

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

**Carrera de Ingeniería en Sistemas
Computacionales**

“Sistema Académico - Módulo Alumnos”

TESIS DE GRADO

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autor(es):

Daniel Almeida Palma

Stalin Chilán Quimis

Lubert Quiroz Pita

GUAYAQUIL – ECUADOR

Año: 2006

AGRADECIMIENTO

Daniel Almeida Palma

Dejo constancia de agradecimiento a Dios Todopoderoso, nuestro padre celestial, por darme día a día el don de la vida. A mis padres que durante toda mi existencia me han apoyado incondicionalmente, a mis hermanos que siempre han estado allí para ayudarme en todo momento, a mis tíos, a mis primos y demás familia que de alguna manera han colaborado con su granito de arena durante toda mi formación académica. Gracias a ellos, de todo corazón.

Stalin Chilán Quimis.

Agradezco a Dios por darme vida, salud, sabiduría e inteligencia para poder alcanzar mis metas y cumplir mi objetivo el cual es culminar mi carrera profesional, a mis padres, que me han ayudado y siempre han estado a mi lado, a mis hermanos que han estado conmigo en todo momento brindándome su ayuda para poder cumplir mis sueños y a mis tíos que me han servido como maestros desde mi etapa de colegio.

Lubert Quiroz Pita

Agradezco a Dios por darme vida y salud, a mis Abuelitos que aunque no estén cerca de mí, pero se que de donde se encuentren me bendicen, a mis padres por haberme dado mis estudios y por saber aconsejarme, a mis tíos por haberme apoyado.

DEDICATORIA

Daniel Almeida Palma

Este proyecto está dedicado a mis queridos padres el Sr. Daniel Almeida Freire y la Sra. Cecilia Palma de Almeida por apoyarme día a día a alcanzar la meta que me he propuesto y quienes a cambio de nada me lo han dado todo, a mi hermana Gissella que con ayuda incondicional siempre ha estado junto a mi lado en las buenas y en las malas, a mis tíos el Ing. Antonio Galarza y el Ing. Domingo Almeida que con sus sabios consejos me han servido de guía y de ejemplo, a mis primos, que de alguna u otra forma han aportado con su granito de arena durante toda mi formación académica.

Stalin Chilán Quimis

Dedico este proyecto a Dios por darme la sabiduría fortaleza y confianza, a mi padre Darío Antonio Chilán por ayudarme en todo momento y ser un ejemplo a seguir guiándome por el camino del bien, a mi madre Maria Quimis por todo su ayuda y cuidados en todo este largo proceso, a mis hermanos David y Mercy ya que ellos han sido la base fundamental y el pilar en los cuales siempre me apoyo, a mis sobrinitos Breiner David, Maria Jose y Maria Paula ya que ellos con su existencia me han dado muchas alegrías y son el futuro de la Patria.

Lubert Quiroz Pita

Dedico este trabajo a mis abuelitos, a mi padre el Sr. Luber Quiroz Franco que siempre ha estado a mi lado, a mi madre Juana Pita Rosales por querer siempre lo mejor de mí, a mi hermano por estar siempre a mi lado, a mis tíos por todos los consejos que me dieron, a todas aquellas personas que me conocen y en algún momento compartieron conmigo gran parte de mi vida estudiantil.

TRIBUNAL DE GRADUACION

Ing. Matilde Alvear
Directora

Ing. Abel Alarcón
Coordinador de Tesis

Ing. Gary Reyes
Vocal

Ing. Carlos Carranza
Vocal

Ab. Juan Chávez
Secretario

DECLARACION EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos, 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).

Daniel Almeida P.

Stalin Chilán Q.

Lubert Quiroz P.

