón “Nuevo” y digitamos el respectivo código del docente o bien presionamos el botón derecho para la búsqueda del mismo.
- Presionamos el botón “Nueva Asignatura” para ingresar cada asignatura que va a impartir. Se ingresan a continuación la materia, grupo y el cupo asignado.
- Se da clic en el botón para guardar la carga horaria asignada. Para asignar más materias al mismo docente se repiten estos dos últimos pasos. Para cambiar de docente se da clic en el botón Nuevo.

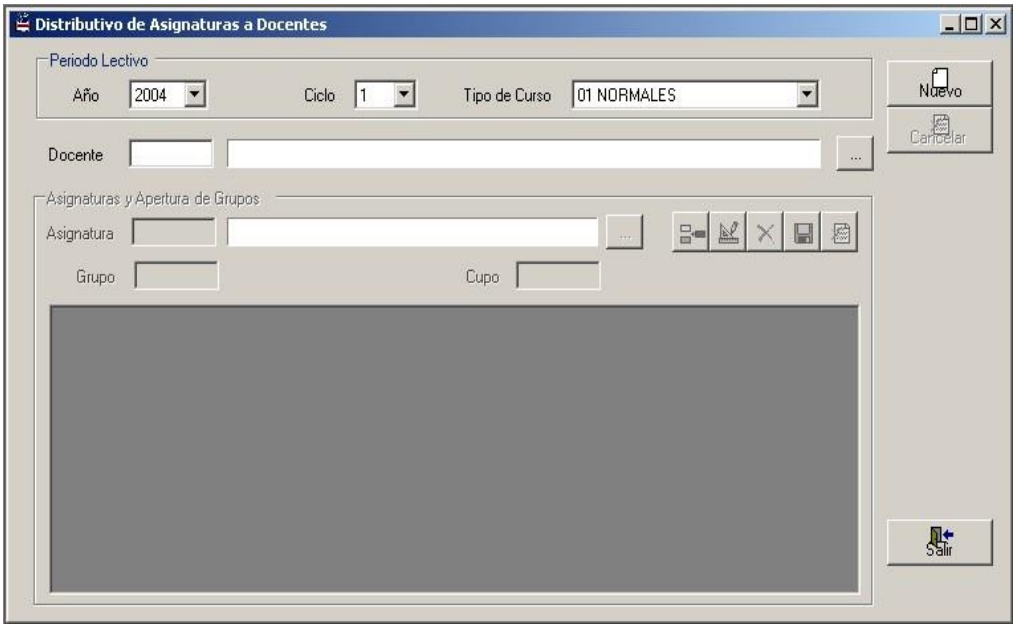


Figura 6.2. Período Lectivo- Paso 1

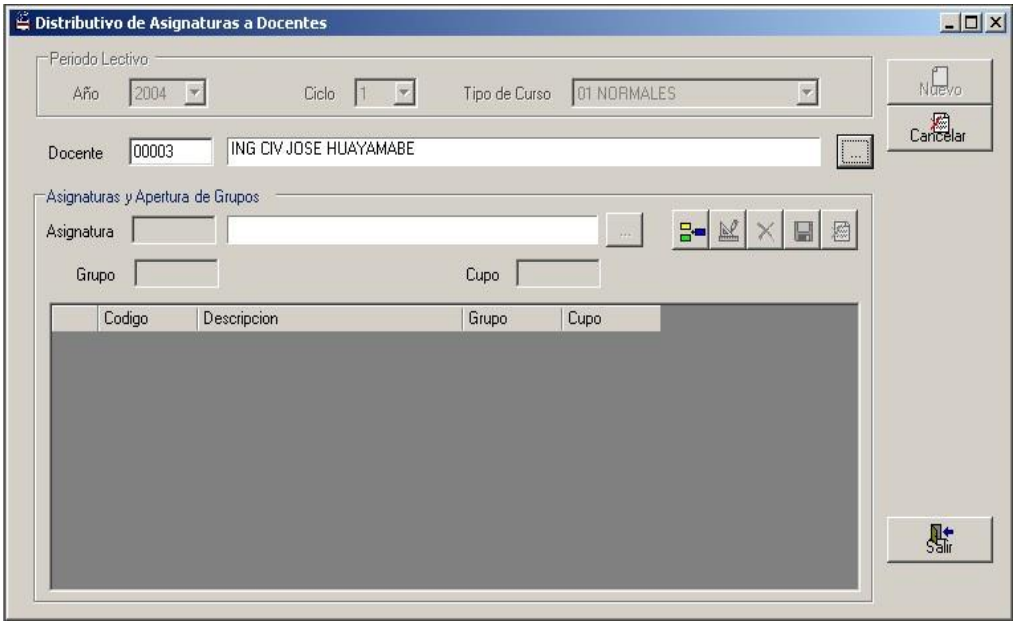


Figura 6.3. Docente- Paso 2

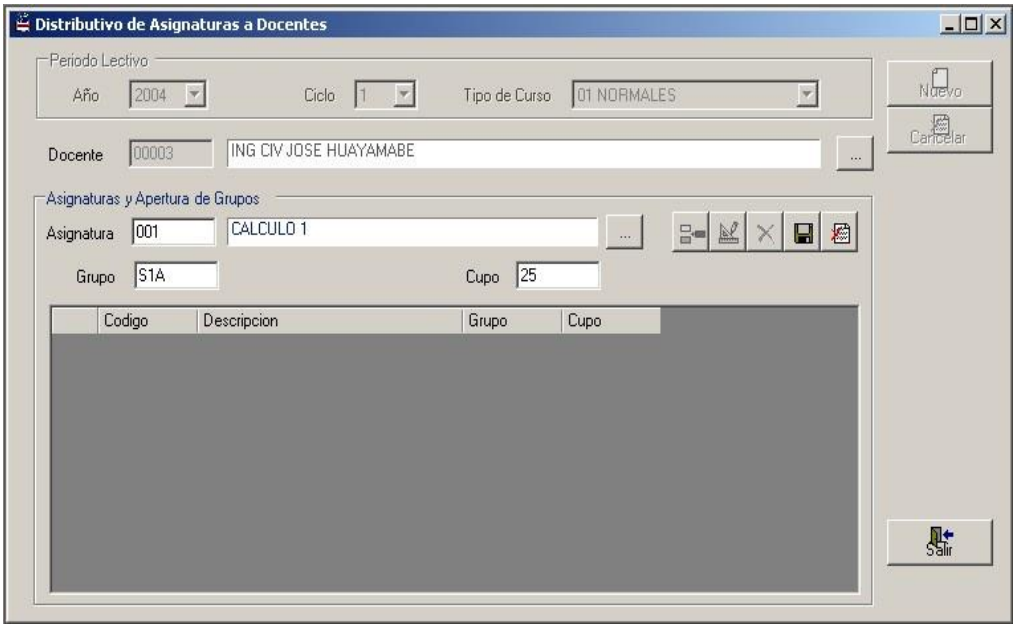


Figura 6.4. Asignaturas y Apertura de Grupos- Paso 3

A continuación vemos como resulta la pantalla después de haber ingresado toda la carga horaria para el Ing. Huayamabe.

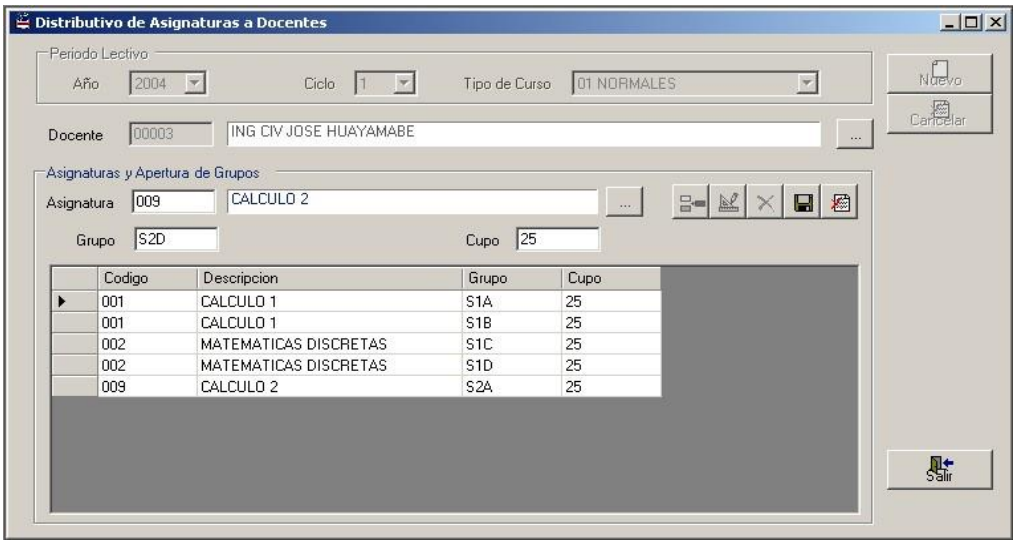


Figura 6.5. Distributivo de Asignaturas a Docentes

En caso de haber ingresado mal algún registro, damos un clic en la fila correspondiente en el Grid, y a continuación en el botón “Modificar Asignatura” . Automáticamente la información de esa fila pasará a las cajas de textos para poder ser modificadas. Cuando el usuario ha actualizado los datos se presiona el botón “Guardar Asignatura”. Cuando se desea eliminar un registro, se señala la fila que contiene esa información y se da clic en el botón “Eliminar Asignatura” .

Por ejemplo, deseamos eliminar la asignatura Matemáticas Discretas correspondiente al grupo S1C y tenemos:

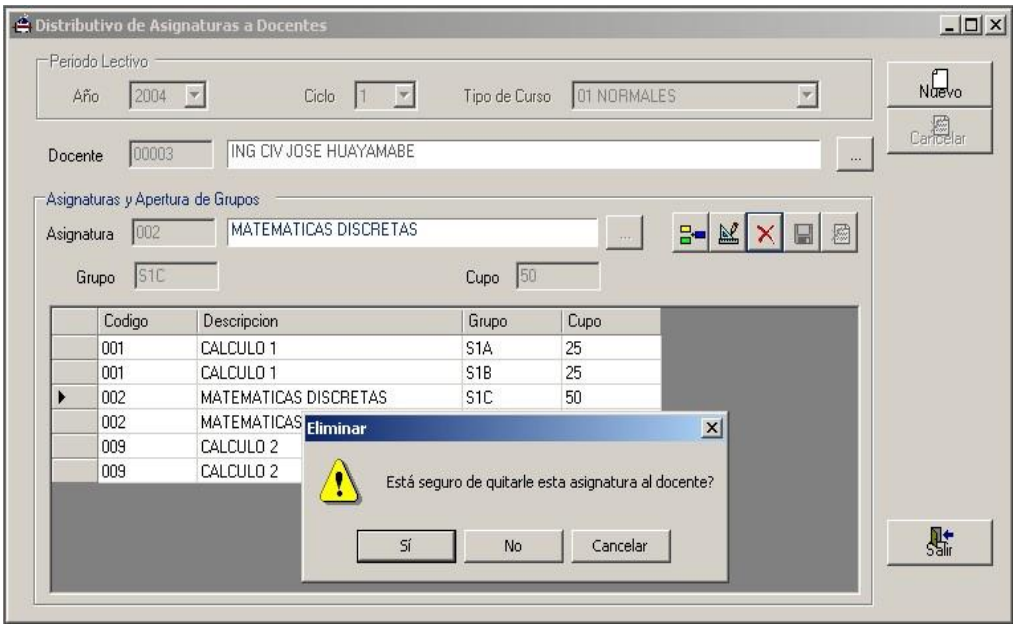


Figura 6.6. Ejemplo de Eliminación un Registro

6.2. Horarios por Grupo

Entrando en dicha pantalla realizamos lo siguiente:

- Llenamos los combos: año, ciclo y tipo de curso en lo que respecta al “Período Lectivo”. Cabe destacar que en esta pantalla se considerarán sólo los periodos lectivos con estado “Inicio de Matriculación”.
- Presionamos el botón para llenar en el combo “Grupo” todos los cursos que se han abierto en el período lectivo presente o que se ha sido asignado algún docente para cierta asignatura.
- Presionamos el botón “Nuevo” para crear el horario del grupo y sección seleccionados.

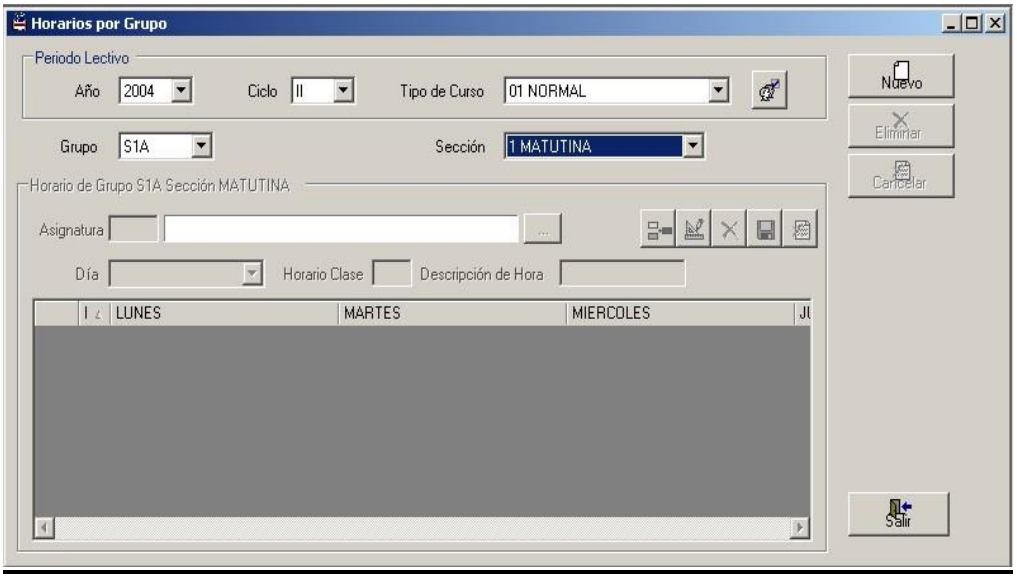


Figura 6.7. Horarios por Grupo-Paso1

- Presionamos el botón para crear cada hora-clase, seleccionando la asignatura para dicho grupo, ya sea digitando su código o presionando el botón derecho.
- Llenar el combo día, y las dos cajas de textos: horario de clase y descripción de la hora.

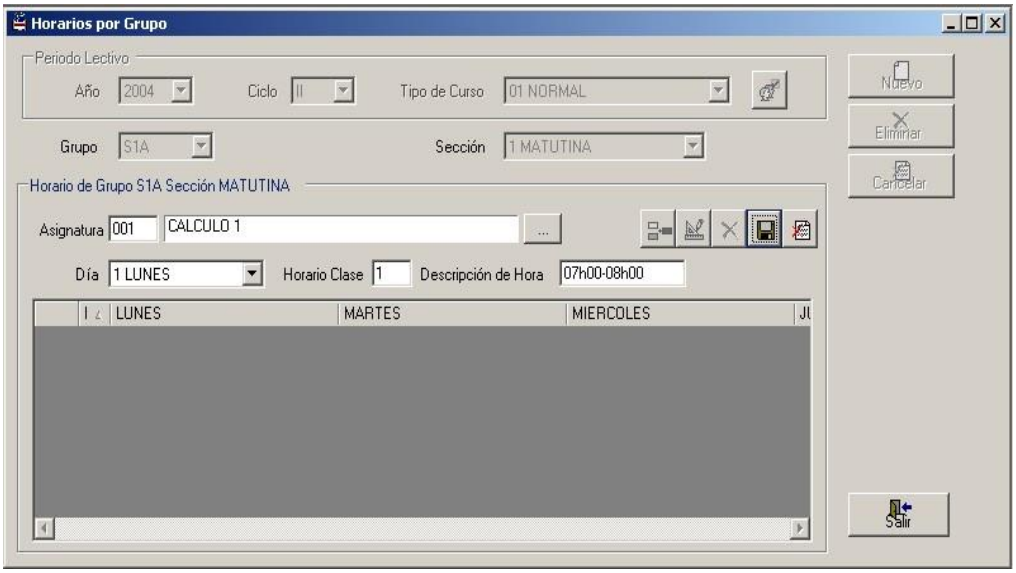


Figura 6.8. Horarios por Grupo-Paso2

- Finalmente se da clic en el botón para guardar la hora-clase ingresada.

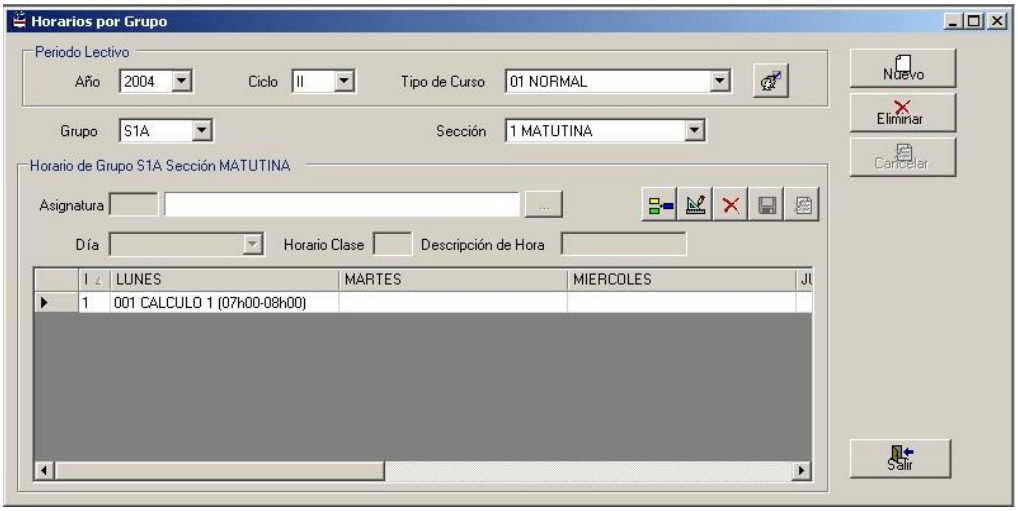


Figura 6.9. Horarios por Grupo-Paso 3

Manera de modificar un registro

En primer lugar se ha creado un horario de clases para un solo grupo y sección, como se ve a continuación:

GRUPO: S1A **SECCION: MATUTINO**

Hora Clase	Lunes	Martes	Miércoles	Jueves	Viernes
1	Cálculo 1	Prog. 1	Contab.	Prog. 1	Cálculo 1
2	Cálculo 1	Prog. 1	Contab.	Prog. 1	Cálculo 1
3	Español	Enfoq.Sist.	Español	Enfoq. Sist.	Español
4	Mat.Discr.	Informática	Inglés	Informática	Mat.Discr.
5	Mat.Discr.	Informática	Inglés	Informática	Mat.Discr.

HORARIO DE CLASES

Supóngase que se comete un error al ingresar los datos en el registro, como en este ejemplo se observa que se ingresó una asignatura que no corresponde a la hora-clase, es decir, para la materia de Inglés no corresponde a la 3era hora del día Miércoles como se muestra en el horario de clases.

Para corregir damos un clic en el registro en donde se ha ingresado mal la información y a continuación presionamos el botón “Modificar” .

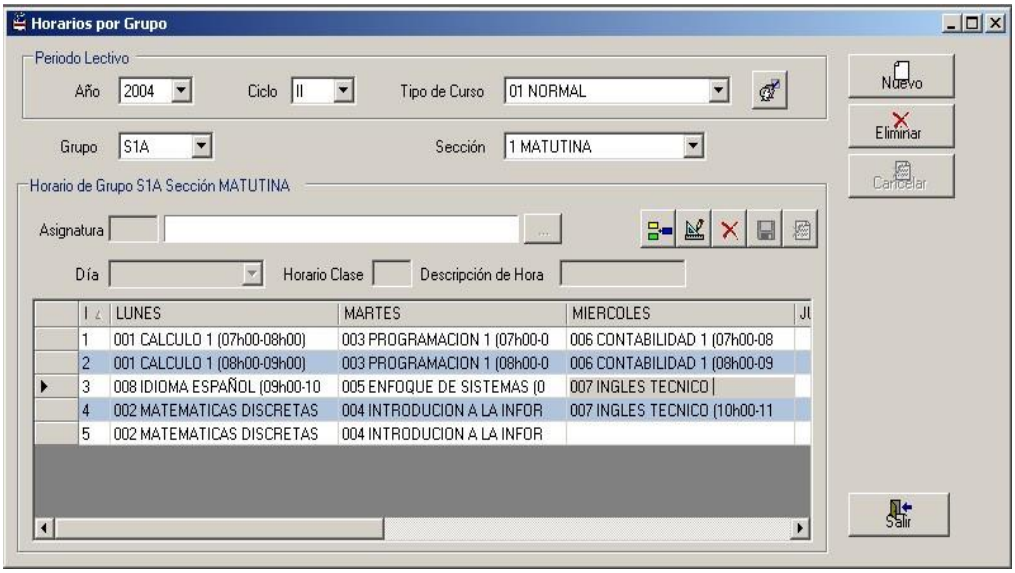


Figura 6.10. Modificar Hora-Clase (Paso 1)

Al presionar dicho botón se activará nuevamente la caja de texto “Asignatura”, como se muestra en la pantalla, para que sea modificado la respectiva hora-clase de dicha materia.