RESUMEN

Como problema principal tenemos la falta de Información automatizada vía Web de los alumnos de la Carrera de Ingeniería en Sistemas Computacionales, es decir para acceder a los datos de los alumnos se debe acudir a las instalaciones de la carrera, como solución a la problemática antes mencionada ofreceremos tener toda la información de los Alumnos en el momento que lo necesite, es decir disponibles a cualquier hora y desde cualquier lugar mediante una conexión a Internet y un Navegador Web, la aplicación fue diseñada usando una herramienta llamada Jdeveloper la cual nos ayuda a desarrollar aplicaciones J2EE y a crear Paginas Web, Servlets y JAVA, y como Manejador de la Base de Datos utilizamos Postgress.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE GRADUACION	IV
DECLARACION EXPRESA	V
RESUMEN	VI
INDICE GENERAL	VII
INTRODUCCION	1
CAPITULO I	
IMPLEMENTACION PROYECTO SISTEMA ACADEMICO	
MODULO ALUMNO	
1.1. Problemática	2
1.2. Definición de la Problemática	3
1.3. Definición del Proyecto	4
1.4. Misión y Visión	6
1.4.1. Misión	6
1.4.2. Visión	6
1.5. Objetivos del proyecto	7
1.5.1. Objetivos Generales	7
1.5.1.1. Obtener información al instante	7
1.5.2. Objetivos Específicos	7

1.6. Alcance del proyecto	7
1.6.1. Crear base de datos de los Alumno.	8
1.6.2. Relacionar con los diferentes módulos existente en el sistema	9
1.6.3. Reporte y/o visualización de todas las consultas posibles	9
1.6.3.1. Consulta por código	9
1.6.3.2. Consulta por numero de cedula	10
1.6.3.3. Consulta por apellido	10
1.6.4. Ventana de sugerencias para mejoramiento	10
1.6.5. El modulo a su vez va ser adaptable a otra institución educativa	10
1.6.6. Además contara con un modulo de Actualización de Datos	11
1.7. Beneficios del proyecto	11
1.8. Cronograma	12
1.9. Recursos necesarios para el proyecto	13
1.9.1. Hardware	13
1.9.2. Software	14
1.9.2.1. IReport	15
1.9.2.1.1. Creación del Reporte	15
1.9.2.1.2. Secciones de un Reporte en iReport	15
1.9.2.1.3. Diseños del Reporte	17
1.9.2.2. Jdeveloper	18
1.9.2.2.1. Procedimiento a Seguir	18
1.9.2.3. Sistema Operativo	20

1.9.2.4. Postgres	20
1.9.3. Humano	22
1.10. Metodología del proyecto	22
1.10.3. Metodología del Análisis	22
1.10.1.1. Los productos entregables	23
1.10.1.2. Los no entregables	23
1.10.1.3. Tarea 1: Definir Problemática	23
1.10.1.4. Tarea 2: Solución de la Problemática	24
1.10.1.5. Tarea 3: Desarrollar prototipos	24
1.10.2. Metodología del diseño	24
1.10.2.1. Tarea 1.- Diseño de datos	25
1.10.2.2. Tarea 2.- Diseño Arquitectónico	25
1.10.2.3. Tarea 3.- Diseño de Interfaz	26
CAPITULO II	
ANALISIS DEL MODULO ALUMNO	
2.1. Necesidades de los usuarios	28
2.1.1. Reconocimiento del problema	28
2.1.2. Evaluación y Síntesis	28
2.1.3. Especificación	29
3.7. Evaluar que conceptos tiene el usuario del sistema para establecer su viabilidad.	29
2.2.1. Viabilidad económica.	30

2.2.2. Viabilidad Técnica.	30
2.2.3. Viabilidad Legal.	31
2.3. Análisis Técnico y económico.	31
2.4. Funciones de los recursos	31
2.4.1. El Hardware	32
2.4.2. El Software	32
2.4.3. Personal	32
2.4.4. Base de Datos	33
2.4.5. Documentación	33
2.4.6. Procedimientos	33
2.5. Restricciones de presupuestos y planificación temporal.	33
2.5.1. Restricciones de presupuestos.	33
2.6. Casos de Uso	34
2.6.1. Actores	34
2.6.2. Casos de uso	35
2.6.3. Inclusión	35
2.6.4. Extensión	36
2.6.5. Herencia	36
2.6.6. Communicates	36
2.7. Elaboración de los casos de uso del proyecto.	37
2.8. Diagrama de Casos de Usos	38