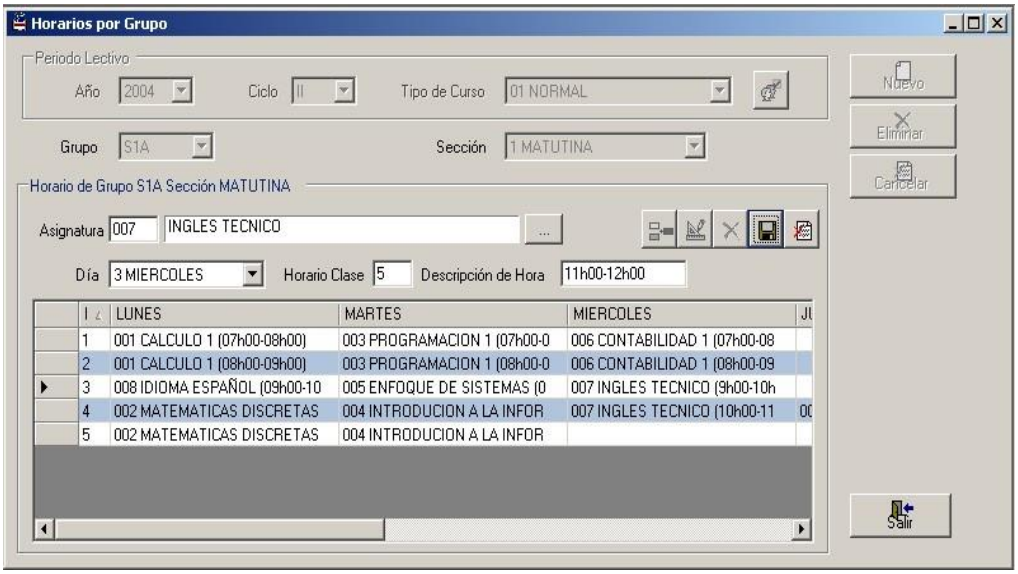


Figura 6.11. Modificar Hora-Clase (Paso2)

O bien para eliminar, así mismo algún registro dentro del horario por grupo, presionamos el botón “Eliminar” , como se muestra en la pantalla:

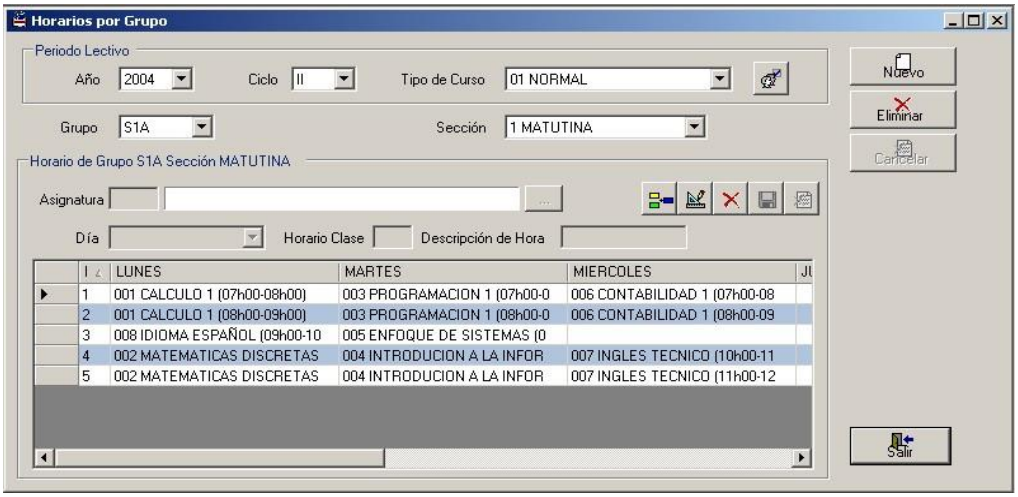


Figura 6.12. Corrección de Hora-Clase (Paso 3)

A continuación vemos como resulta la pantalla después de haber ingresado todos los horarios por grupo adecuadamente:

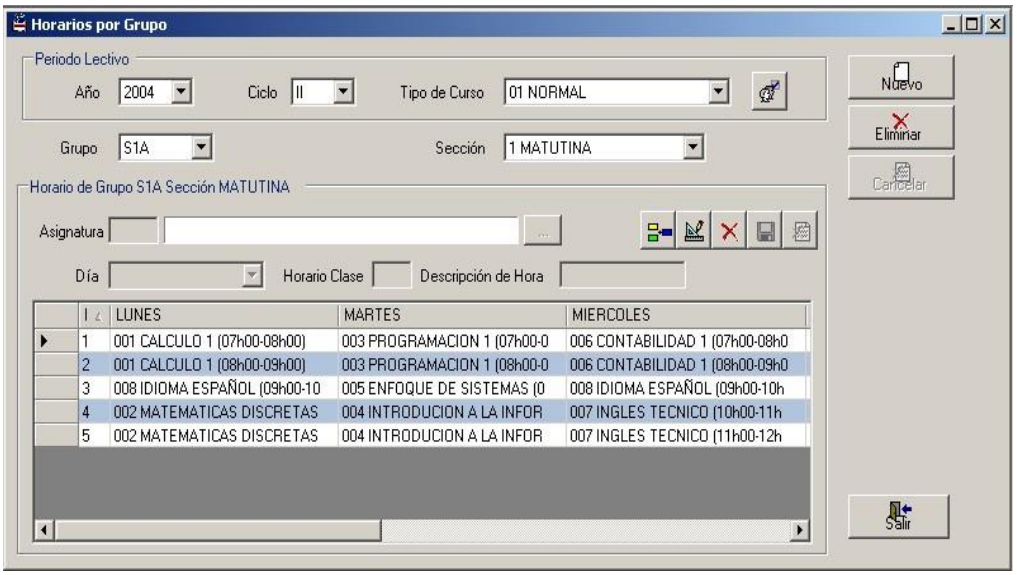


Figura 6.13. Ejemplo de Horarios por Grupo

CAPITULO 7

7. INSCRIPCIÓN Y MATRICULACIÓN

7.1. Inscripción de Estudiante a Subunidad

Antes que un estudiante vaya a matricularse, primeramente debe estar registrado en la unidad académica.

Figura 7.1. Acceso a Inscripción de Estudiante a Subunidad

Para llenar el respectivo registro de Estudiantes en la Unidad Académica se deberá realizar los siguientes pasos:

- Presionar el botón “Nuevo” para seleccionar el pénsum que es precisamente con el que va a ingresar el estudiante a la subunidad.

- Digitar el código del estudiante o seleccionándolo a través de la consulta presionando el botón derecho.
- Al seleccionar la Condición Académica se tiene lo siguiente:

  Activo: Condición que se asigna mientras el estudiante realiza sus estudios en la subunidad.

  Inactivo: Condición que se le asigna cuando el estudiante se ha retirado de la subunidad.

  Graduado: Cuando el estudiante ha terminado sus estudios y ha realizado tesis o seminario de graduación.

  Egresado: Cuando al estudiante le falta sólo de realizar seminario o tesis de grado para culminar su carrera.

  Por lo general se deberá seleccionar en la Condición Académica: “Activo”.
- Se selecciona el último nivel aprobado, por defecto deberá ser 0. Este último nivel aprobado se refiere a que cada nivel en un pénsum tiene un determinado número de materias y cuando el estudiante aprueba todas las asignaturas de un nivel, esto quiere decir, que ha aprobado dicho nivel.
- Digitar el Número de Cruces Permitidos, que se refiere en ingresar cuántas materias se pueden cruzar en el horario y en el grupo escogido por el estudiante a la hora de matricularse.

- Se digita la fecha de ingreso del estudiante a la unidad académica.
- Se ingresa alguna observación si es necesario.
- Las demás cajas de textos aparecerán deshabilitados, ya que éstos servirán solamente de información cuando el estudiante haya culminado su carrera en dicha Subunidad Académica.

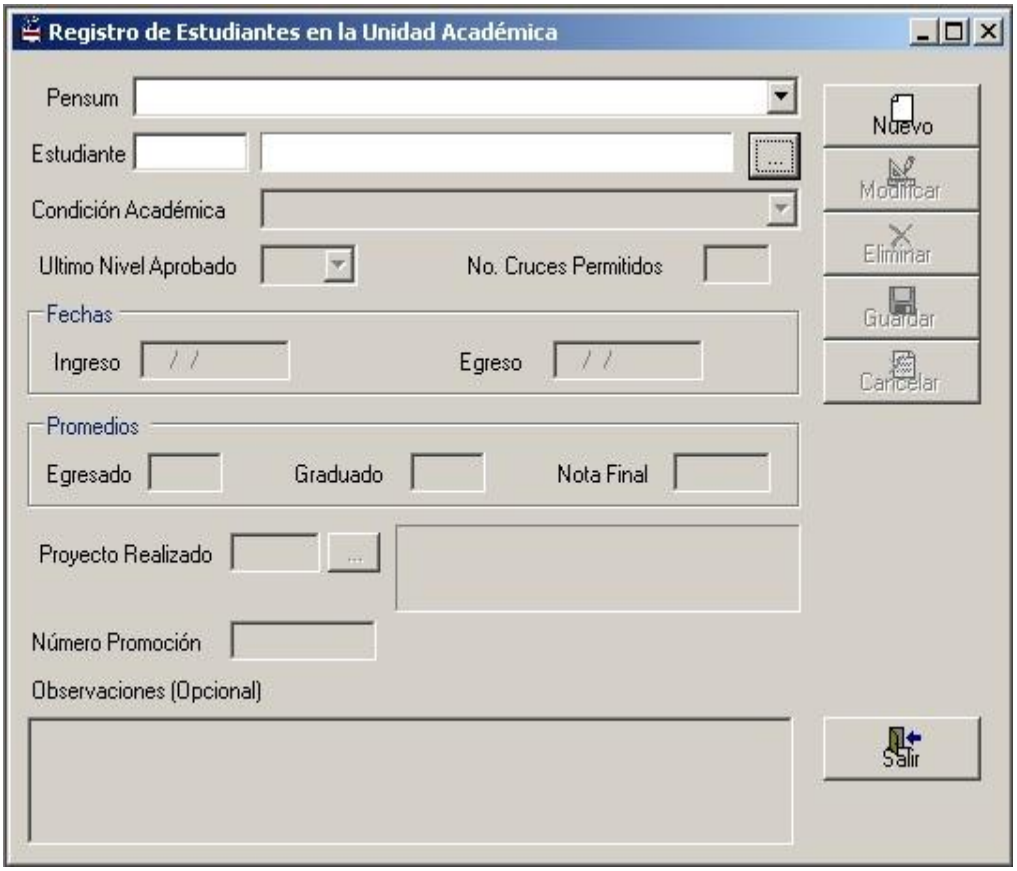


Figura 7.2. Registro de Estudiantes en la Unidad Académica

Llenando el registro con la información adecuada se tiene:

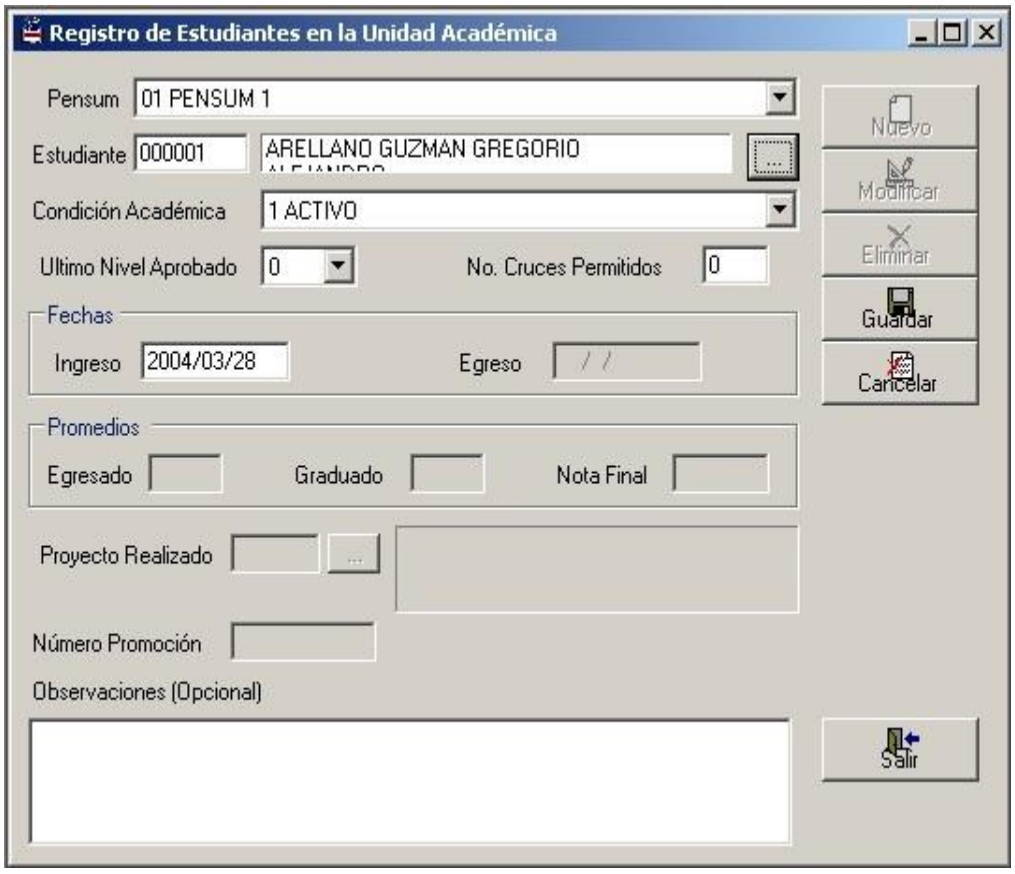


Figura 7.3. Ejemplo del Registro de Estudiantes en la Unidad Académica

Para modificar o eliminar el registro de un estudiante en una subunidad académica sólo basta con digitar el código del alumno o seleccionándolo a través de la consulta presionando el botón derecho.

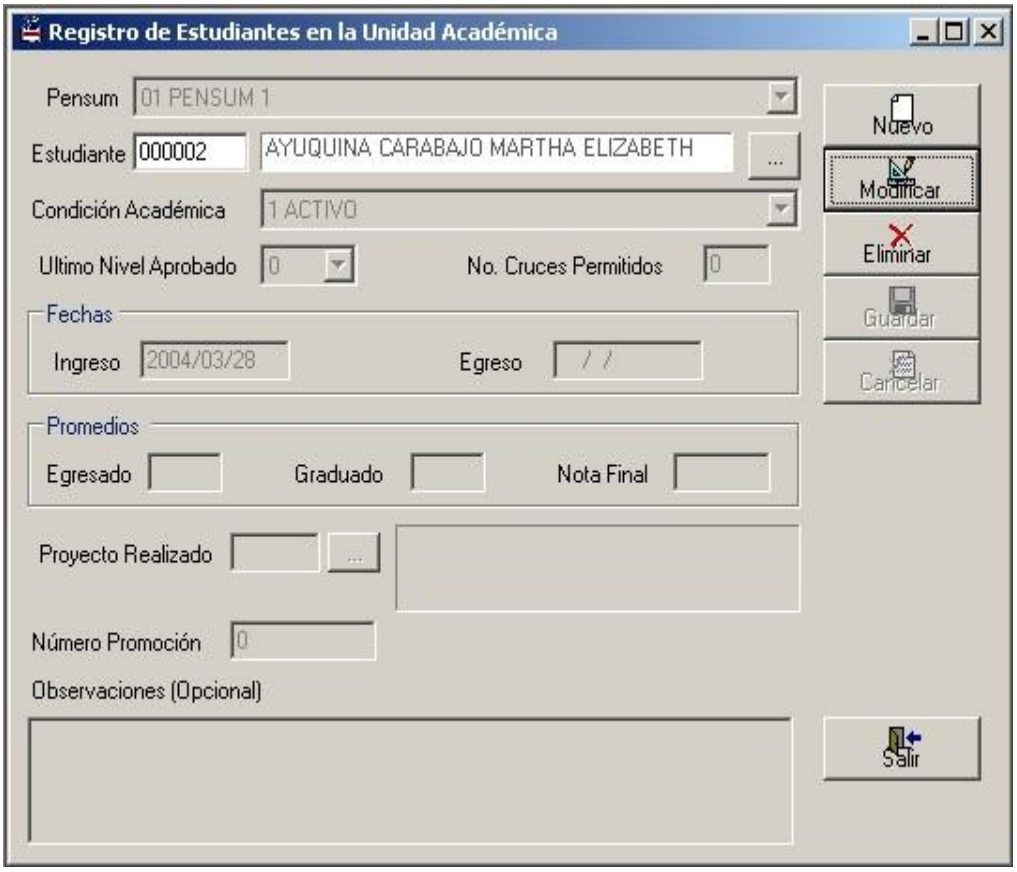


Figura 7.4. Consultando Registro de Estudiante en una Subunidad

Si presionamos el botón Eliminar, dicho registro será borrado. En caso de que presionemos el botón Modificar, se habilitarán ciertas cajas de textos que podrán ser modificados manualmente.

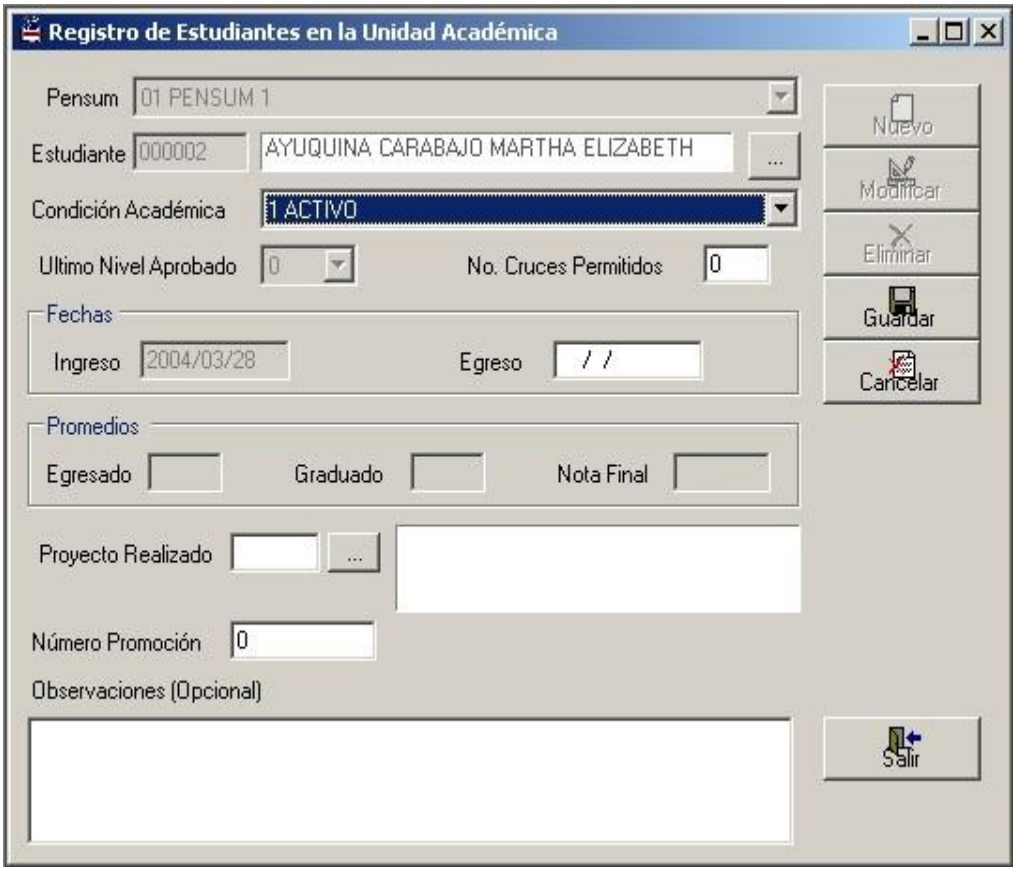


Figura 7.5. Modificando Registro de Estudiante en una Subunidad Académica

Los campos que no se pueden modificar en esta pantalla son los siguientes:

- **Pénsum:** Existe una pantalla especial para indicar las razones por la que se ha cambiado un estudiante de malla curricular.
- **Ultimo nivel aprobado:** Este campo se incrementa automáticamente al término de un período lectivo cuando el estudiante ha aprobado todas las asignaturas de un nivel.

- **Fecha de ingreso:** Sólo se le ingresa al momento que el estudiante se registra a la subunidad.
- **Promedio de egresado, promedio de graduado y promedio final:** Se generan cuando el estudiante finaliza su vida académica en la subunidad.

7.2. Inscripción de Estudiante a una Asignatura

Para llenar dicha pantalla se deberá efectuar los siguientes pasos:

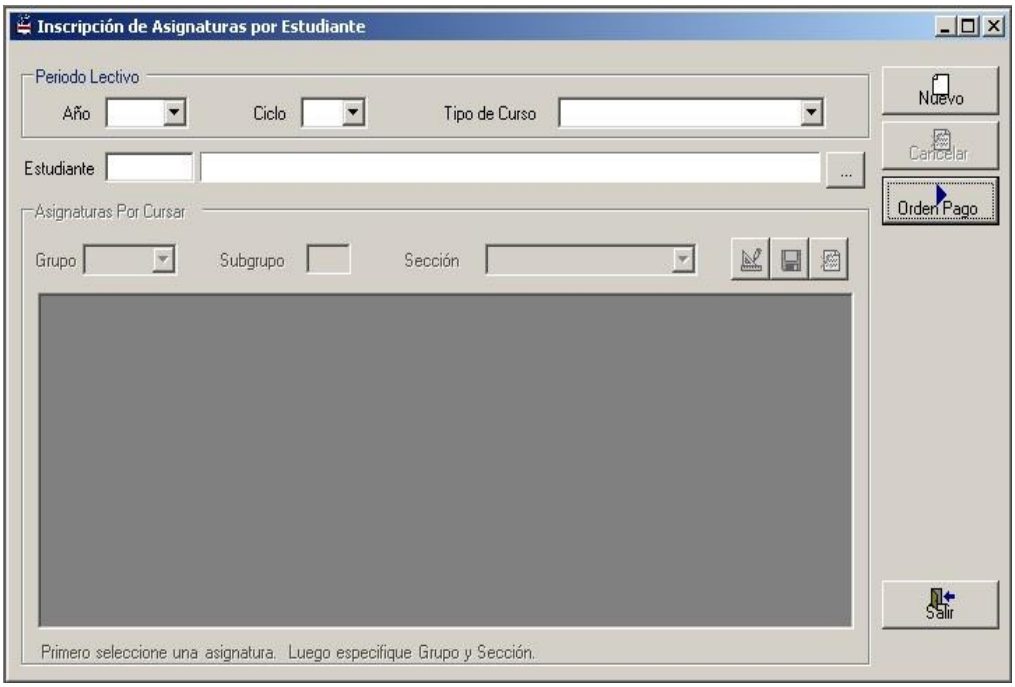


Figura 7.6. Inscripción de Asignaturas por Estudiante

- Seleccionar un "Período Lectivo" que se abrirá para la matriculación del estudiante. Cabe destacar que en esta

pantalla se considerarán sólo los periodos lectivos con estado “Inicio de Matriculación”.

- Hacer clic en ”Nuevo” para digitar el código del estudiante o seleccionándolo a través de la consulta presionando el botón derecho, automáticamente aparecerá todas las asignaturas que podría ver el alumno según la malla curricular que le haya sido asignada.
- Para poder inscribir al estudiante para cualquier materia se hace un clic en el botón “Modificar” . Cuando se selecciona una materia, si ésta se la ve en varios grupos, se habilitará los combos “Grupo” y “Sección”.
- Para elegir la asignatura en el grupo asignado se deberá hacer un clic en la columna “Cursar”.

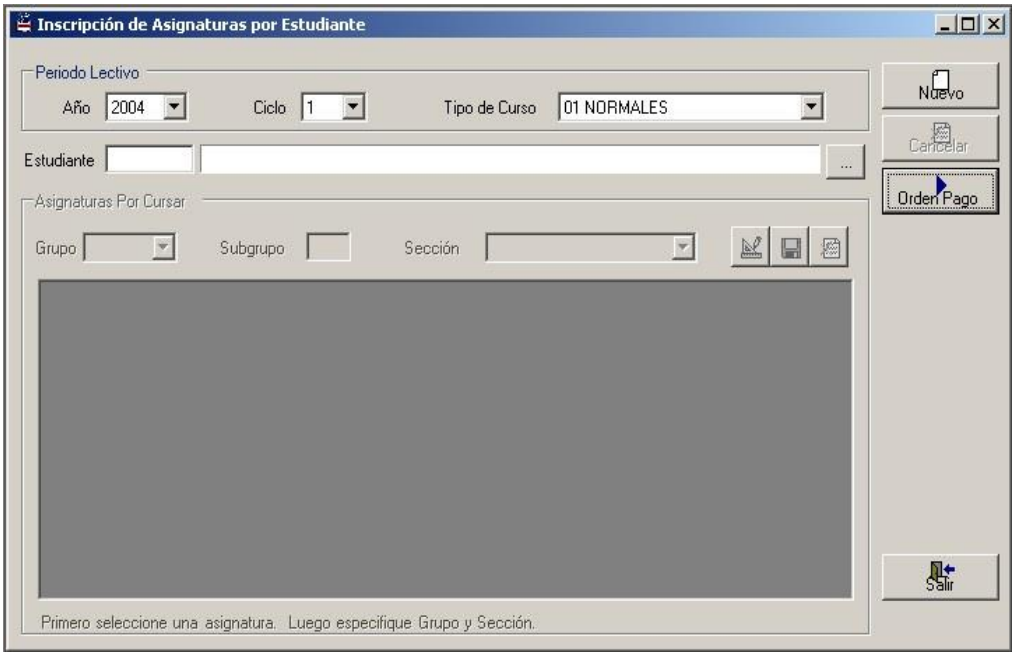


Figura 7.7. Inscripción de Asignaturas por Estudiante-Paso 1

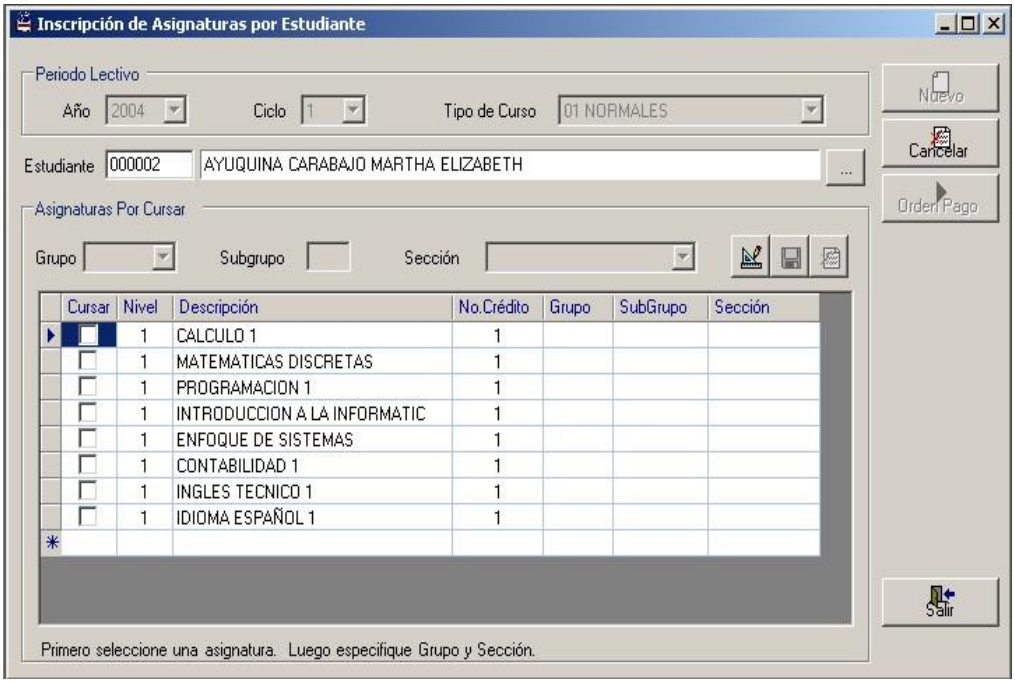


Figura 7.8. Inscripción de Asignaturas por Estudiante- Paso 2

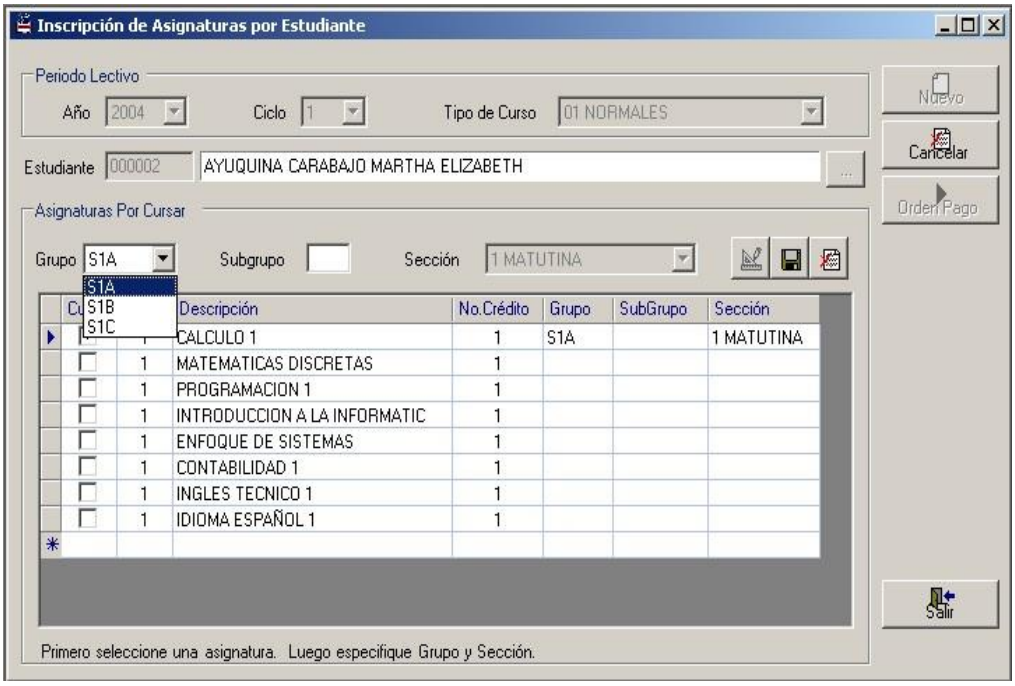


Figura 7.9. Inscripción de Asignaturas por Estudiante- Paso 3

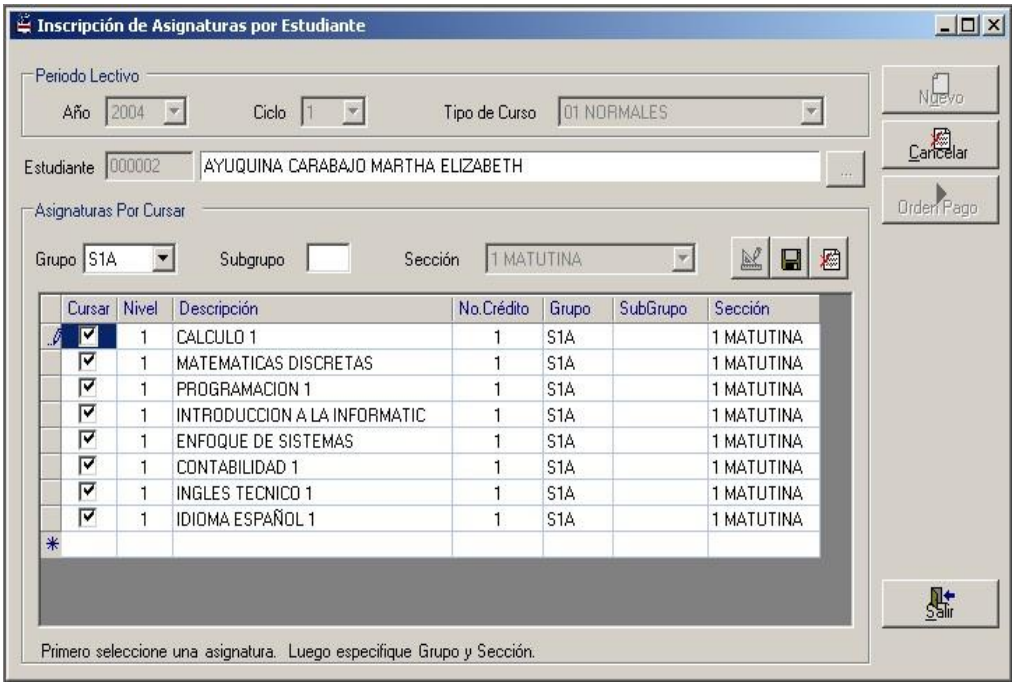


Figura 7.10. Inscripción de Asignaturas por Estudiante- Paso 4

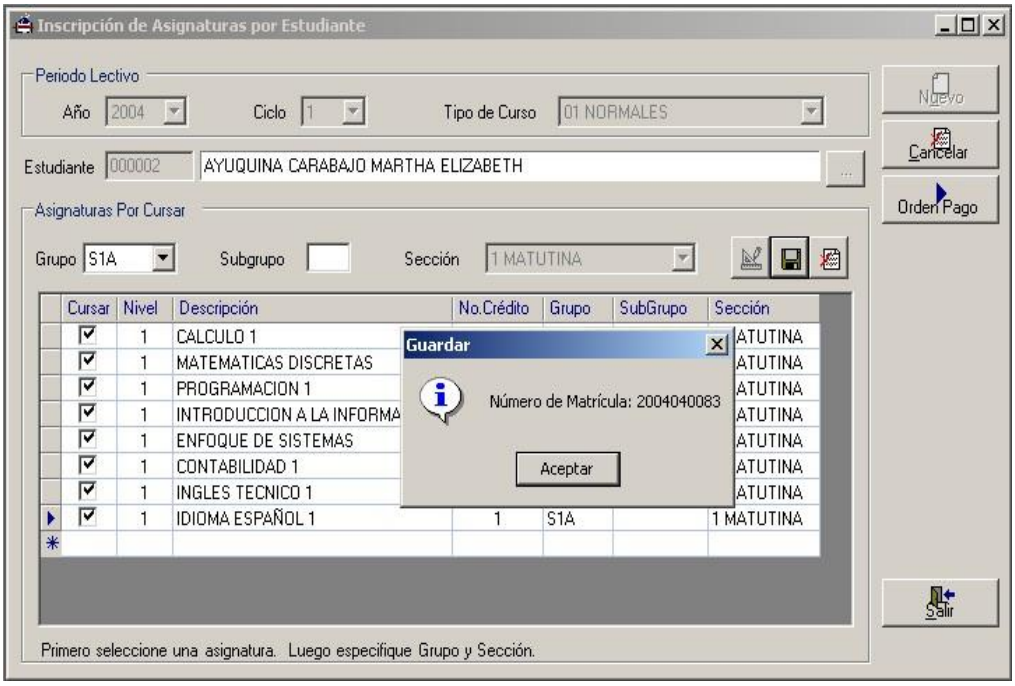


Figura 7.11. Inscripción de Asignaturas por Estudiante- Paso 5

Otros puntos importantes:

- Para seleccionar una asignatura, ésta deberá estar dentro de un horario, caso contrario será imposible tomarla.
- Cuando una asignatura pertenece solamente a un grupo, los combos “Grupo” y “Sección” quedarán deshabilitados.
- La caja de texto “Subgrupo” es cuando un grupo se la está dividiendo en varios paralelos.
- El botón “Orden de Pago” activa la pantalla que veremos a continuación.

7.3. Orden de Pago

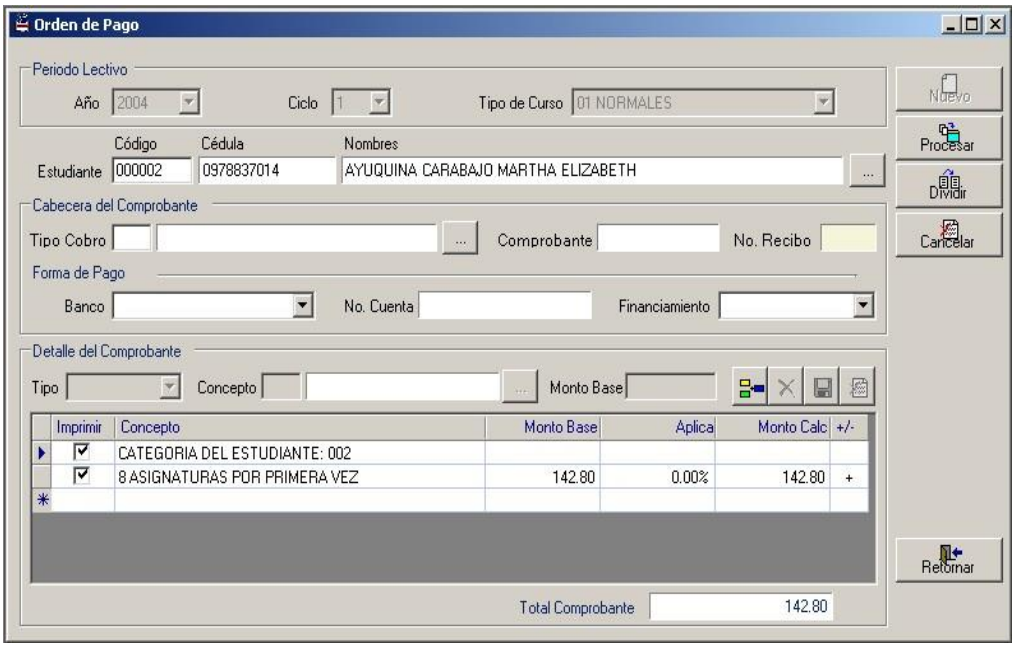


Figura 7.12. Orden de Pago

En lo que se refiere a:

- "Cabecera de Comprobante" se tiene que seleccionar el Tipo o concepto de Cobro. El Número de Comprobante y el Número de Recibo aparecerán después de dar clic en el botón "Procesar".
- "Forma de Pago" se debe seleccionar el Banco y el tipo de Financiamiento para dicho pago, así mismo aparecerá el Número de Cuenta.

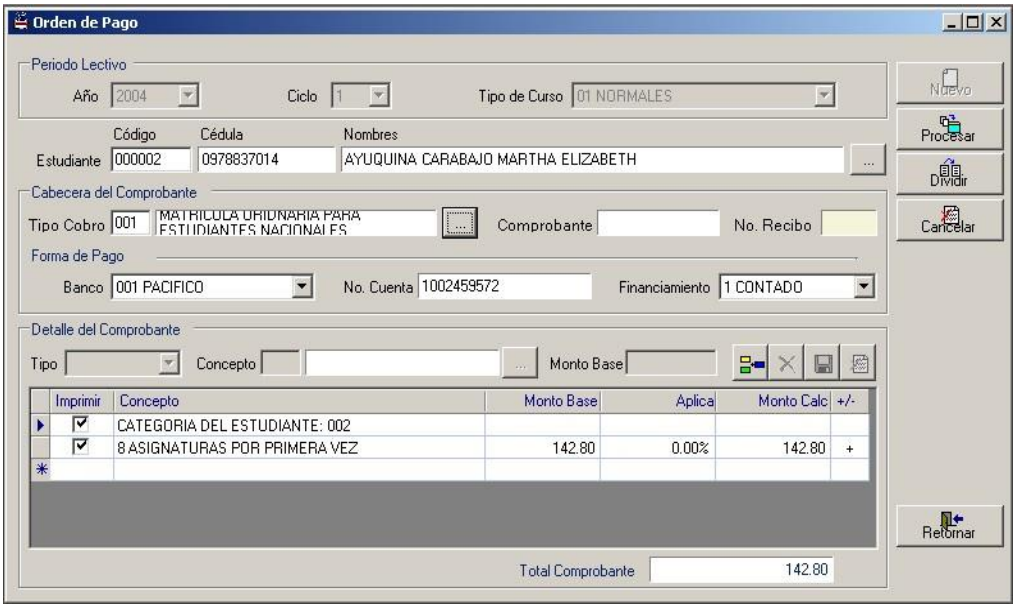


Figura 7.13. Orden de Pago- Paso 1

Damos clic en el botón “Procesar” para dejar registrada en la base de datos la orden de pago referente a la matrícula de ese estudiante.

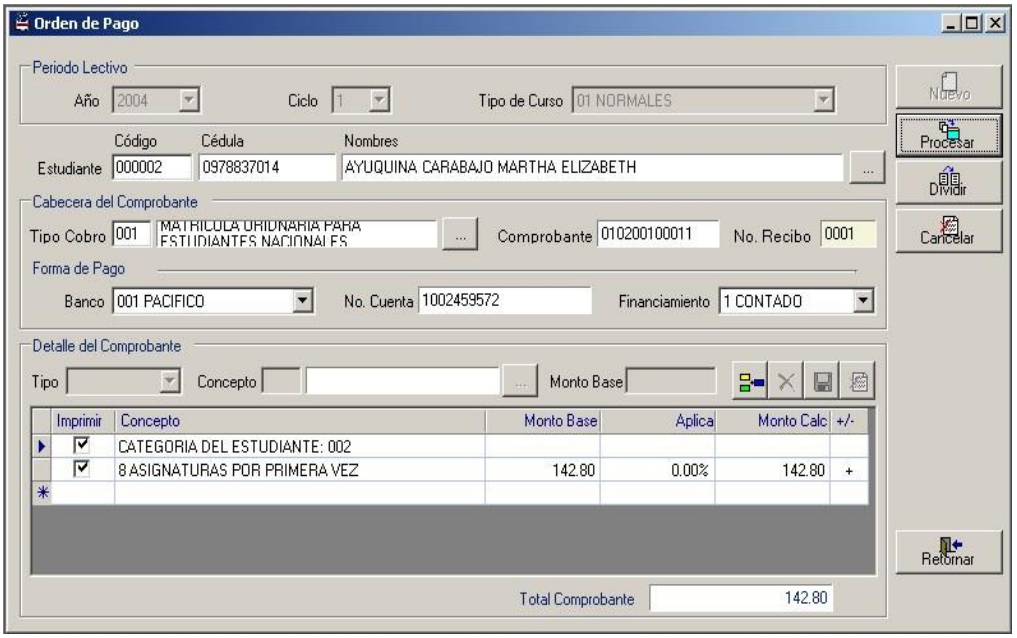


Figura 7.14. Orden de Pago- Paso 2

- Si uno desea ingresar nuevos valores ya sea de tipo aranceles o descuentos se deberá presionar el botón “Nuevo” .

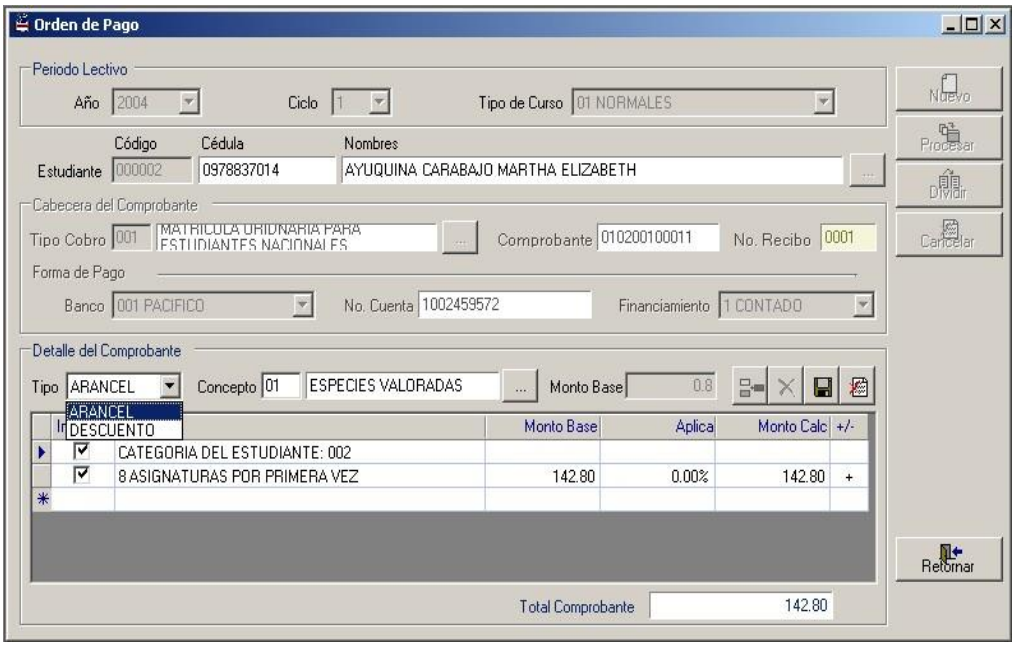


Figura 7.15. Orden de Pago- Paso 3

- Al presionar el botón “Guardar” aparecerá en el grid de consulta los nuevos valores ingresados.

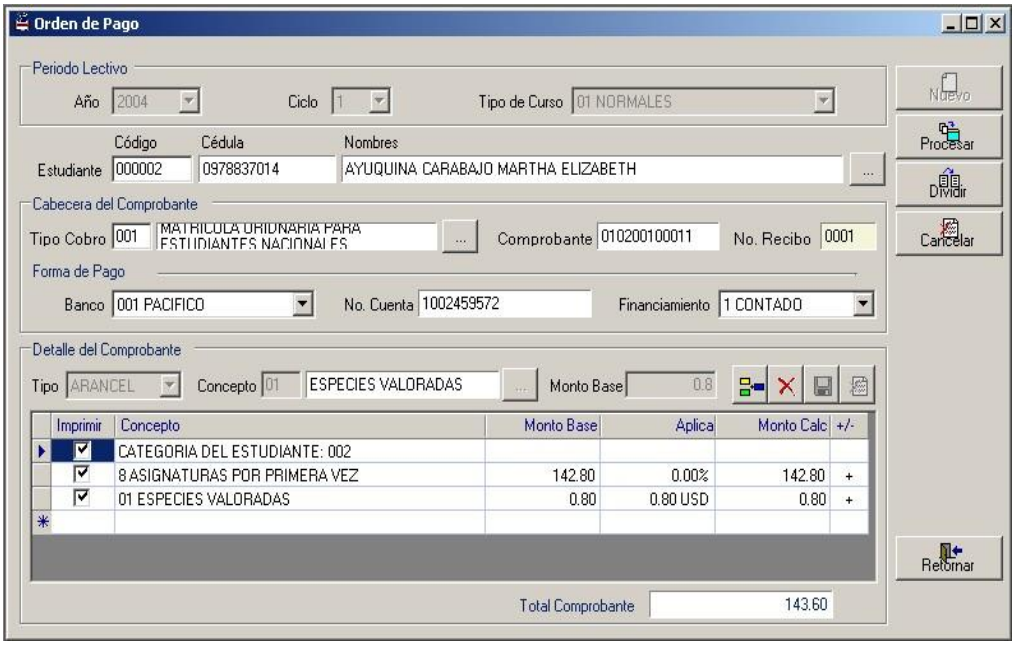


Figura 7.16. Orden de Pago- Paso 4

Supongamos que a este estudiante se le haga un descuento por sus calificaciones del año anterior, se lo ingresará de la misma forma del arancel anterior.

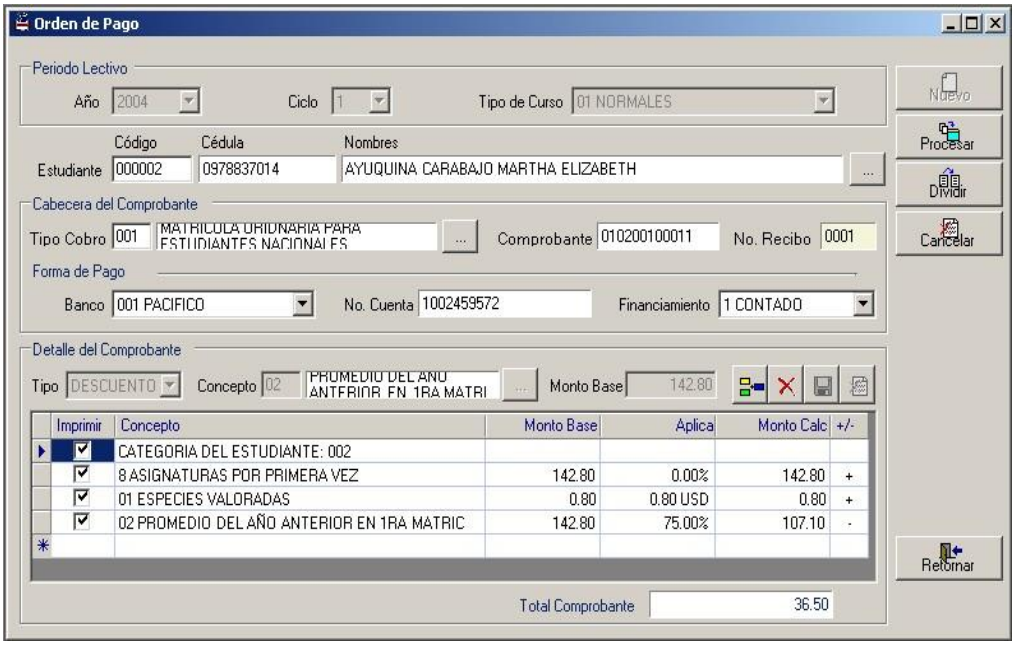


Figura 7.17. Orden de Pago- Paso 5

Cuando uno termina de ingresar todos los aranceles y descuentos para esa matrícula se debe presionar el botón “Procesar”. Para poder imprimir esta orden de pago en una o más cuotas se presiona el botón “Dividir”.