CAPITULO III

DISEÑO DEL MODULO ALUMNO

3.1. Especificaciones preliminares del diseño.	42
3.2. Diseño de variables.	42
3.3 Diseño de validaciones.	43
3.3.1. Validación de código	43
3.3.2. Validación del número de cédula	44
3.3.3. Validación de longitud de caracteres de las variables	44
3.3.4. Validación de tipo de datos de las variables	44
3.3.5. Validación del sexo	44
3.3.6. Validación de estado civil	45
3.4. Diseño de entidades.	45
3.4.1. Tipo de entidades.	46
3.5. Diseño de atributos.	47
3.5.1. Simples o compuestos	47
3.5.2. Monovaluados o multivaluados	47
3.5.3. Almacenados o derivados	48
3.5.4. Entidad alumno	48
3.5.5. Entidad persona	49
3.5.6. Entidad datos_laborales	49
3.6. Vínculo o relación de las entidades.	50

3.6.1. Correspondencia de cardinalidad.	51
3.6.1.1. Tipo de cardinalidad.	51
3.6.1.2. Tipos de participación de las entidades en una relación.	52
3.6.1.2.1. Opcional	52
3.6.1.2.2. Obligatoria	52
3.7. Diseño de tablas.	53
2.7.1. Para las entidades.	53
2.7.2. Para las relaciones.	55
3.8. Diseño de procedimientos.	56
3.9. Diseño de salidas.	62
3.9.1. Pantalla de Bienvenida al sistema de integración académico	62
3.9.2. Ingreso del usuario al sistema académico integrado	63
3.9.3. Pantalla de Bienvenida al módulo alumno	64
3.9.4. Ingreso de un Alumno	65
3.9.5. Consulta de un Alumno por Nombre	68
3.9.6. Consulta de un Alumno por Código	70
3.9.7. Consulta de un Alumno por Cedula	71
3.9.8. Eliminación de Alumno	74
CAPITULO IV	
DESARROLLO Y PRUEBA DEL MODULO ALUMNO	
4.1.- Desarrollo del Sistema	77
4.1.1. Creación de la base de datos	77

4.1.1.2 Clave Primaria	78
4.1.1.3. Clave Foránea	78
4.1.2. Descripción de la Base de Datos	79
4.1.2.1. Tabla Persona	79
4.1.2.2. Tabla Alumno	80
4.1.2.3. Tabla datospersonales	80
4.2. Objetivo.	81
4.3. Definición Estructura de Datos.	81
4.3.1. Tabla ALUMNO	81
4.3.2. Tabla PERSONA	82
4.3.3. Tabla DATOS LABORALES	82
4.3.4. Tabla ALUM_PERSONA	83
4.4. Diccionario de Datos.	84
4.4.1. Tabla persona	84
4.4.2. Tabla alumno	86
4.4.3. Tabla datoslaborales	87
4.5. Script de la Tablas de Nuestra base de Datos	88
4.5.1 Tabla persona.	88
4.5.2. Tabla alumno.	89
4.5.3. Tabla datos laborales.	89
4.6. Estándares de la Lógica del Negocio.	89
4.6.1. ALUMNOBO	91

4.6.2. ALUMNODAO	95
4.7 Seguridades	99
4.7.1. Seguridad de sistema operativo	100
4.7.2. Seguridad física	102
4.7.3. Seguridades de red	103
4.7.4. Seguridades del administrador	104
4.8 Etapa de pruebas	104
4.8.1. Prueba de especificaciones.	104
4.8.2. Prueba Unitaria	105
4.8.2.1. La prueba Ascendente	105
4.8.2.2. La prueba descendente	106
4.8.3. Prueba de Sistemas	106
4.8.4. Pruebas especiales de Sistemas	107
4.8.4.1. Prueba de Carga Pico	107
4.8.4.2. Prueba de almacenamiento	107
4.8.4.3. Prueba de desempeño en tiempo	107
4.8.4.4. Prueba de Recuperación	108
4.8.4.5. Prueba de procedimiento	108
4.8.5. Diseño de Datos de prueba	109
4.8.5.1. Utilización de datos reales de prueba	109
4.8.5.2. Utilización de datos artificiales de prueba	110
4.8.6. Bibliotecas de prueba	110