7.4. Cuotas

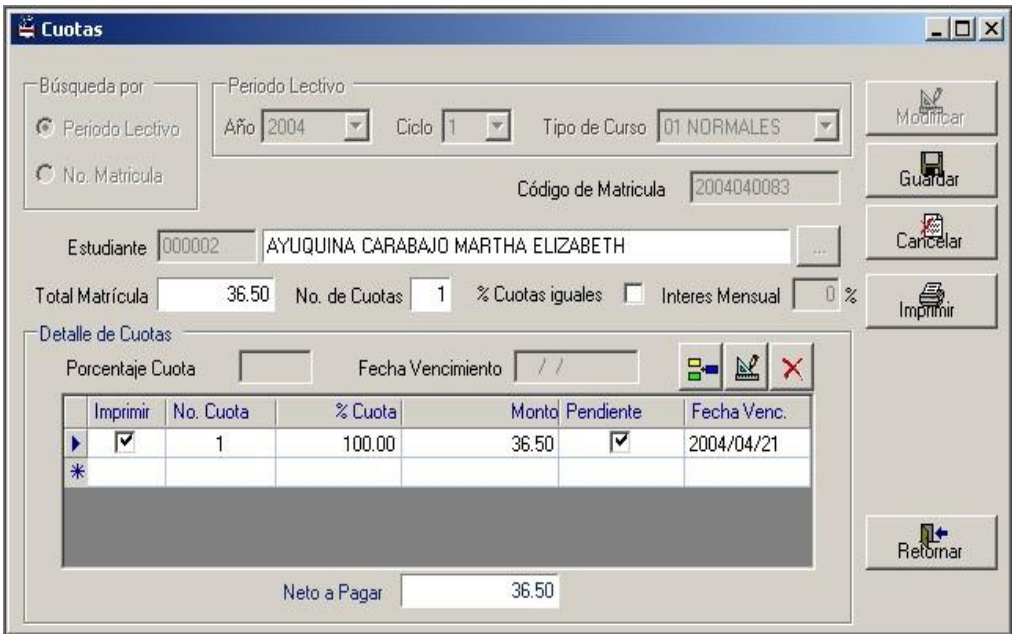


Figura 7.18. Cuotas- Paso 1

En el caso de que la orden de pago no se divida en cuotas se presionan los botones “Guardar” y después “Imprimir” directamente. De otra manera se pulsará el botón “Nueva Cuota” .

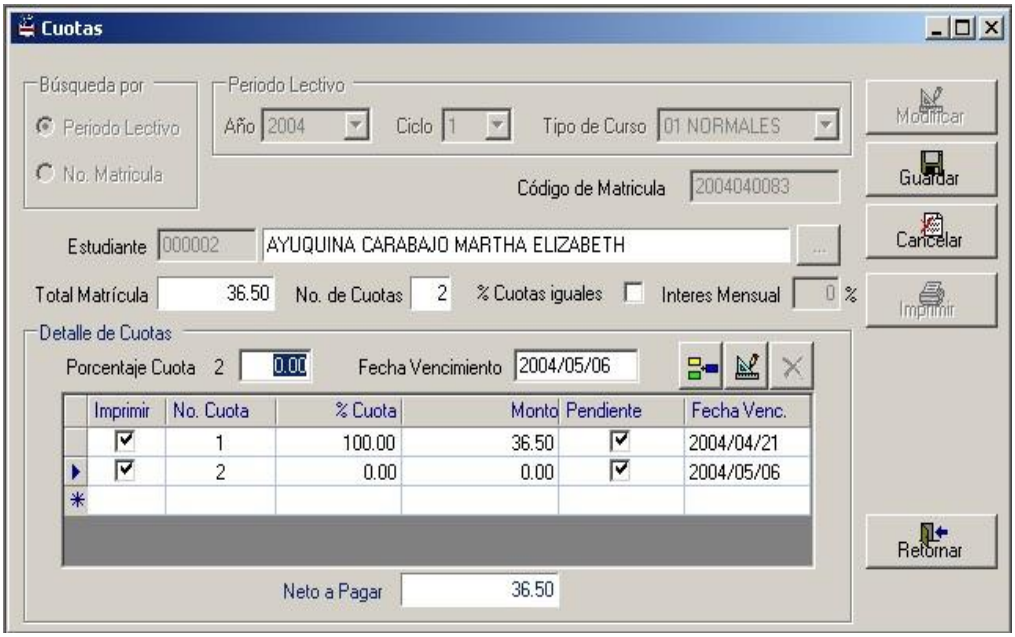


Figura 7.19. Cuotas- Paso 2

Una vez que se haya creado una nueva cuota se puede modificar los valores de porcentaje de cuota y la fecha de vencimiento de cada una, basta con ubicarse en cada fila y el sistema automáticamente detectará a qué cuota se refiere.

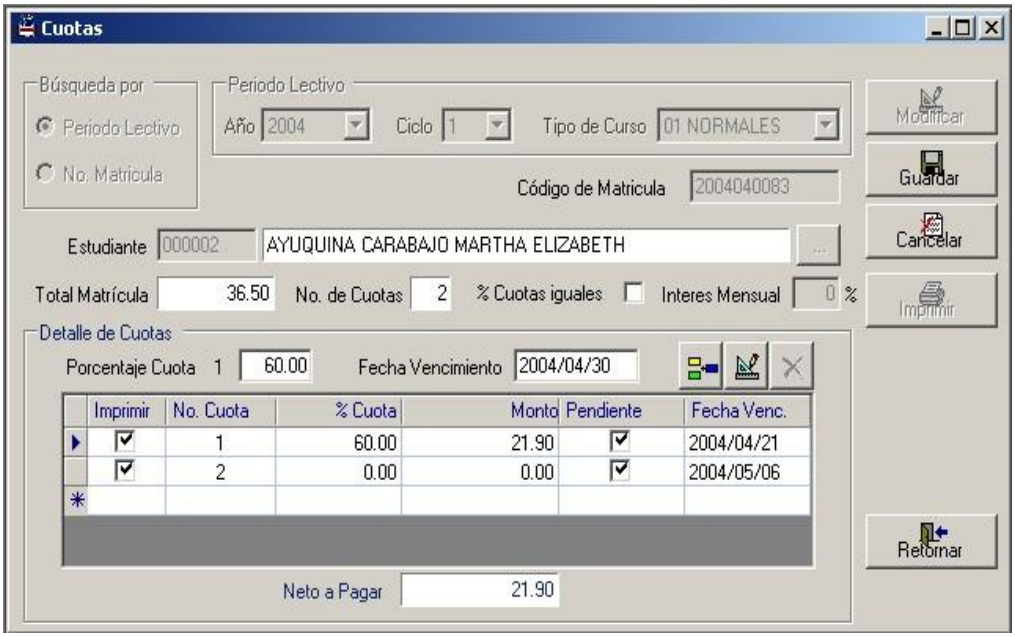


Figura 7.20. Cuotas- Paso 3

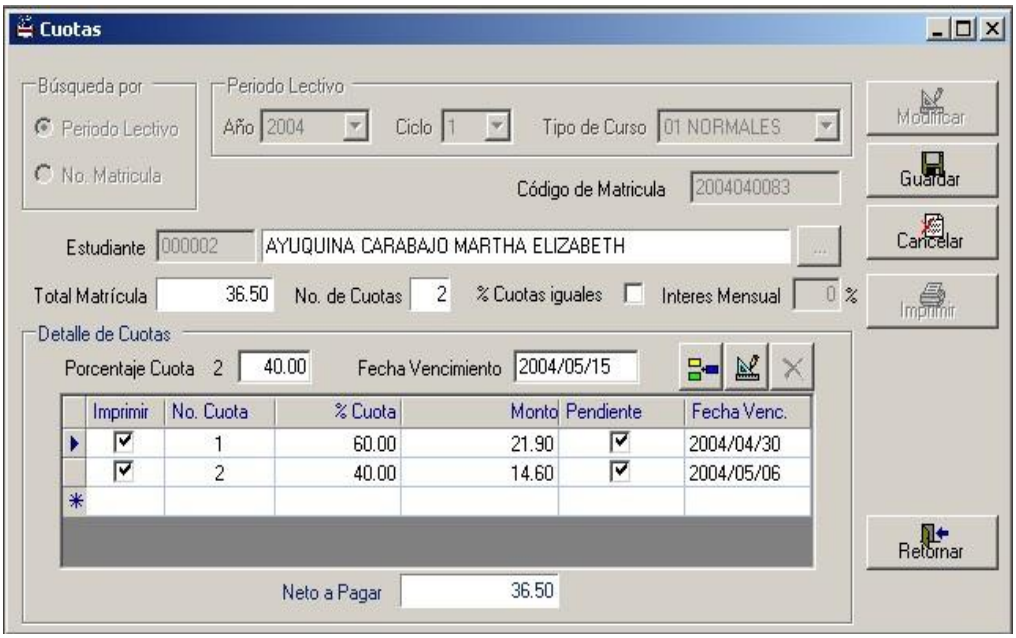


Figura 7.21. Cuotas- Paso 4

En el caso que el valor de la matrícula se divida en varias partes iguales con un porcentaje de interés mensual se da clic en la opción "% Cuotas Iguales" y se ingresa el Interés Mensual.

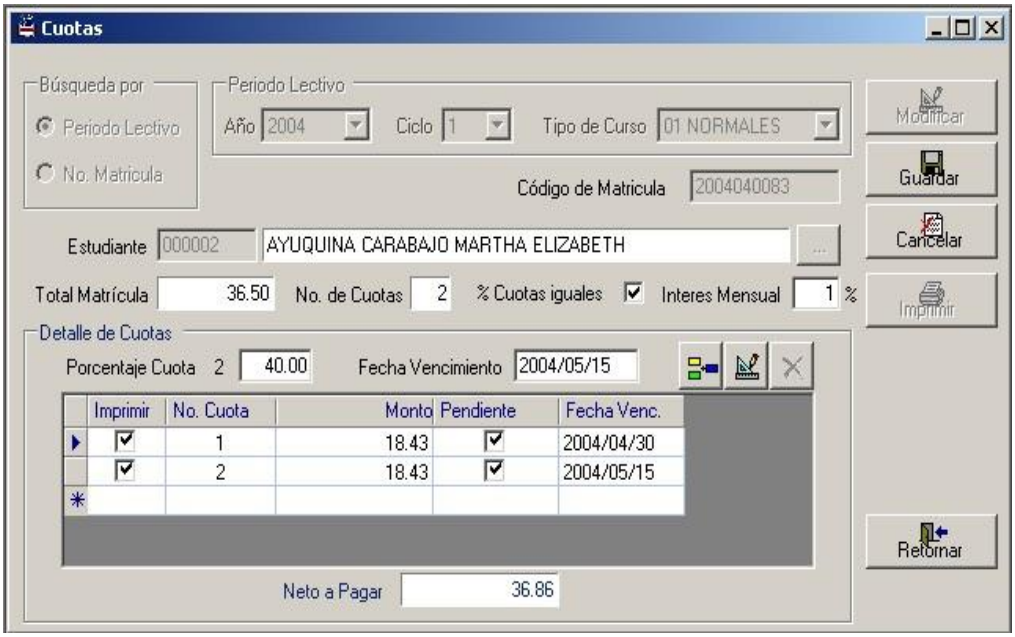


Figura 7.22. Cuotas- Paso 5

Una vez realizado esto, se presiona el botón "Guardar" y luego el botón "Imprimir" para la emisión de las cuotas. Se imprimirán sólo las cuotas que tengan visteadas la primera columna.

7.5. Cambio de Categoría

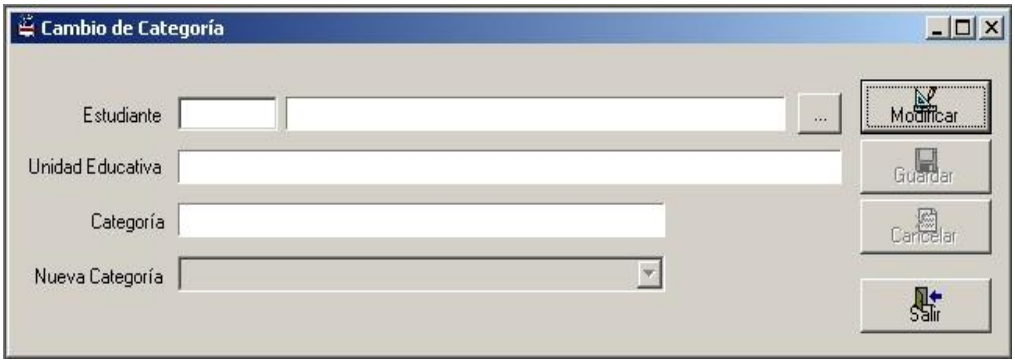


Figura 7.23. Cambio de Categoría

En dicha pantalla se debe realizar lo siguiente:

- En "Estudiante" digitar el código del alumno o seleccionándolo a través de la consulta presionando el botón derecho.
- Luego presionar el botón "Modificar" en donde se habilitarán las cajas de textos "Unidad Educativa" y "Categoría", en las cuales aparecerá tanto la Unidad Educativa o Colegio de donde proviene el estudiante y la categoría que pertenece dicha institución. Así mismo se habilitará el combo "Nueva Categoría" para realizar el cambio.

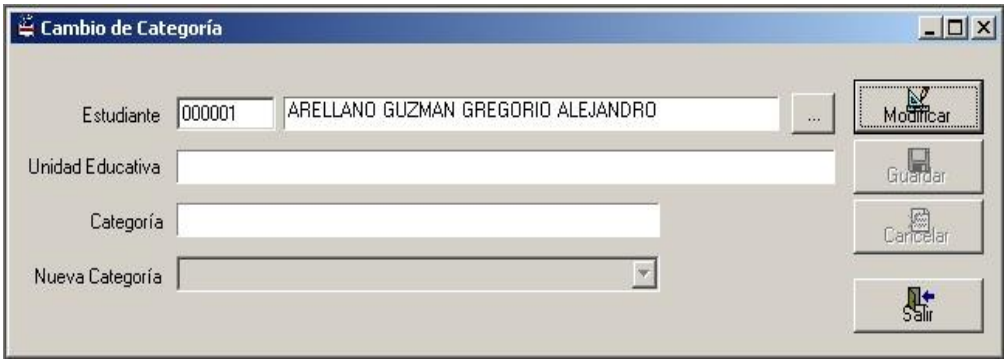


Figura 7.24. Ejemplo de Cambio de Categoría- Paso 1

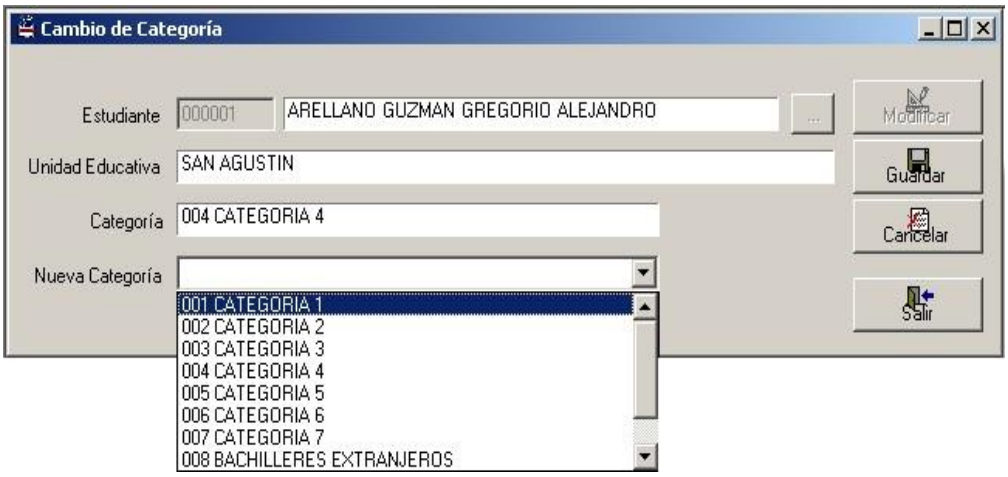


Figura 7.25. Ejemplo de Cambio de categoría- Paso 2

7.6. Cambio de Pénsum

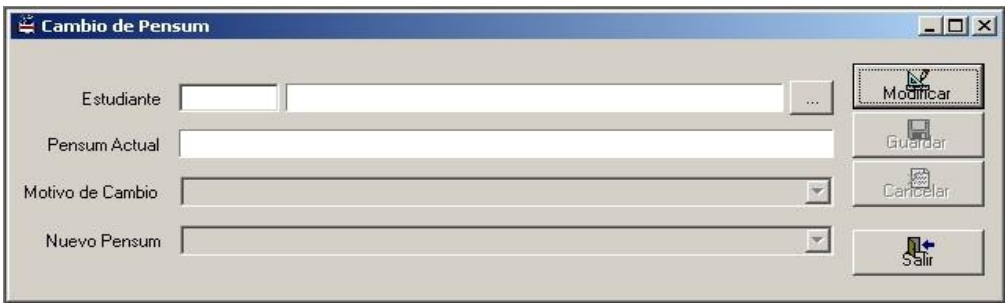


Figura 7.26. Cambio de Pénsum

En esta pantalla se debe de realizar los siguientes pasos:

- En "Estudiante" digitamos el código del alumno o seleccionándolo a través de la consulta presionando el botón derecho.
- Al presionar el botón "Modificar" se habilitará la caja de texto "Pénsum Actual" en donde aparecerá el pénsum. Al mismo tiempo se habilitarán los combos "Motivo de Cambio" y "Nuevo Pénsum".

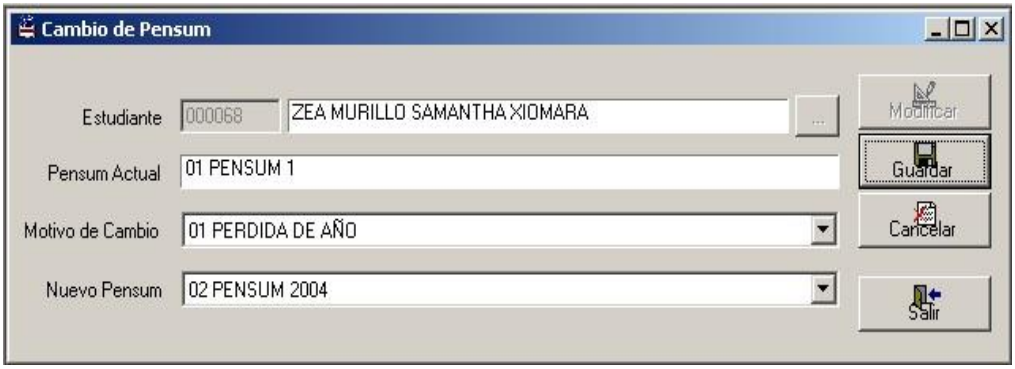


Figura 7.27. Ejemplo de Cambio de Pénsum

7.7. Registro de Órdenes Canceladas

Esta pantalla sirve para registrar las órdenes de pago de un estudiante que hayan sido canceladas.

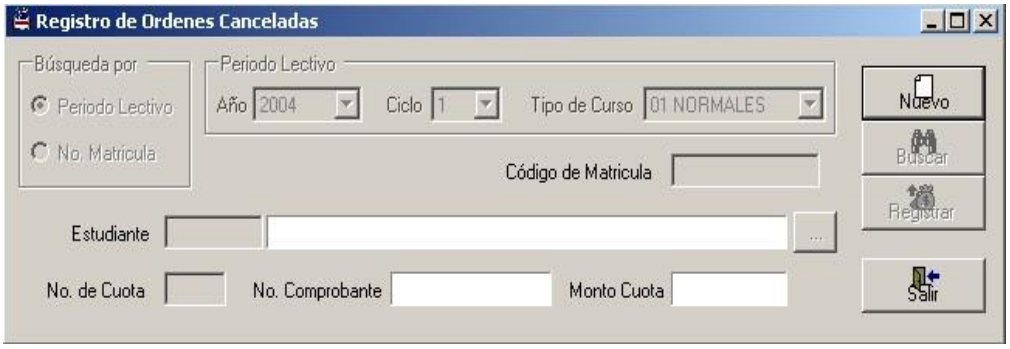


Figura 7.28. Registro de Órdenes Canceladas

Los pasos a seguir son:

- En “búsqueda por”, elegir si se desea por “Periodo Lectivo” o por “Número de Matrícula”
- Si se elige “Buscar por Periodo Lectivo” se selecciona el año, ciclo y tipo de curso del periodo actual, caso contrario se digita el número de matrícula generado al momento de inscribirse.
- Se ingresa el código del estudiante y el número de cuota y se presiona el botón “Buscar”. Automáticamente aparece el número del comprobante y el monto, y se habilita el botón “Registrar”, al dar clic, el sistema procesará a modificar el estado de la cuota, de pendiente a Cancelado.