CAPITULO V**IMPLEMENTACION DEL SISTEMA**

5.1. Implementación de Sistemas	112
5.2. Objetivo Implementación de Sistemas	112
5.2.1. Seguir el cronograma de implantación	113
5.2.2. Determinar la capacitación	113
5.2.3. Implementar lo establecido en las necesidades	113
5.3. Elementos Físicos	114
5.4. Elemento Lógico	114
5.5. Elementos Humanos	115

CAPITULO VI**RECOMENDACIONES Y CONCLUSIONES**

6.1.Recomendaciones	116
6.1.1. Hardware	116
6.1.2. Software	116
6.1.3. Cableado	117
6.1.4. Puesta en marcha	117
6.2. Conclusión	119

INTRODUCCIÓN

Antes de hablar del Sistema Informático de la Carrera de Ingeniería en Sistemas Computacionales, específicamente del módulo de alumnos necesitamos saber que es un Sistema Informático (Software), un Software es un conjunto de programas escrito en cualquier lenguaje de programación que sirve para resolver mediante las computadoras los problemas de una aplicación determinada.

En la actualidad para muchas instituciones educativas los datos de los alumnos son de vital importancia y objeto de gran consideración para poder llevar un control respectivo de cada uno de ellos.

Esta aplicación provee información a los usuarios a través de un Web browser, haciendo uso de todos los beneficios que nos provee actualmente la gran red mundial de información Internet, además los informes y las salidas producidas por el sistema deben ser precisos, confiables y completos.

Para el desarrollo de este proyecto se ha dividido en capítulos (etapas) y en cada uno de ellos se hace un análisis, así como una orientación de los diferentes criterios aprendido durante nuestros estudios universitarios.

CAPITULO I

1. IMPLEMENTACION SISTEMA ACADEMICO

MODULO ALUMNO

1.1. Problemática

Como problema principal tenemos la falta de Información automatizada vía Web de los alumnos de la Carrera, es decir carecen de una base de datos en donde se pueda consultar todos los detalles acerca de ellos, por este modulo es de mucha importancia para nuestro sistema informático debido a que es la parte inicial a la cual vamos a estar accedando en el día a día.

Debemos considerar que este modulo va a estar ligado a cada uno de los demás módulos, es por eso que es de suma importancia que su contenido debe ser manejado con mucha precaución.

1.2. Solución de la Problemática

Como solución a la problemática antes mencionada ofreceremos tener toda la información de los Alumnos de la Carrera de Ingeniería de Sistemas Computacionales vía Web (información virtual) en el momento que lo necesite, es decir disponibles a cualquier hora y desde cualquier lugar mediante una conexión a Internet y un Navegador Web, para así poder realizar consultas de toda índole, a mas de consultas el estudiante tendrá la opción de realizar modificaciones y eliminaciones en el momento que este lo necesite.

En dicho modulo también podemos realizar mantenimiento de información adicional que va a ser de suma importancia para el alumno como son las Provincias, ciudades, parroquias, colegios, y especializaciones, para cada uno de ellos tendremos la opción de realizar ingresos, consultas, modificaciones y eliminaciones.

Adicionalmente el alumno podrá revisar exámenes que hayan sido tomados anteriormente en cada una de las materias debido a que crearemos una opción para subir dichos exámenes a la red para que el alumno pueda acceder en el momento que lo necesite.

Al igual que en los exámenes el alumno podrá revisar Syllabus (Ayudas) de cada una de las materias que existen en la carrera debido a que crearemos una opción para subir dichos Syllabus a la red para que el alumno pueda acceder en el momento que lo necesite.