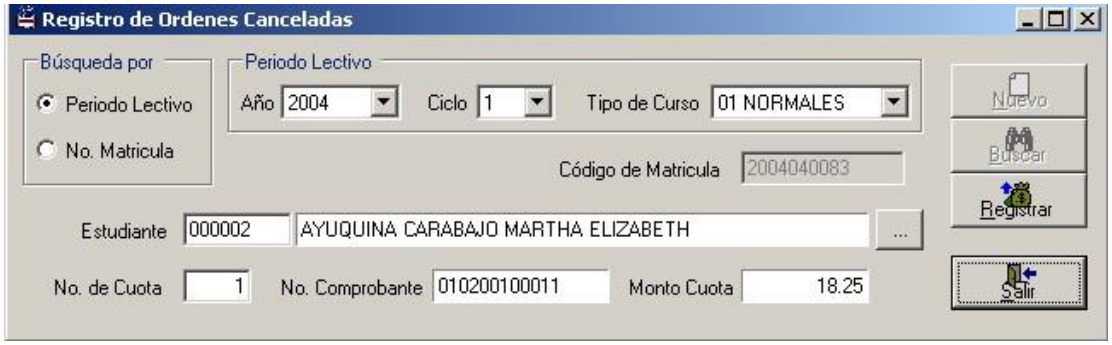


Figura 7.29. Ejemplo de Registro de Órdenes Canceladas

CAPITULO 8

8. Registro de Calificaciones

8.1. Calificaciones Parciales

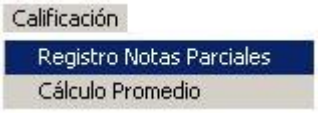


Figura 8.1. Acceso a Registro de Calificaciones

La pantalla inicial es la siguiente

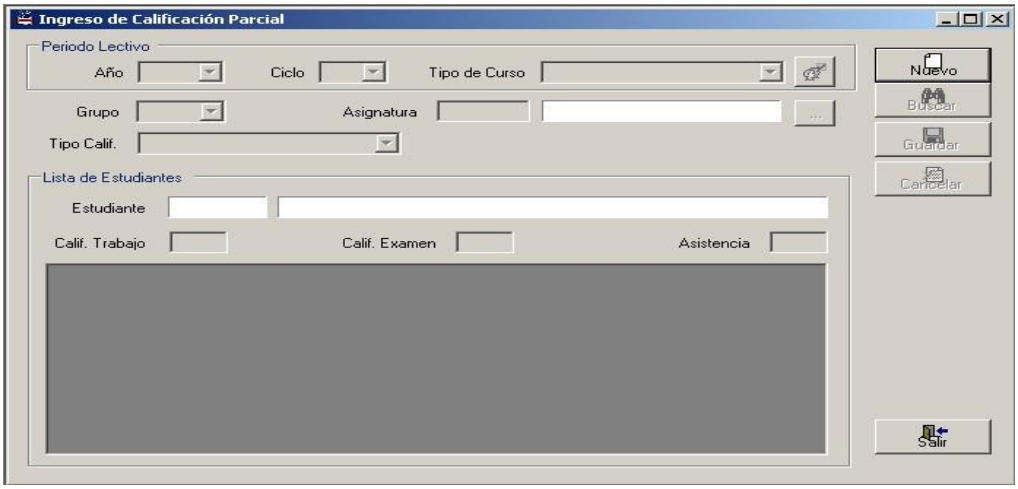


Figura 8.2. Ingreso de Calificación Parcial

En esta pantalla se deberá realizar lo siguiente:

- Seleccionar un “Período Lectivo” presionando el botón “Nuevo”. En esta pantalla se consideran sólo los periodos lectivos cuyo estado actual sea “Cierre de Matriculación”.
- Presionamos el botón “Buscar Grupos” en la cual se habilitarán los combos “Grupo” y “Tipo de Calificaciones”, de igual manera la caja de texto “Asignatura”.
- Al presionar el botón “Buscar” saldrá el listado de todos los estudiantes inscritos en dicho grupo que no se le haya ingresado ninguna nota con respecto a ese tipo de calificaciones.

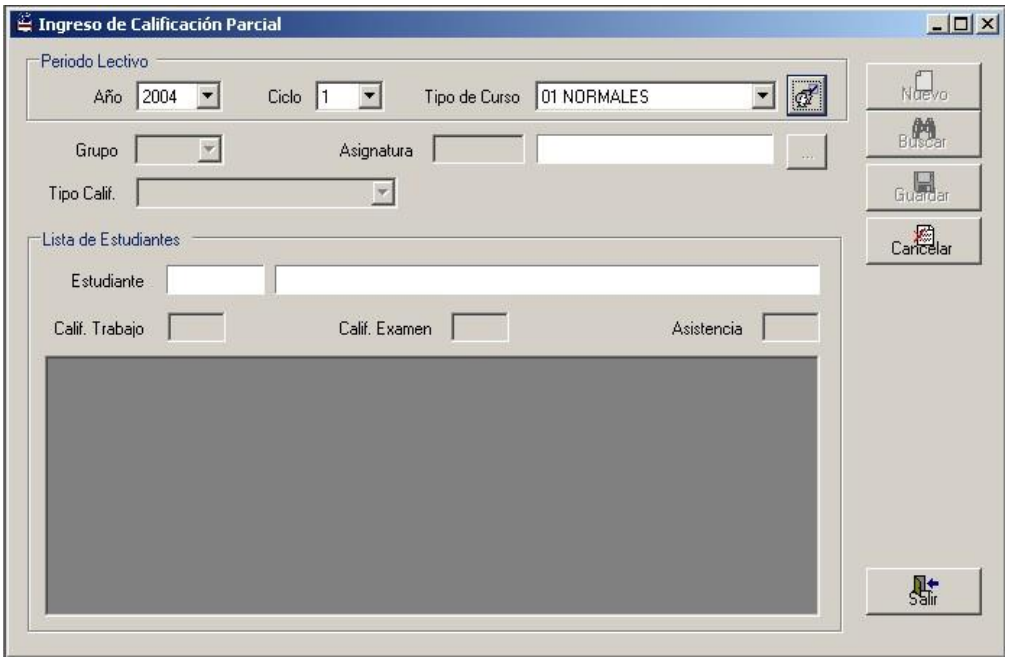


Figura 8.3. Ingreso de Calificación Parcial- Paso 1

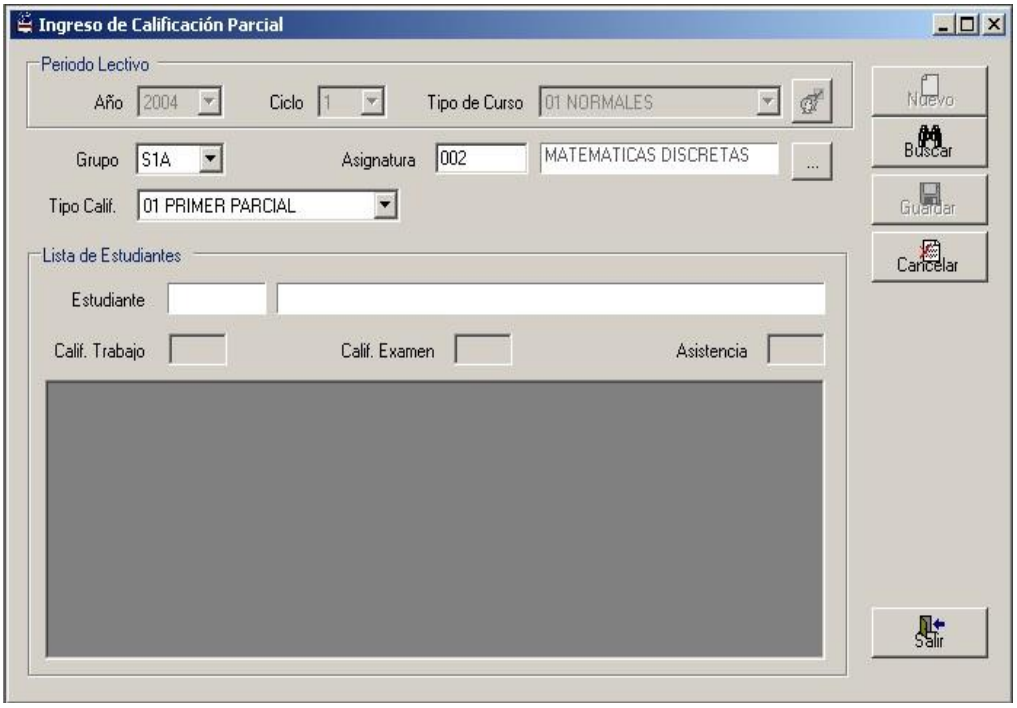


Figura 8.4. Ingreso de Calificación Parcial- Paso 2

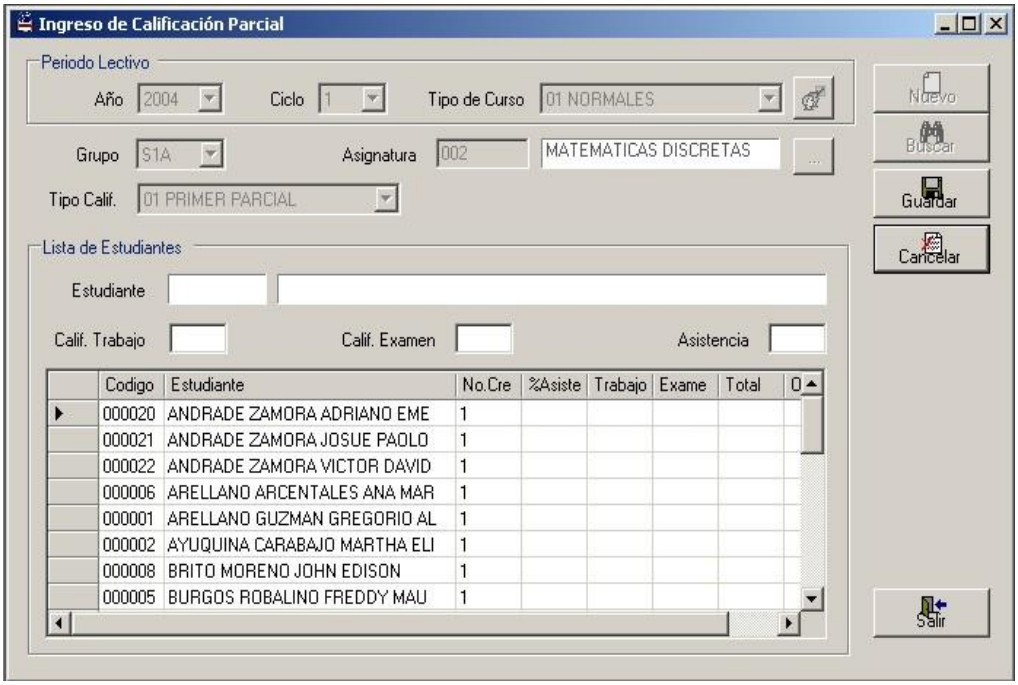


Figura 8.5. Ingreso de Calificación Parcial- Paso 3

Para ingresar los datos de cada uno de los estudiantes que se encuentran en el listado correspondiente a la calificación de trabajo, calificación de examen y de asistencia se deberá dar un clic en cada celda

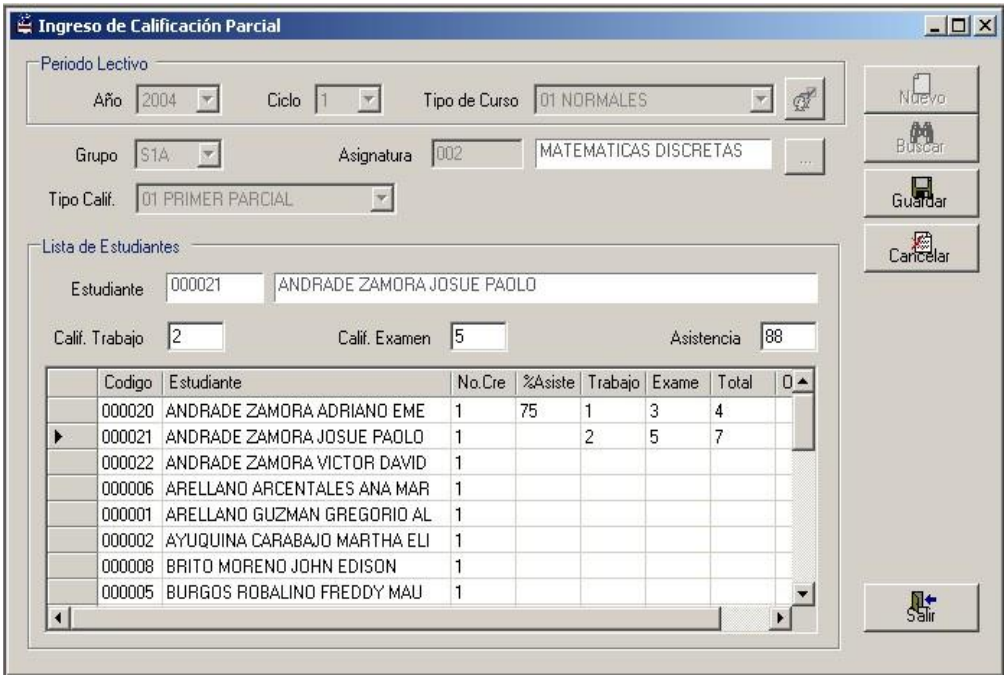


Figura 8.6. Ingreso de Calificación Parcial- Paso 4

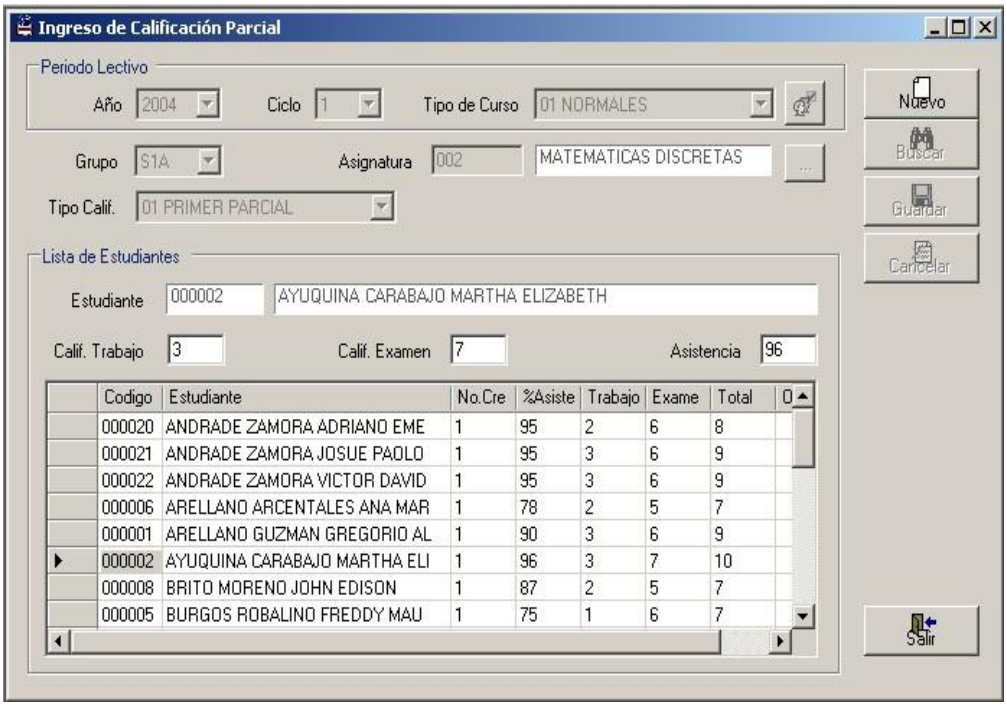


Figura 8.7. Ejemplo de Ingreso de Calificación Parcial

8.2. Promedio de Calificaciones

Mediante esta pantalla se generan los promedios de cada grupo respecto a una materia.

Se deben seguir los siguientes pasos:

- Se ingresa el periodo lectivo actual y se da clic en el botón “Buscar Grupos”
- Se elige el grupo que se desea sacar el promedio y se ingresa la asignatura.

- Se presiona el botón “Buscar” y sale la lista de estudiantes inscritos a esa asignatura en el curso seleccionado.

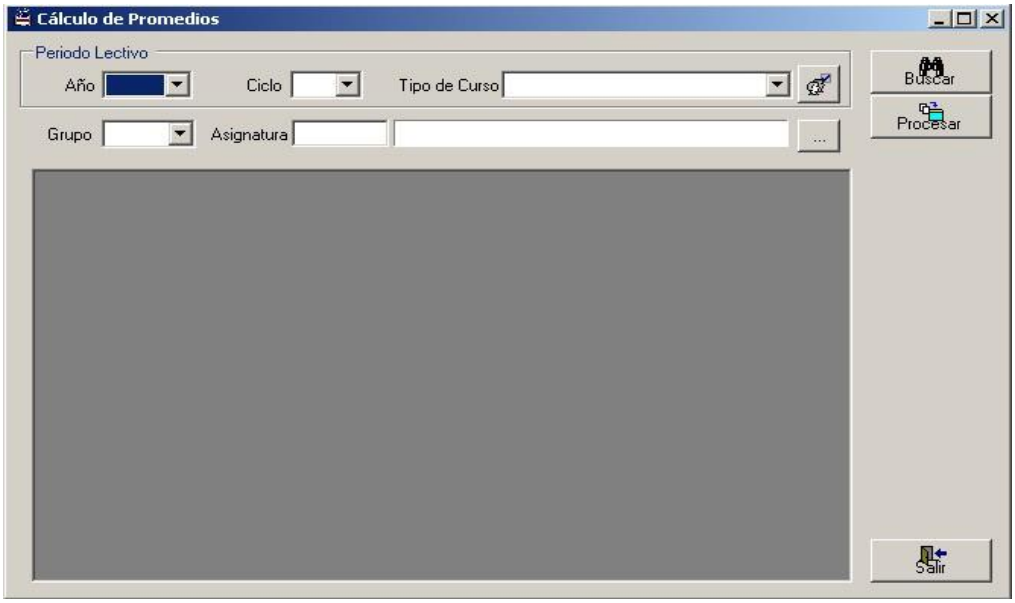


Figura 8.8. Cálculo de Promedios

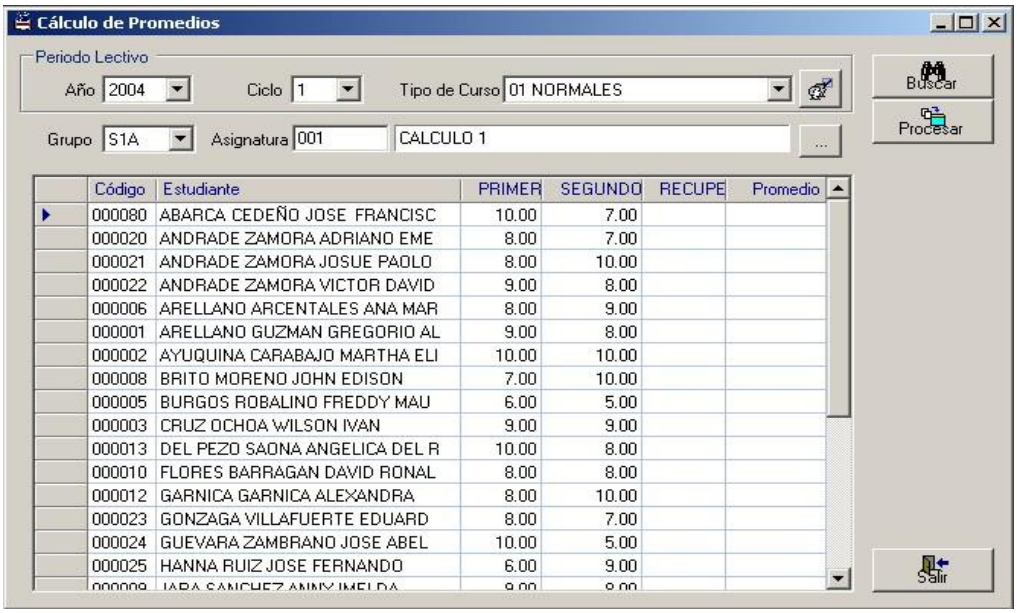


Figura 8.9. Ejemplo de Cálculo de Promedios

- Para generar los promedios basta dar clic en el botón “Procesar” y el sistema automáticamente muestra los promedios de cada estudiante de la lista.

CAPITULO 9

9. Registro de Solicitudes

9.1. Solicitudes Especiales

Figura 9.1. Acceso a Solicitudes

La pantalla inicial es la siguiente:

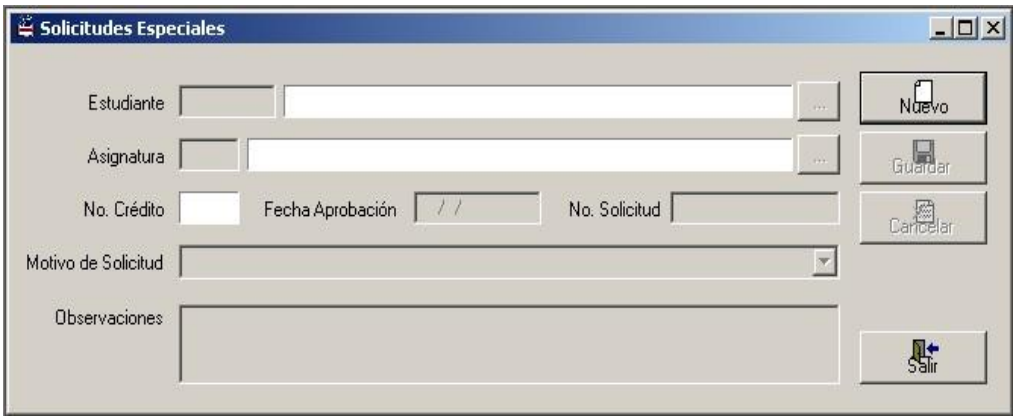


Figura 9.2. Solicitudes Especiales

En esta pantalla se deberá seguir los siguientes pasos:

- Presionar el botón “Nuevo” y luego se digitará el código del estudiante o bien seleccionándolo a través de la consulta presionando el botón derecho.
- Digitar el código de la asignatura o seleccionándolo a través de la consulta presionando el botón derecho.
- Ingresar la “Fecha de Aprobación” de la solicitud, así mismo el número de la misma.
- Seleccionar el “Motivo de Solicitud”, y presionar el botón “Guardar”.

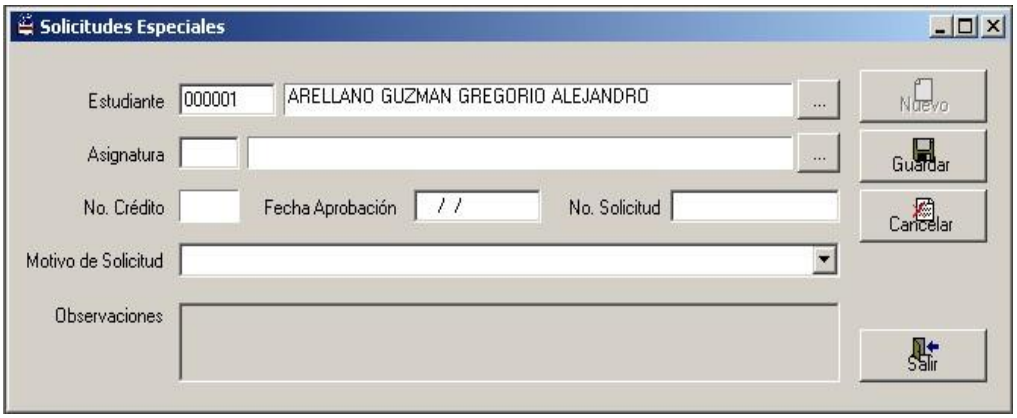


Figura 9.3. Solicitudes Especiales- Paso 1

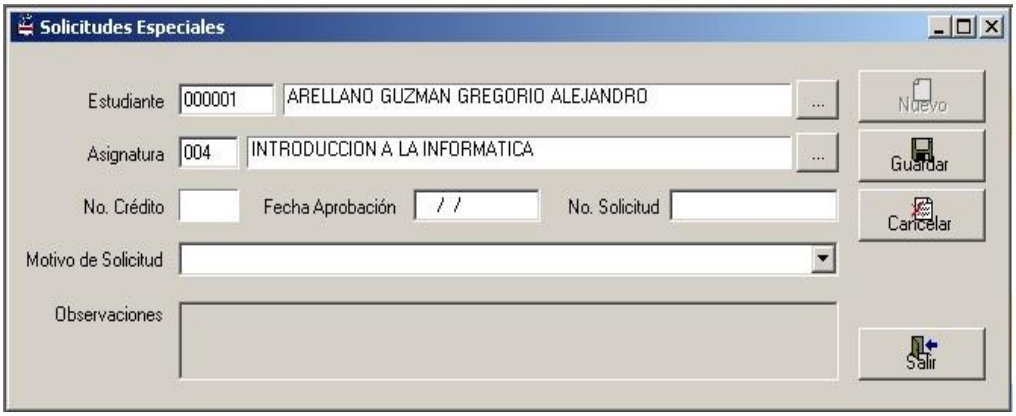


Figura 9.4. Solicitudes Especiales- Paso 2

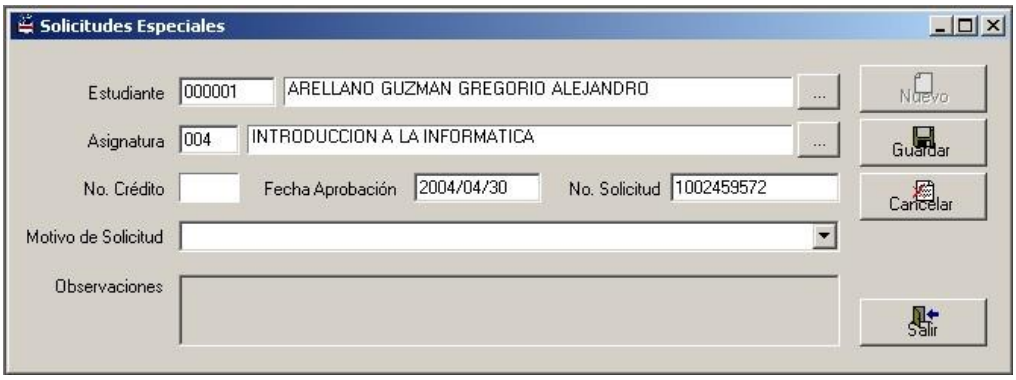


Figura 9.5. Solicitudes Especiales- Paso 3

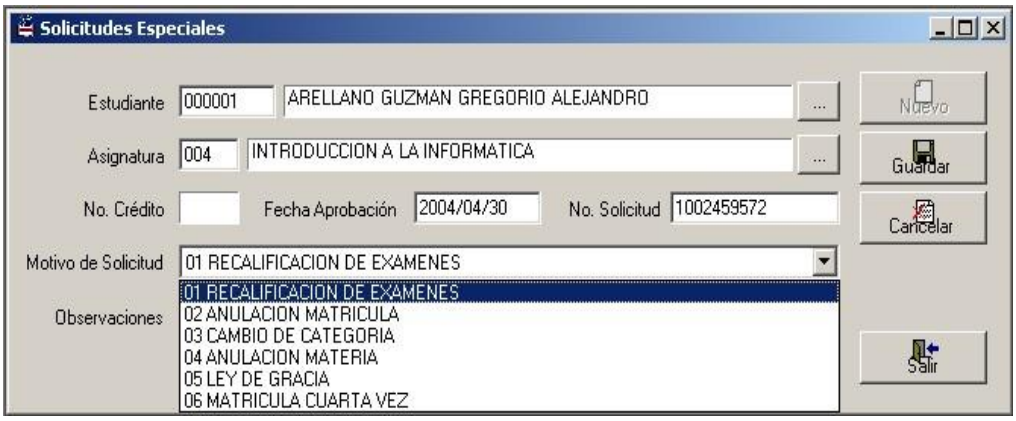


Figura 9.6. Solicitudes Especiales- Paso 4

9.2. Recalificación de Exámenes

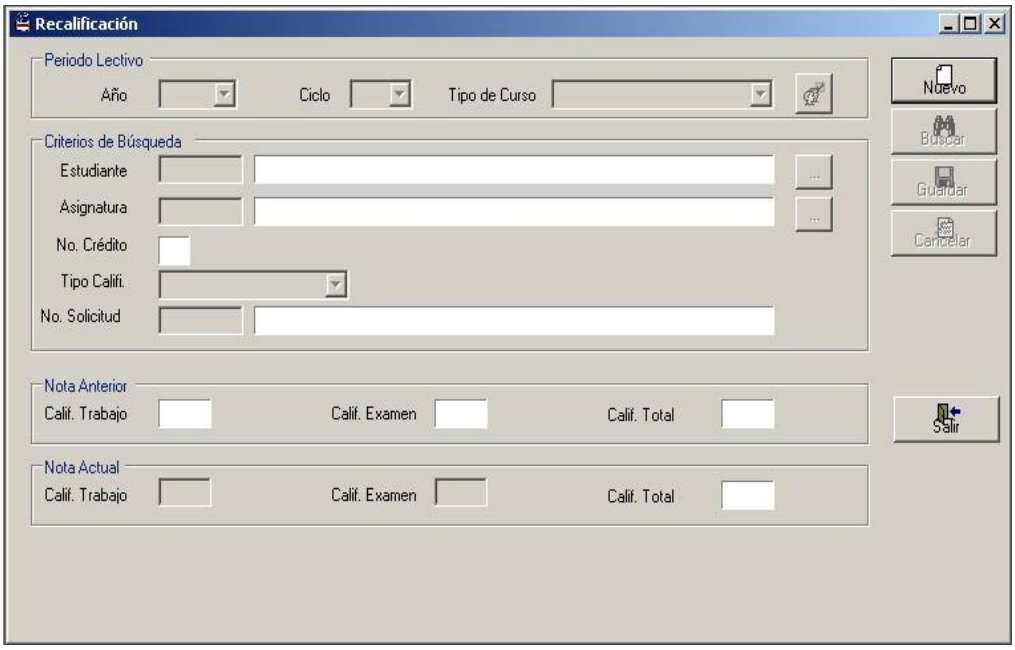


Figura 9.7. Recalificación

En dicha pantalla se deberá realizar lo siguiente:

- Seleccionar un “Período Lectivo” presionando el botón “Nuevo”. En esta pantalla sólo se consideran los periodos lectivos con estado “Cierre de Matriculación”.
- Al presionar el botón “Verificar Período Lectivo” se habilitarán las siguientes cajas de textos en lo que respecta a “Criterios de Búsqueda”:

  Estudiante: Se digitará el código del alumno o seleccionándolo a través de la consulta presionando el botón derecho.

Asignatura: Digitar el código de la asignatura o bien seleccionándolo a través del botón derecho.

No. Crédito: Aparecerá automáticamente el número de crédito.

Tipo de Calificación: Seleccionar el tipo de calificación ya sea del primer o segundo parcial, o bien, el de recuperación.

No. Solicitud: Se digitará el número de dicha solicitud, y a continuación se presionará el botón “Buscar”.

- En “Nota Anterior” aparecerán las siguientes notas: “Calif. Trabajo”, “Calif. Examen”, y “Calif. Total” si han sido registradas anteriormente.
- En lo que respecta a “Nota Actual” es para modificar las calificaciones de ese estudiante en el parcial y la asignatura indicada.

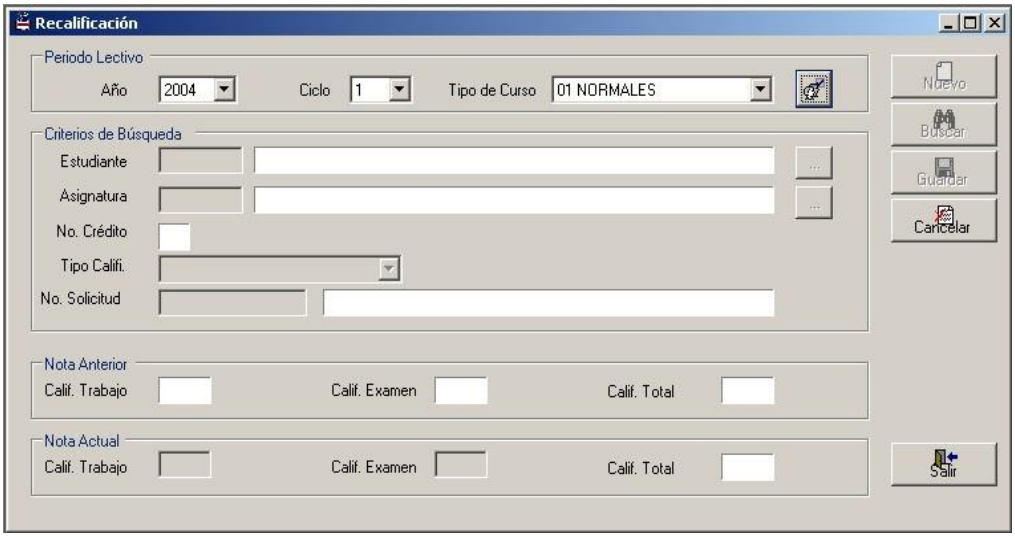


Figura 9.8. Recalificación- Paso 1

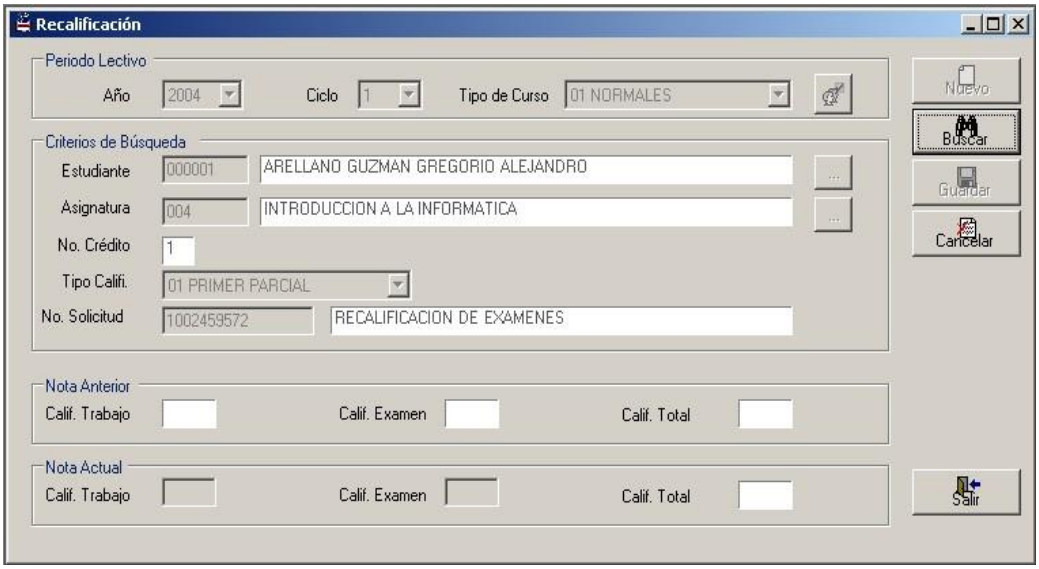


Figura 9.9. Recalificación- Paso2

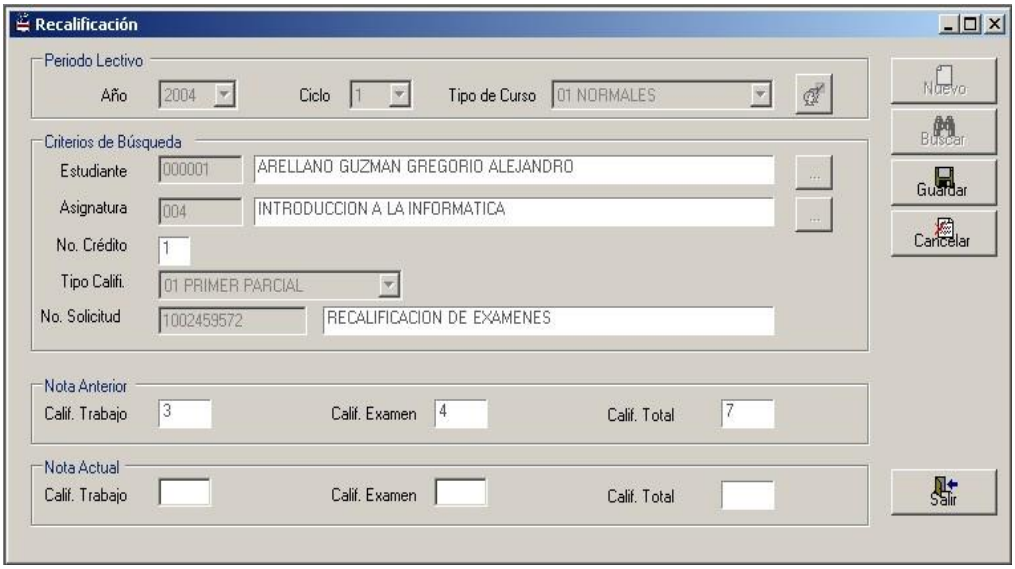


Figura 9.10. Recalificación- Paso 3

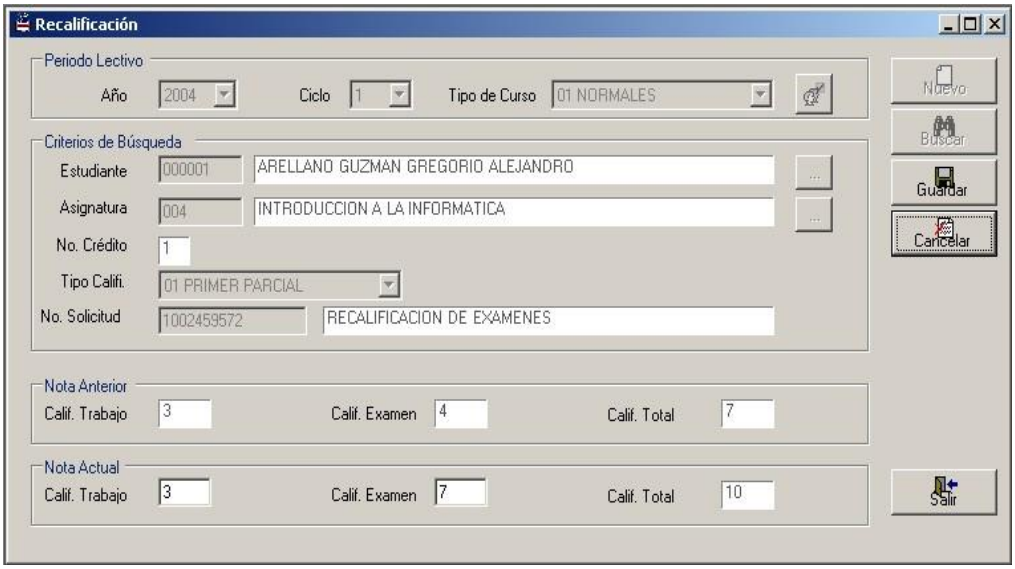


Figura 9.11. Recalificación- Paso 4

Es importante aclarar que la pantalla de recalificaciones servirá para actualizar una nota post-evaluada, es decir, que permitirá aclarar que la nota anterior está anulada porque existió una aprobación previa del Consejo Académico para reevaluar la calificación de un examen. En caso de modificar la calificación porque fue ingresad erróneamente, utilice la pantalla “Modificación de Notas” de la opción Calificación.

CAPITULO 10

10. Reportes del Sistema Académico

10.1. Informe de Estudiantes por grupo

La pantalla inicial será la siguiente:

Figura 10.1. Informe de Estudiantes por Grupo

Al ingresar dicha pantalla debemos realizar lo siguiente:

- Presionar el botón “Nuevo”, y seleccionamos un período lectivo.
- Presionamos el botón “Buscar Grupos” en la cual se selecciona el grupo que se desee sacar en el reporte, así mismo si se desea incluir al docente se deberá hacer un clic en “Incluir Docente”. Con respecto a “Asignatura” se seleccionará alguna o todas las materias correspondientes a dicho grupo.
- Por último, se deberá presionar el botón “Imprimir” en la cual saldrá el reporte del listado de los estudiantes.

Figura 10.2. Informe de Estudiantes por Grupo-Paso 1

Figura 10.3. Informe de Estudiantes por Grupo-Paso 2

UNIVERSIDAD DE GUAYAQUIL

Unid. Académica INGENIERIA DE SISTEMAS

LISTADO DE ESTUDIANTES

GRUPO S1A

Fecha: 2004/06/01
Hora: 11:04:03
No. Página: 1

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo: 1

Usuario: G_ARELLANO_G

- RptLista_Estudpt -

DOCENTE ING CIV JOSE LUIS NAVARRO

Asignatura CALCULO 1

No.	APELLIDOS	NOMBRES					
1	ANDRADE ZAMORA	ADRIANO EMERSON					
2	ANDRADE ZAMORA	VICTOR DAVID					
3	ANDRADE ZAMORA	JOSUE PAOLO					
4	ARELLANO ARCENTALES	ANA MARIA					
5	ARELLANO GUZMAN	GREGORIO ALEJANDRO					
6	AYOQUINA CARABAJO	MARTHA ELIZABETH					
7	BRITO MORENO	JOHN EDISON					
8	BURGOS ROBALINO	FREDDY MAURICIO					
9	CRUZ OCHOA	WILSON IVAN					
10	DEL PEZO SAONA	ANGELICA DEL ROCIO					

Figura 10.4. Ejemplo del Informe de Estudiantes por Grupo

10.2. Informe de Inscripciones por Estudiante

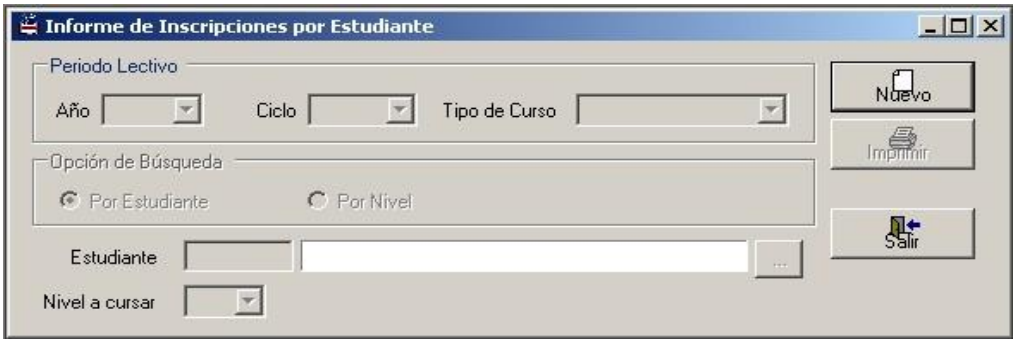


Figura 10.5. Informe de Inscripciones por Estudiante

Al ingresar al registro de inscritos se debe de seguir los siguientes pasos:

- Presionar el botón “Nuevo” y seleccionamos un período lectivo.
- En “Opción de Búsqueda” en **estudiante** se habilitará la caja de texto “Estudiante” en donde se digitará el código del estudiante o bien se lo selecciona a través de la consulta presionando el botón derecho. Si se elige la opción **nivel** se habilitará un combo que contiene todos los niveles abiertos.
- Finalmente, se presionará el botón “Imprimir” donde saldrá el informe de inscripciones por estudiante.

Figura 10.6. Informe de Inscripciones por Estudiante- Paso 1

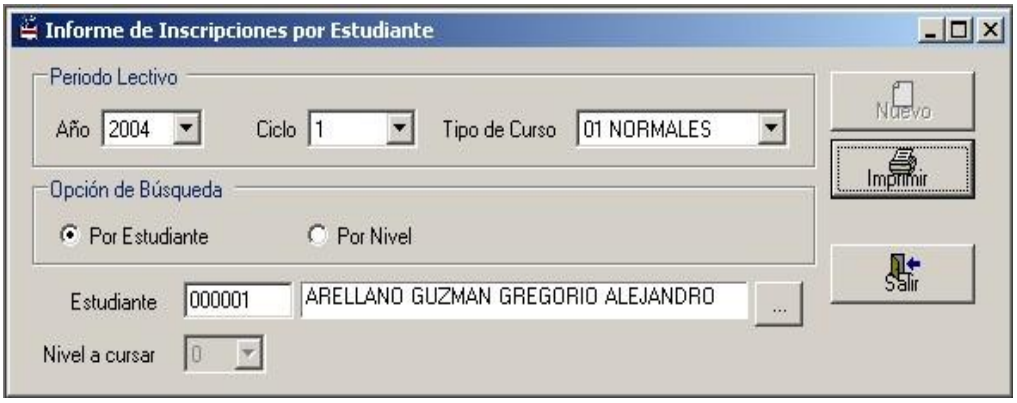


Figura 10.7. Informe de Inscripciones por Estudiante- Paso2

UNIVERSIDAD DE GUAYAQUIL

Unid. Académica INGENIERIA DE SISTEMAS

INSCRIPCIONES

Fecha: 2004/06/01
Hora: 13:13:00
No. Página: 12

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo: 1

Usuario: G_ARELLANO_G

- RptLista_Insc.rpt -

Datos del Estudiante

Código: 000001 **Nombres:** ARELLANO GUZMAN GREGORIO ALEJANDRO

No.	GRUPO	SUBGRU	ASIGNATURA	NIVEL	VEZ	DOCENTE
89	S1A		0102 CALCULO 1	1	1	ING CIV JOSE LUIS NAVARRO
90	S1A		0102 MATEMATICAS DISCRETAS	1	1	ING CIV JOSE HUAYAMABE
91	S1A		0102 PROGRAMACION 1	1	1	ANL SIST YOSEPH ONTANEDA
92	S1A		0102 INTRODUCCION A LA	1	1	ING SIST FERNANDO CASTRO
93	S1A		0102 ENFOQUE DE SISTEMAS	1	1	ING SIST ERICK RICAURTE
94	S1A		0102 CONTABILIDAD 1	1	1	ING CIV ALEJANDRO BARBA
95	S1A		0102 INGLES TECNICO 1	1	1	LCDO ROSEMARIE CAICEDO
96	S1A		0102 IDIOMA ESPAÑOL 1	1	1	LCDO MIGUEL SARMIENTO

Entregado por

Figura 10.8. Ejemplo de Informe de Inscripciones por Estudiante

Ahora a continuación daremos un ejemplo cuando se selecciona por nivel:

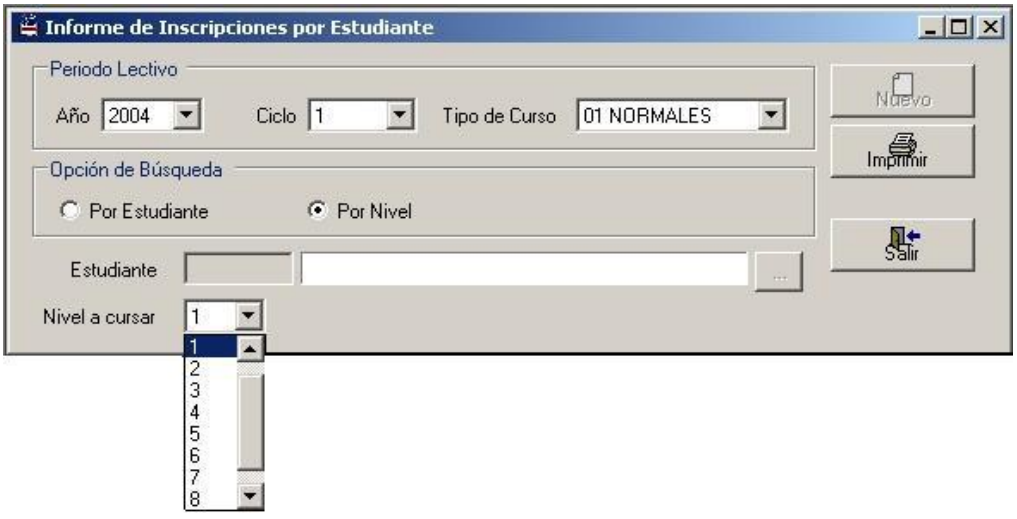


Figura 10.9. Informe de Inscripciones por Estudiante- Opción Nivel

Al escoger la opción **nivel** saldrá más de un informe de todo el alumnado que se haya registrado a dicho nivel.