1.3. Definición del Proyecto

Nuestro proyecto tiene como nombre Sistema Académico de la Carrera de Ingeniería en Sistemas Computacionales, y tenemos como propósito principal realizar el módulo alumno, este módulo es el que se encarga de capturar los datos de los alumnos de la carrera, dicho módulo se encuentra relacionado con los demás módulos del sistema.

En la (fig. 1-1) mostraremos la relación que existe entre los Módulos de Sistema Académico con el Módulo de Alumnos.

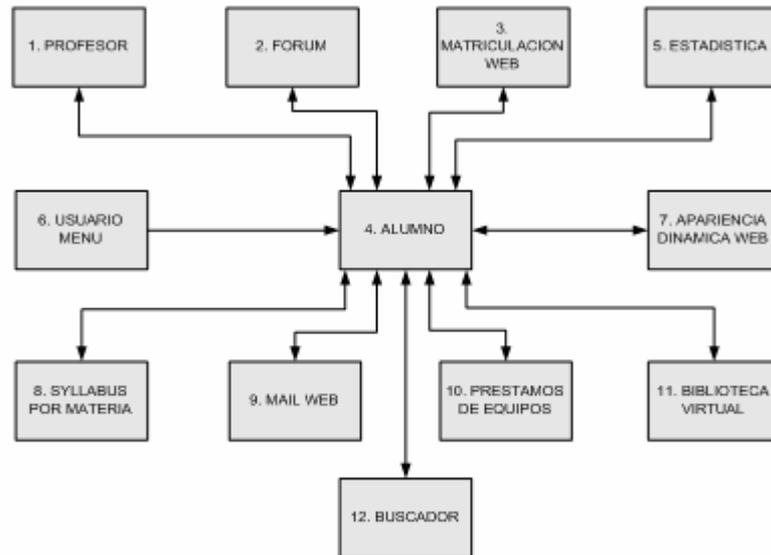


Fig. 1-1. Relación de todos los módulos del Sistema Académico con el Módulo de Alumnos.

Para hacer uso del módulo alumno necesitamos al usuario que va hacer uso del sistema en este caso un alumno, una vez que haya ingresado al sistemas tendrá la opción de interactuar con el módulo alumno, dicho módulo le ayudará al alumno a realizar ingresos, consultas, modificaciones y eliminaciones de cada usuario que ingrese al sistema, al realizar este proceso los datos ingresado viajaran hacia la base de datos para poder ser recuperadas en el momento que se los necesite, tal como se muestra en la (fig. 1-2).

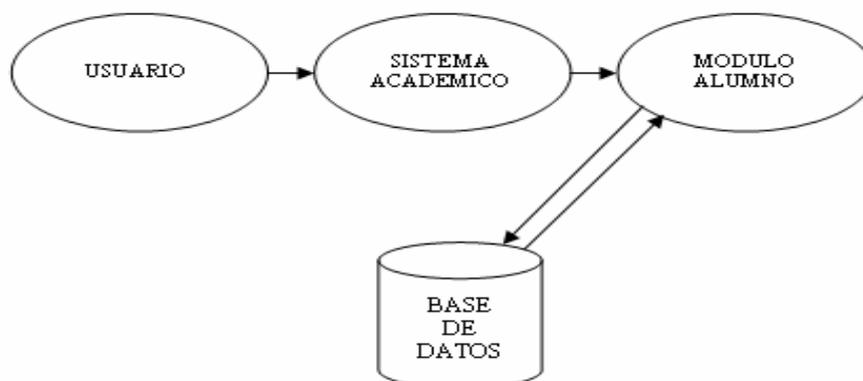


Fig. 1-2. Interacción del Usuario con el Sistema Académico.

1.4. Misión y Visión

1.4.1. Misión

Nuestra Misión es participar en el desarrollo de tecnología de punta cumpliendo con estándares y con excelencia de nuestros servicios, realizando capacitación, estudios, proyectos de acuerdo a la realidad y necesidades de los usuarios, para obtener satisfacción permanente, y así contribuir al desarrollo de la sociedad de la que formamos parte.