10.3. Consulta de Docentes por Subunidad

Figura 10.10. Consulta de Docentes por Subunidad

Una vez ingresado en el registro de Docentes por Subunidad realizamos lo siguiente:

- Presionamos el botón “Nuevo” para seleccionar un período lectivo.
- Por último, presionar el botón “Imprimir” en donde saldrá el listado de los docentes por subunidad de acuerdo al área académica.

UNIVERSIDAD DE GUAYAQUIL

LISTADO DE DOCENTES

Fecha: 2004/05/11
Hora: 8:45:00
No. Página: 1

CARRERA INGENIERIA DE SISTEMAS

Año Lectivo: 2004
Curso: NORMALES Ciclo: 1

Usuario: G_ARELLANO_G

- RptLista_Doce.rpt -

Areas Académicas	*Código*	*Nombres*
ADMINISTRACION		
	00008	ING CIV ALEJANDRO BARBA
	00016	EC CARLOS LUIS ORTIZ
	00017	CPA SOFIA ARREAGA
	00023	ANL SIST JAVIER BRITO
	00024	ING COM CARMEN CARDENAS
CULTURA GENERAL		
	00009	LCDO ROSA SANTOS
	00010	ING CIV MARCOS VARAS
	00012	LCDO MIGUEL SARMIENTO
	00018	LCDO ROSEMARIE CAICEDO
	00019	AB JOSE ORDEÑANA
INFORMATICA		
	00006	ING CIV EDUARDO LOPEZ
	00007	ANL SIST YOSEPH ONTANEDA
	00015	ING ELECTRON MILTON MARIDUEÑA
	00021	ING SIST GLEISTON GUERRERO
	00022	ING ELECT VICTOR LOPEZ
	00025	ING SIST MIGUEL AVELLAN
	00028	ING SIST JOSE VILLALTA
	00029	ING SIST FERNANDO CASTRO
	00030	ING SIST ERICK RICAURTE
	00031	ING SIST CARLOS RIVERA MENDEZ

Figura 10.11. Ejemplo del Listado de Docentes por Áreas Académicas

10.4. Reporte de Horarios por Grupos

Figura 10.12. Acceso al Reporte de Horarios

La pantalla inicial se presentará de la siguiente manera:

Figura 10.13. Reporte de Horarios por Grupos

En esta pantalla se deberá realizar los siguientes pasos:

- Presionar el botón “Nuevo” para seleccionar un período lectivo. Al mismo tiempo se habilitará el combo “Sección” en donde se escogerá la sección seleccionada.
- Presionamos el botón “Buscar Grupos” en la cual se escogerá el grupo que se desee sacar en el reporte.

- Finalmente, presionamos el botón “Imprimir” en donde se mostrará el respectivo registro horario por grupos.

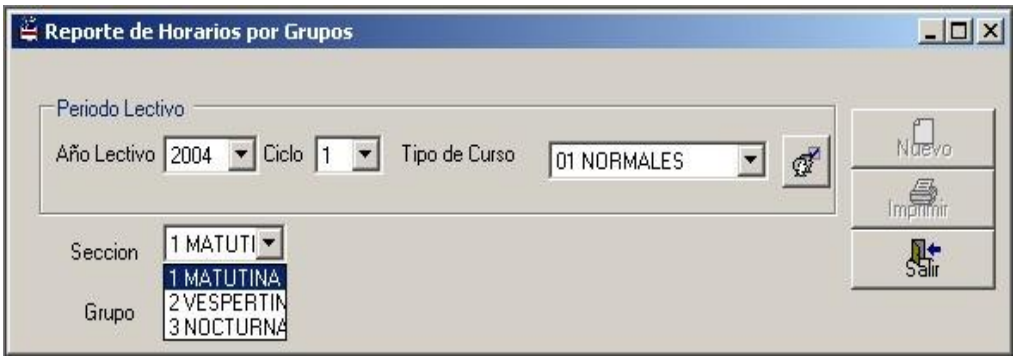


Figura 10.14. Reporte de Horarios por Grupos- Paso 1

Figura 10.15. Repote de Horarios por Grupos- Paso 2

UNIVERSIDAD DE GUAYAQUIL
CARRERA INGENIERIA DE SISTEMAS

HORARIO
GRUPO S1A

Fecha: 2004/04/16
Hora: 10:03:15
No. Página: 1

Año Lectivo: 2004
Curso: NORMALES Ciclo: 1
Sección: MATUTINA

Usuario: G_ARELLANO_G

- RptHorar_Grup rpt -

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
1 07h00-08h00	MATEMATICAS	PROGRAMACION I	CONTABILIDAD I	CALCULO I	ENFOQUE DE SISTEMAS		
2 07h00-08h00	MATEMATICAS	PROGRAMACION I	CONTABILIDAD I	CALCULO I	ENFOQUE DE SISTEMAS		
3 08h00-09h00	CALCULO I	INTRODUCCION A LA	INGLES TECNICO I	CONTABILIDAD I			
4 09h00-10h00	CALCULO I	INTRODUCCION A LA	INGLES TECNICO I	INTRODUCCION A LA			
5 09h00-10h00	IDIOMA ESPANOL I	IDIOMA ESPANOL I	MATEMATICAS	PROGRAMACION I			

Figura 10.16. Ejemplo deL Registro de Horario por Grupos

10.5. Reporte de Horarios por Docentes

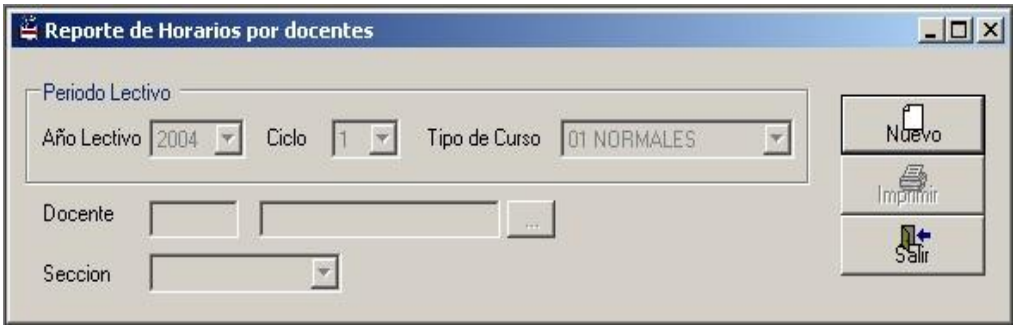


Figura 10.17. Reporte de Horarios por Docentes

Al ingresar a dicha pantalla realizamos lo siguiente:

- Presionamos el botón “Nuevo” en donde se habilitarán tanto la caja de texto “Docente” en la cual se digitará el código del docente o seleccionándolo a través de la consulta presionando

el botón derecho, así mismo el combo “Sección” en la cual aparecerá la sección seleccionada a dicho docente.

- Por último, se presiona el botón “Imprimir” en donde aparecerá el horario que corresponde al docente escogido.

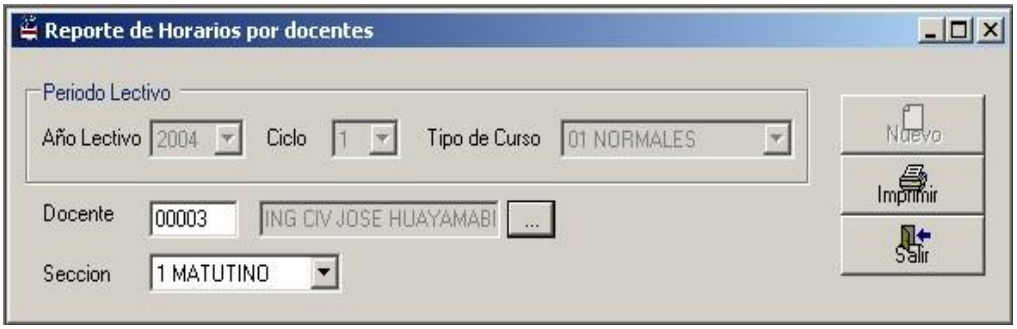


Figura 10.18. Reporte de Horarios por Docentes- Paso 1

UNIVERSIDAD DE GUAYAQUIL
CARRERA INGENIERIA DE SISTEMAS

HORARIO
DOCENTE: ING CIV JOSE

Fecha: 2004/04/16
Hora: 11:24:00
No. Página: 1

Año Lectivo: 2004
Curso: NORMALES Ciclo: 1
Sección: 1 MATUTINO

Usuario:

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
1 0700-0800	MATEMATICAS SIA						
2 0750-0850	MATEMATICAS SIA						
3 0800-0900							
4 0900-1000							
5 0950-1050			MATEMATICAS SIA				

Figura 10.19. Ejemplo de Registro de Horarios por Docentes

10.6. Resumen de Descuentos y Aranceles por Período

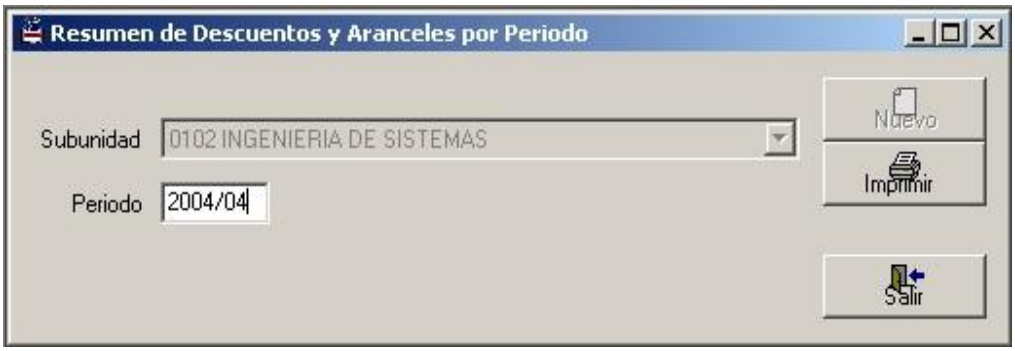
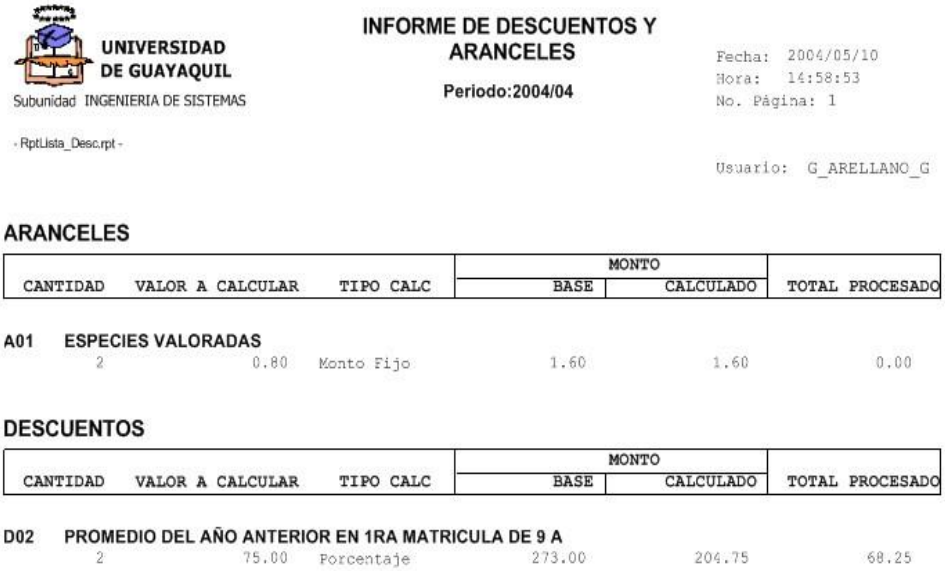


Figura 10.20. Resumen de Descuentos y Aranceles

Este reporte muestra cuánto se ha descontado y se ha cobrado de aranceles en las órdenes de pago emitidas por concepto de matriculación.

En esta pantalla se deberá realizar lo siguiente:

- Se presiona el botón “Nuevo” y se habilita automáticamente la caja de texto del período.
- Se digita el período en formato “Año/Mes” y a continuación se presiona el botón “Imprimir”.



UNIVERSIDAD DE GUAYAQUIL
Subunidad INGENIERIA DE SISTEMAS
- RptLista_Desc.rpt -

INFORME DE DESCUENTOS Y ARANCELES
Periodo:2004/04

Fecha: 2004/05/10
Hora: 14:58:53
No. Página: 1
Usuario: G_ARELLANO_G

ARANCELES

	CANTIDAD	VALOR A CALCULAR	TIPO CALC	MONTO BASE	MONTO CALCULADO	TOTAL PROCESADO
A01	ESPECIES VALORADAS					
	2	0.80	Monto Fijo	1.60	1.60	0.00

DESCUENTOS

	CANTIDAD	VALOR A CALCULAR	TIPO CALC	MONTO BASE	MONTO CALCULADO	TOTAL PROCESADO
D02	PROMEDIO DEL AÑO ANTERIOR EN 1RA MATRICULA DE 9 A					
	2	75.00	Porcentaje	273.00	204.75	68.25

Figura 10.21. Informe de Descuentos y Aranceles

10.7. Reporte de Estudiantes Matriculados

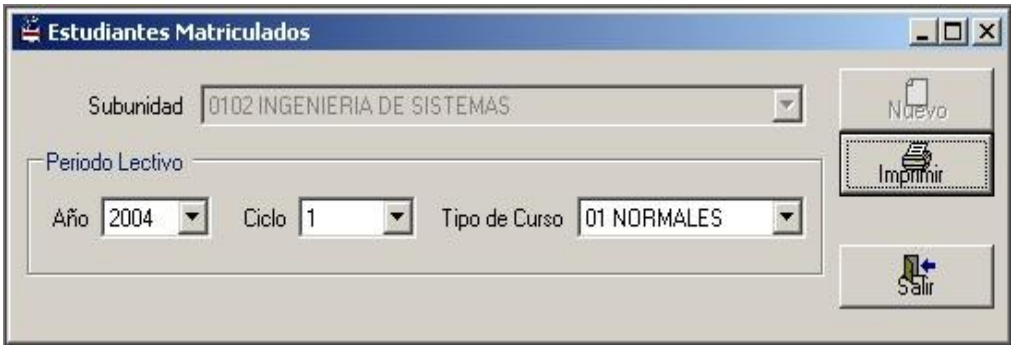
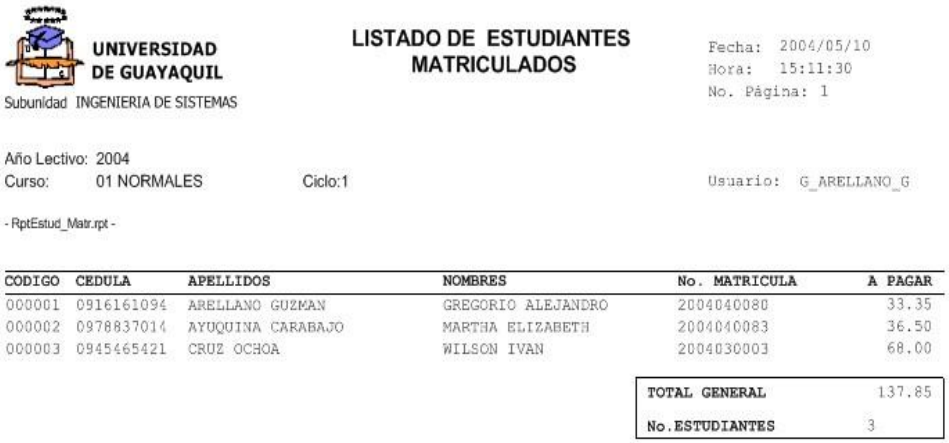


Figura 10.22. Estudiantes Matriculados

En dicha pantalla, al presionar el botón “Nuevo” se deberá seleccionar el período lectivo y presionar el botón “Imprimir” en donde aparecerá el

informe de aquellos estudiantes inscritos a los que se les haya procesado la orden de pago.



UNIVERSIDAD DE GUAYAQUIL
Subunidad INGENIERIA DE SISTEMAS

LISTADO DE ESTUDIANTES MATRICULADOS

Fecha: 2004/05/10
Hora: 15:11:30
No. Página: 1

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo:1

Usuario: G_ARELLANO_G

- RptEstud_Matr.rpt -

CODIGO	CEDULA	APELLIDOS	NOMBRES	No. MATRICULA	A PAGAR
000001	0916161094	ARELLANO GUZMAN	GREGORIO ALEJANDRO	2004040080	33.35
000002	0978837014	AYUQUINA CARABAJO	MARTHA ELIZABETH	2004040083	36.50
000003	0945465421	CRUZ OCHOA	WILSON IVAN	2004030003	68.00

TOTAL GENERAL	137.85
No.ESTUDIANTES	3

Figura 10.23. Listado de Estudiantes Matriculados

10.8. Reporte de Estadísticas de Inscritos

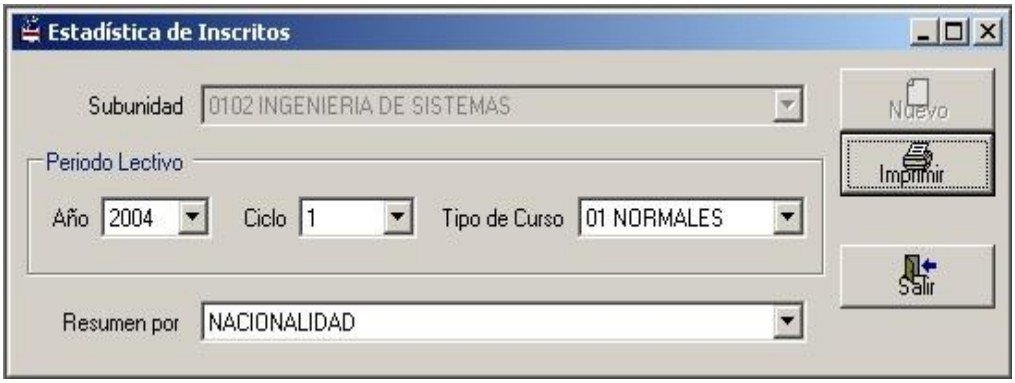


Figura 10.24. Estadísticas de Inscritos

En esta pantalla se debe realizar lo siguiente:

- Al presionar el botón “Nuevo” se habilitará automáticamente los combos del “Período Lectivo” y el tipo de resumen estadístico que se desee imprimir.
- Se selecciona el período lectivo y el resumen sea por nacionalidad, edad, sexo, etc.

Figura 10.25. Estadísticas de Estudiantes Inscritos por Nacionalidad

Figura 10.26. Estadísticas de Estudiantes Inscritos por Categoría de Colegio

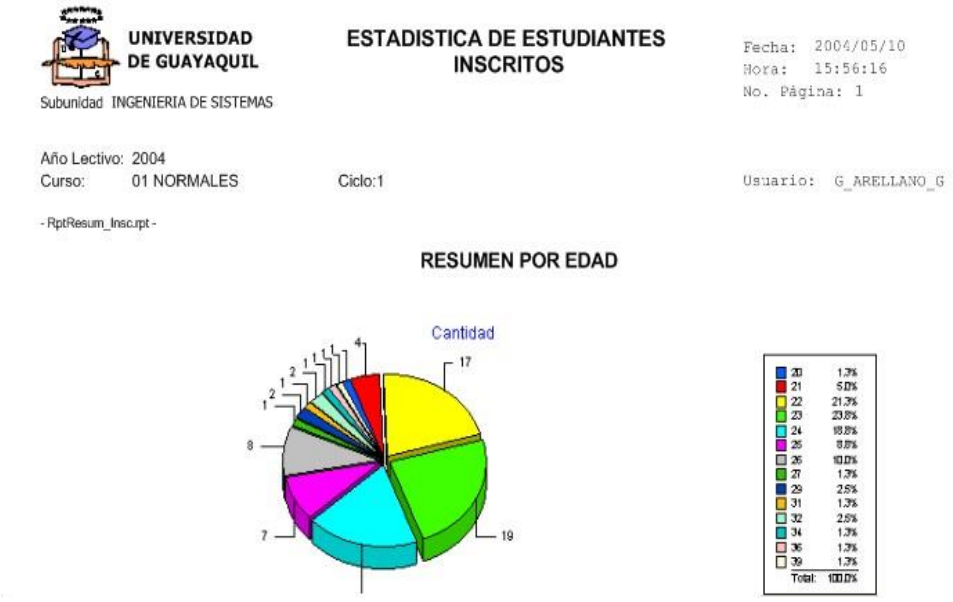


Figura 10.27. Estadísticas de Estudiantes Inscritos por Edad

Figura 10.28. Estadísticas de Estudiantes Inscritos por Sexo

10.9. Informe de Calificaciones por Asignatura

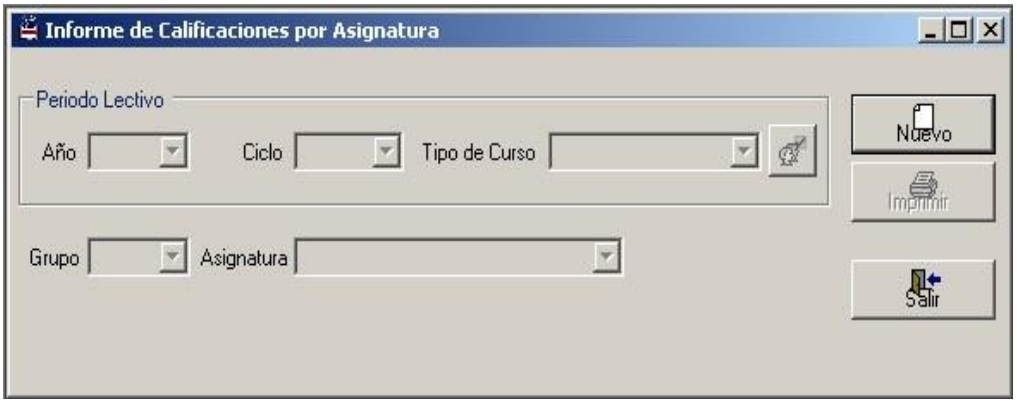


Figura 10.29. Informe de Calificaciones por Asignatura

Para esta pantalla se debe efectuar los siguiente pasos:

- Presionar el botón “Nuevo”, y seleccionamos el período lectivo.

- Presionamos el botón “Buscar Grupos” en la cual se selecciona el grupo que se desee sacar en el reporte, así mismo, con respecto a “Asignatura” se seleccionará alguna o todas las materias correspondiente a dicho grupo.
- Por último, se deberá presionar el botón “Imprimir” en la cual saldrá el reporte del listado de las calificaciones por asignatura.

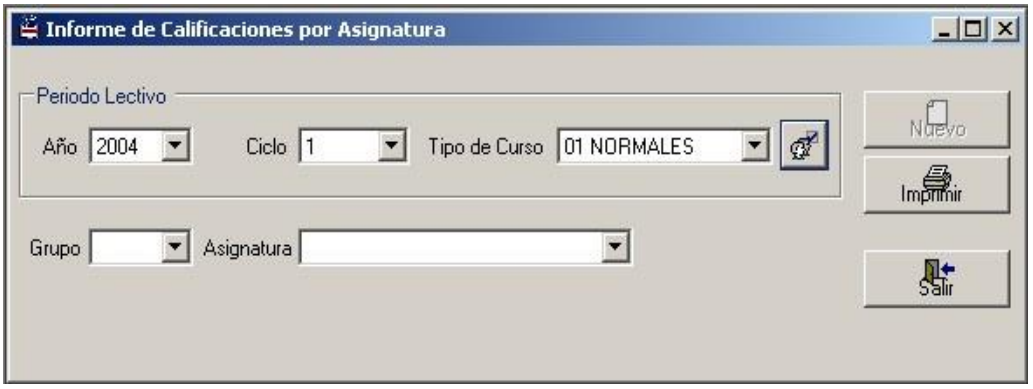


Figura 10.30. Informe de Calificaciones por Asignatura- Paso1

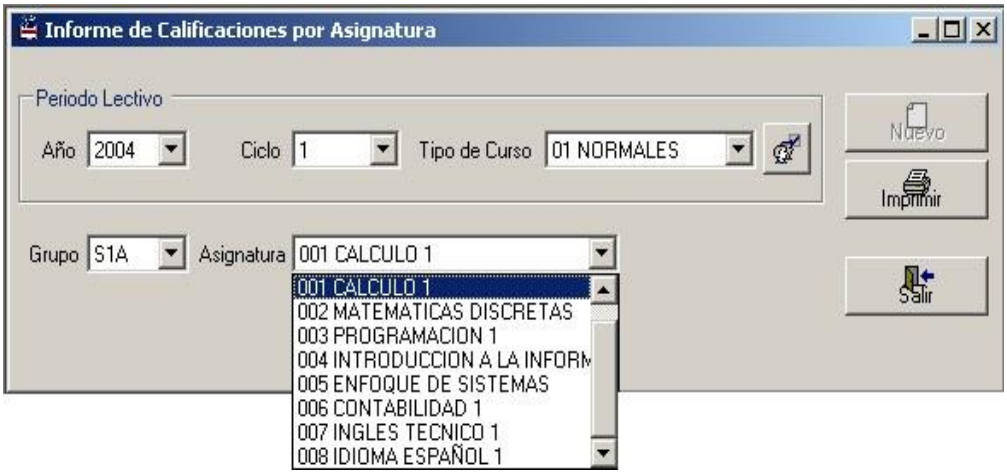


Figura 10.31. Informe de Calificaciones por Asignatura- Paso 2

UNIVERSIDAD DE GUAYAQUIL

Sub1.00010INGENIERIA DE SISTEMAS

LISTADO DE CALIFICACIONES

GRUPO S1A

Fecha: 2004/05/10
Hora: 16:32:16
No. Página: 1

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo:1

Usuario: G_ARELLANO_G

- RptLista_Cali.rpt -

DOCENTE ING CIV JOSE LUIS NAVARRO

Asignatura CALCULO 1

ESTUDIANTES	PRIMER PARCIAL	SEGUNDO PARCIAL
01.- ABARCA CEDEÑO JOSE FRANCISCO	10.00	7.00
02.- ANDRADE ZAMORA ADRIANO EMERSON	8	7
03.- ANDRADE ZAMORA JOSUE PAOLO	8	10
04.- ANDRADE ZAMORA VICTOR DAVID	9	8

Figura 10.32. Listado de Calificaciones por Asignatura

10.10. Resumen de Calificaciones por Estudiante

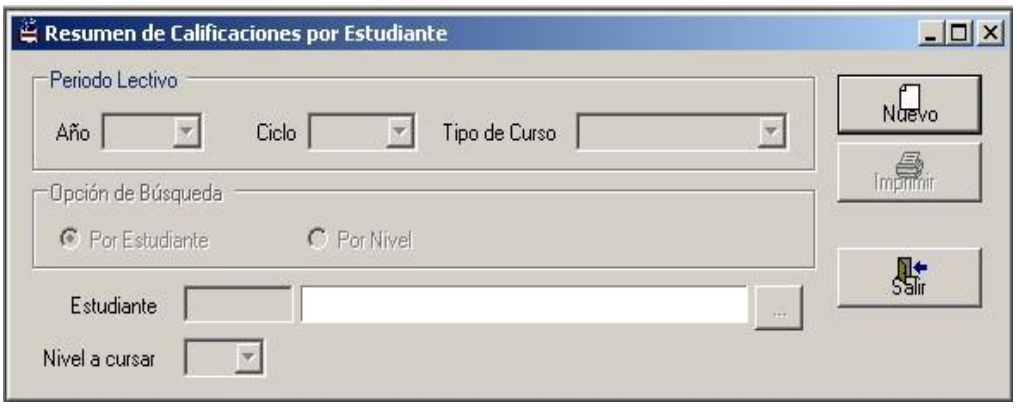


Figura 10.33. Resumen de Calificaciones por Estudiante

En dicha pantalla se deberá realizar lo siguiente:

- Presionar el botón “Nuevo” y seleccionamos un período lectivo.

- En “Opción de Búsqueda” en **estudiante** se habilitará la caja de texto “Estudiante” en donde se digitará el código del estudiante o bien se lo selecciona a través de la consulta presionando el botón derecho. Si se elige la opción **nivel** se habilitará un combo que contiene todos los niveles abiertos.
- Finalmente, se presionará el botón “Imprimir” donde saldrá el informe de inscripciones por estudiante.

Figura 10.34. Resumen de Calificaciones por Estudiante- Paso1

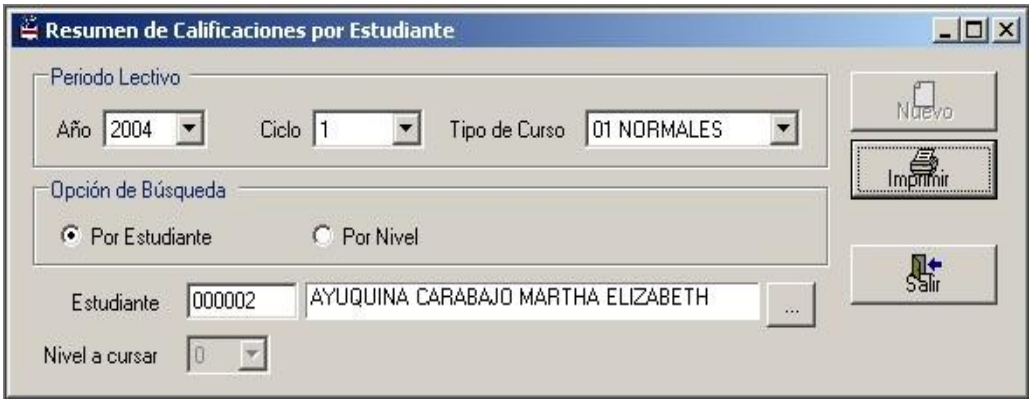


Figura 10.35. Resumen de Calificaciones por Estudiante- Paso2

UNIVERSIDAD DE GUAYAQUIL

Subunidad INGENIERIA DE SISTEMAS

RESUMEN DE CALIFICACIONES

Fecha: 2004/05/11
Hora: 9:20:44
No. Página: 1

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo:1

Usuario: G_ARELLANO_G

- RptLista_Cali_x_Estu.rpt -

Datos del Estudiante
Código: 000002 **Nombres:** AYUQUINA CARABAJO MARTHA ELIZABETH

	PRIMER PARCIAL	SEGUNDO PARCIAL
CALCULO 1	10.00	10.00
MATEMATICAS DISCRETAS	6.80	10.00

Figura 10.36. Ejemplo de Resumen de Calificaciones por Estudiante

10.11. Detalle de Calificaciones Parciales

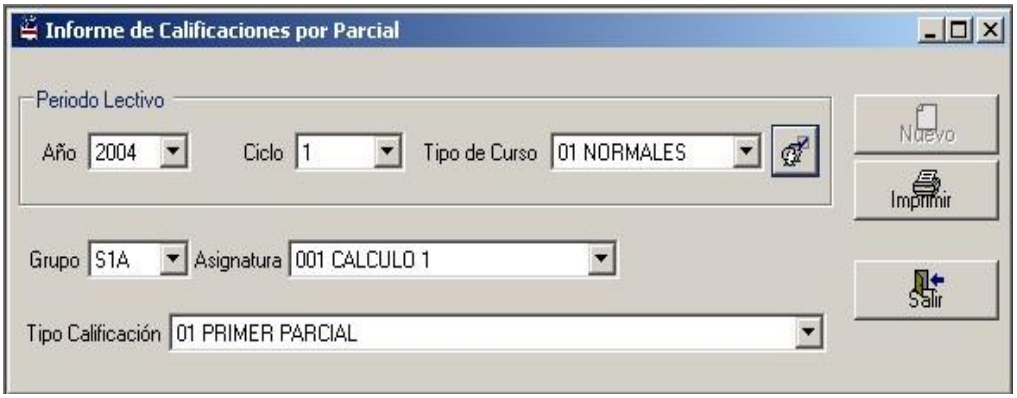


Figura 10.37. Pantalla de Informe de Calificaciones Parciales

Este reporte muestra un detalle de las calificaciones de cada estudiante dentro de un grupo en cada materia. Para emitir este reporte se siguen los siguientes pasos:

- Presionamos el botón “Nuevo” y escogemos el período lectivo y se da clic en el botón “Buscar Grupos”
- Se selecciona el grupo, se ingresa la asignatura y a continuación se elige el parcial.
- Se presiona el botón “Imprimir” y saldrá un reporte similar al siguiente:

UNIVERSIDAD DE GUAYAQUIL
Subunidad INGENIERIA DE SISTEMAS

LISTADO DE CALIFICACIONES
GRUPO S1A

Fecha: 2004/05/13
Hora: 12:14:40
No. Página: 1

Año Lectivo: 2004
Curso: 01 NORMALES Ciclo:1

Usuario: G_ARELLANO_G

- RptCalif_Parc.rpt -

DOCENTE ING CIV JOSE LUIS NAVARRO
Asignatura CALCULO 1

Código	APELLIDOS Y NOMBRES	% ASIST	TRABAJO	EXAMEN	CALIF. TOTAL	NOTA LETRAS
000080	ABARCA CEDEÑO JOSE FRANCISCO	25.00	3.00	7.00	10.00	DIEZ
000020	ANDRADE ZAMORA ADRIANO EMERSON	75.00	3.00	5.00	8.00	OCHO
000021	ANDRADE ZAMORA JOSUE PAOLO	80.00	3.00	5.00	8.00	OCHO
000022	ANDRADE ZAMORA VICTOR DAVID	89.00	2.00	7.00	9.00	NUEVE
000006	ARELLANO ARCENTALES ANA MARIA	99.00	2.00	6.00	8.00	OCHO
000001	ARELLANO GUZMAN GREGORIO ALEJANDRO	99.00	3.00	6.00	9.00	NUEVE
000002	AYUQUINA CARABAJO MARTHA ELIZABETH				10.00	DIEZ
000008	BRITO MORENO JOHN EDISON				7.00	SIETE
000005	BURGOS ROBALINO FREDDY MAURICIO	99.00	2.00	4.00	6.00	SEIS
000003	CRUZ OCHOA WILSON IVAN	98.00	3.00	6.00	9.00	NUEVE
000013	DEL PEZO SAONA ANGELICA DEL ROCIO	72.00	3.00	7.00	10.00	DIEZ

TblEstudiantes.Nombres (Cadena)

Figura 10.38. Ejemplo de Detalle de Calificaciones por Parcial

CAPITULO 11

11. SEGURIDAD

11.1. Cambio de usuario

En este formulario el usuario puede terminar su sesión e iniciar la sesión de otro usuario.

En la pantalla se escribirá el usuario y su contraseña. Es la misma pantalla que aparece cuando arranca el sistema por primera vez, es decir, que contiene las mismas validaciones y formas de navegar. Para mayor información consulte Capítulo 1.

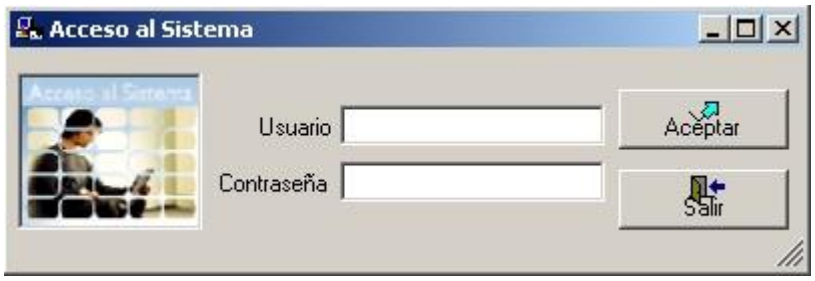


Figura 11.1 Pantalla de Acceso al Sistema

11.2. Cambio de contraseña

La interfaz podrá utilizarse cuando el usuario requiera modificar su clave de inicio de sesión al sistema. Es recomendable que el usuario actualice su contraseña periódicamente y en lo posible que no sean repetitivos para mayor seguridad.

Figura 11.2 Pantalla de Cambio de contraseña

- Una caja de texto llamada “Anterior Contraseña” en la cual se digitará la actual clave que utiliza el usuario
- Una caja de texto llamada “Nueva Contraseña” en la cual se digitará la nueva clave que manejará el usuario.
- Un caja de texto llamada “Confirme contraseña” para verificar que la nueva clave fue ingresada correctamente.

Respecto a los botones, realice las siguientes acciones:

- Presione el botón “Guardar” si acepta el cambio de contraseña.
- Presione el botón “Cancelar” si niega la acción de cambio de contraseña.

CAPITULO 12

12. AYUDA

12.1. Ayuda del Sistema Académico

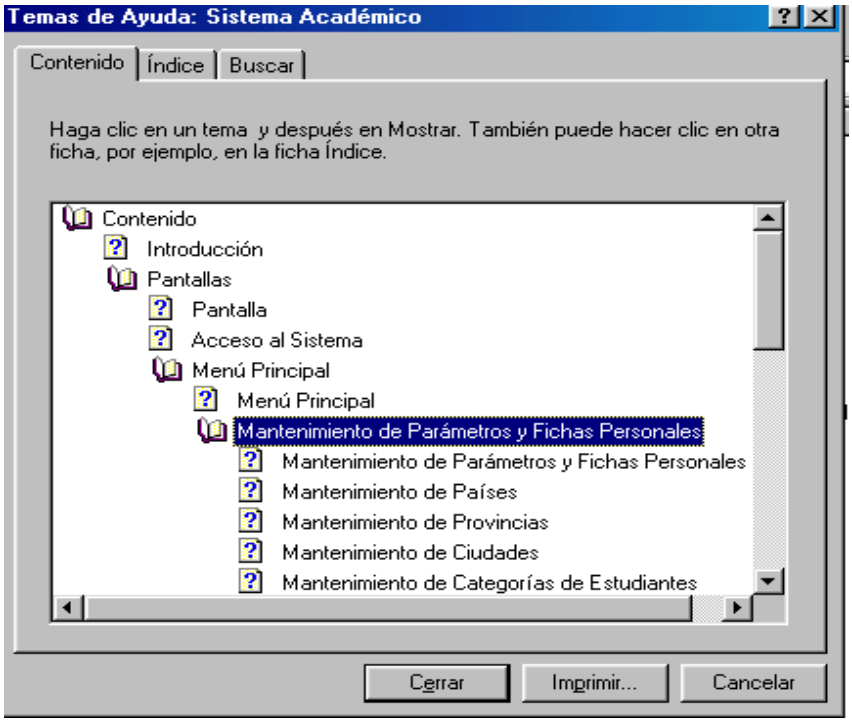


Figura 12.1. Acceso a la Ayuda del Sistema Académico

La ayuda del Sistema Académico tiene como objetivo principal proporcionar al usuario los conocimientos mas esenciales del Sistema Académico, como ingresos de datos o registros, subdivididos en tópicos y temas relacionados para un aprendizaje del usuario

12.2. Ejemplos de la Ayuda del Sistema Académico

Si el usuario tiene dudas sobre el Acceso al Sistema, en el Menú Principal puede buscar la palabra Ayuda , al dar clic en la palabra "Documentación" se abrirá la ventana "Ayuda" como la figura 12.1, luego dará clic o enter en el libro Contenido, luego en Pantallas después en Acceso al Sistema.

Se abrirá automáticamente una ventana que contendrá toda la información del tópico Acceso al Sistema como se muestra en la figura 12.2.

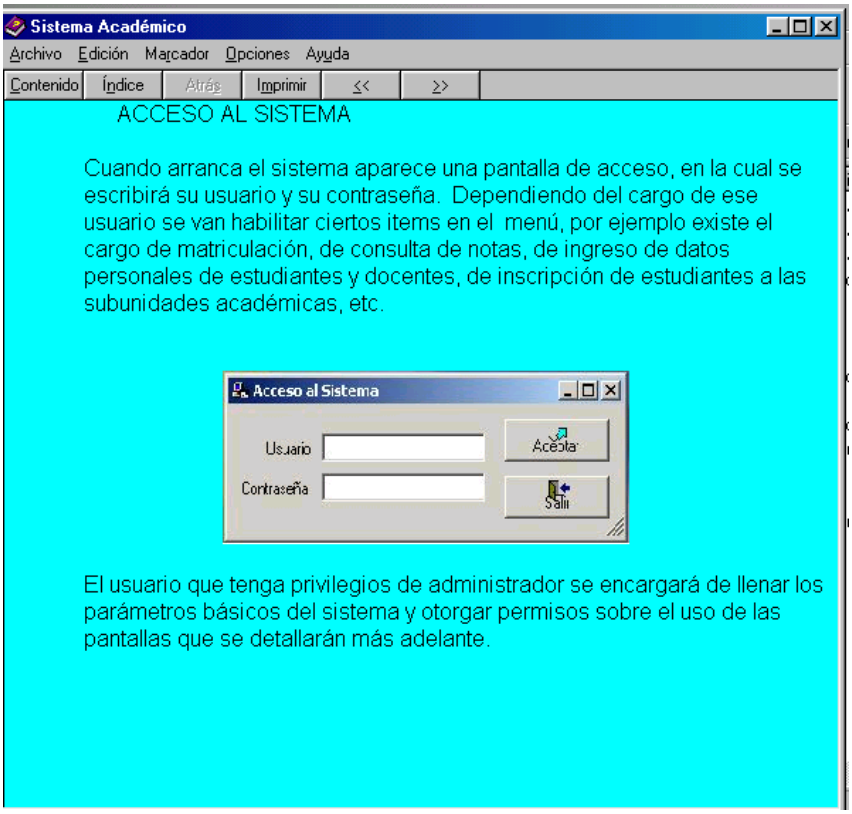


Figura 12.2. Ejemplo del uso de la Ayuda (1)

El usuario tiene la opción de navegar en la ayuda por medio de los botones Atrás o Adelante como puede regresar al Contenido de la Ayuda o Menú Principal de la misma.

El usuario tiene la opción de Imprimir o para una búsqueda mas rápida puede dar clic en el botón Índice, En esta ventana se presenta las palabras claves de toda la Ayuda del Sistema Académico, figura 12.2.

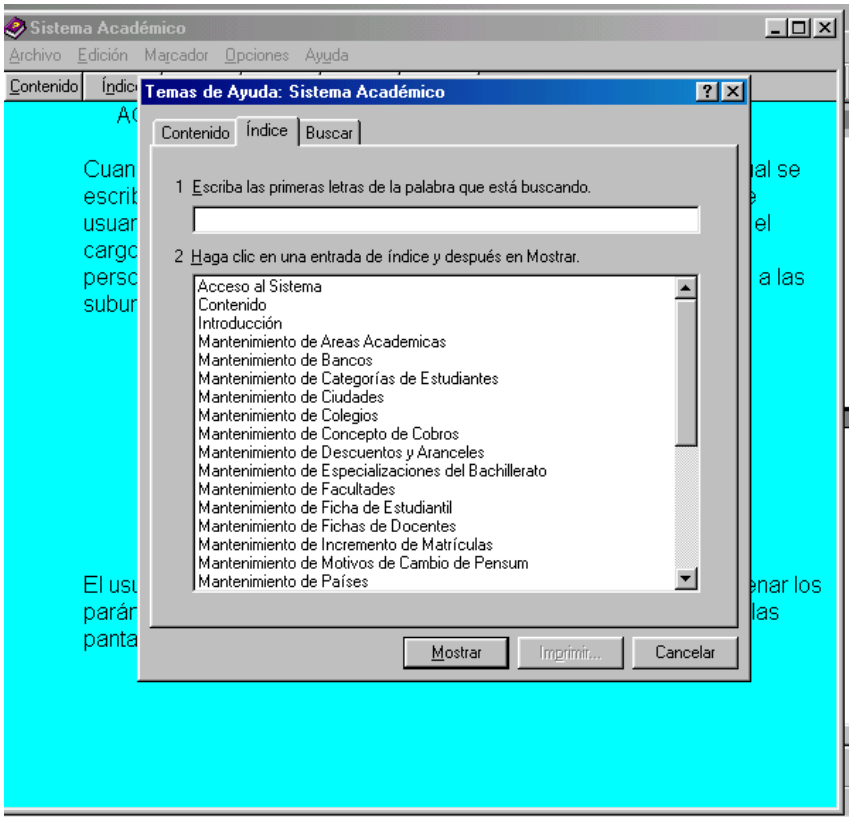


Figura 12.3. Ejemplo del uso de la Ayuda (2)

El usuario puede usar el índice de acuerdo a sus necesidades, al dar clic en Mostrar automáticamente se abrirá una ventana con la respectiva información, por ejemplo si da clic en Mantenimiento de Parámetros y Fichas Personales, se abrirá una ventana con la información requerida, tal como lo muestra la figura 12.3

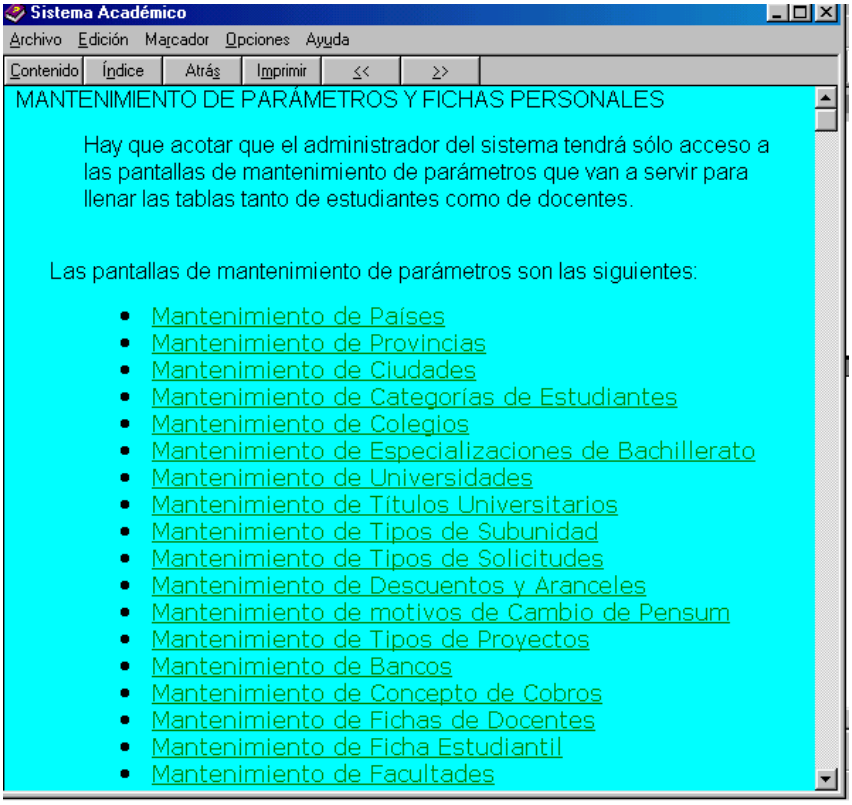


Figura 12.4. Ejemplo del uso de la Ayuda (3)

Así mismo, el usuario puede usar las palabras subrayadas (hipervínculos) para poder usar con mayor facilidad la ayuda.