1.4.2. Visión

Nuestro reto es hacer que todas nuestras ideas y conocimientos en tecnologías sean aplicadas en proyectos futuros y que traspase las fronteras a nivel nacional, para que puedan ser medidos en ahorro de tiempo y dinero de acuerdo a la sociedad a la cual pertenecemos.

1.5. Objetivos del proyecto

1.5.1. Objetivos Generales.

1.5.1.1. Obtener información al instante.

El objetivo primordial es tener toda la información de los Alumnos de la Carrera de Ingeniería de Sistemas Computacionales, para así poder realizar consultas de toda índole, además de modificar y eliminar los datos de cada uno de los alumnos.

1.5.2. Objetivos Específicos

- Podemos subir información acerca de los exámenes tomados para que así cada uno de los alumnos puedan comprobar si existe o no algún error.
- También tendremos información de todas las notas de los alumnos por cada periodo educativo.
- Contara con una guía de profesores y de las materias que ellos dictaran en el transcurso del periodo educativo.

1.6. Alcance del proyecto

El modulo de Alumnos cubre una amplia gama de información necesaria para poder acceder a ella, dicha información es de mucha ayuda para

todos los demás módulos del sistema, debido a que para realizar ordenes de pago, matriculación, consulta de materias a cursar, necesitamos de esta información.

- Crear base de datos de alumnos o migrar Base si ya existe.
- Relacionar este módulo con los diferentes módulos ya existente en el sistema.
- Reporte y/o visualización de todas las consultas posibles.
- Ventana de sugerencias para mejoramiento
- El módulo a su vez va ser adaptable a otra institución educativa.
- Además contará con un módulo de Actualización de Datos.

Con la ayuda de esta base podemos dar respuesta a cualquier inquietud planteada de acuerdo a los alumnos de la Carrera.

1.6.1. Crear base de datos de los Alumno.

Consiste en diseñar una colección de información organizada en donde se van guardar los datos de los alumnos para que puedan ser accedidos por cada uno de los usuarios por medio del sistema que vamos a desarrollar, en esta base de datos se va a reflejar toda la información de cada uno de los alumnos de la carrera, esta base va a

ser utilizado por cada uno de los módulos que van a formar parte del sistema académico de la carrera de ingeniería en sistemas.

1.6.2. Relacionar este modulo con los diferentes módulos ya existente en el sistema.

Para que el sistema académico sea completo se necesita hacer la integración de cada uno de los módulos, una vez integrado vamos a utilizar un servidor Linux el cual se va a encargar de subir nuestro sistema a la red para que pueda ser accesado por los usuarios finales.

1.6.3. Reporte y/o visualización de todas las consultas posibles.

Se va a realizar procedimientos que se van a encargar de mostrar los datos de los alumnos que han sido previamente ingresados, este tipo de consultas se lo hace de acuerdo a tres criterios los cuales se detallan a continuación:

1.6.3.1. Consulta por código.

Esta consulta sirve cuando sabemos el código del alumno a consultar, al ingresar el código nos devuelve toda la información del alumno que corresponde a dicho código.

1.6.3.2. Consulta por número de cédula.

Esta consulta sirve cuando sabemos el número de cédula del alumno a consultar, al ingresar el número de cédula nos devuelve toda la información que corresponde a dicha cedula.

1.6.3.3. Consulta por apellido.

Esta consulta sirve cuando necesitamos información de todos los alumnos con un apellido X, es decir debemos ingresar un apellido y obtendremos la información de cada uno de ellos.

1.6.4. Ventana de sugerencias para mejoramiento.

En esta parte el usuario podrá ingresar las sugerencias que tienen con respecto a nuestro módulo para así poder realizar mejoras en lo posterior.

1.6.5. El módulo a su vez va ser adaptable a otra institución educativa.

Debemos realizar una parametrización para así poder utilizarlo en cualquier unidad educativa, entre las parametrizaciones tenemos la nacionalidad, es decir si es de nacionalidad Ecuatoriana tendrá la opción de escoger la Provincia, Ciudad y Parroquia, si no es de nacionalidad Ecuatoriana debe escribir el país al cual pertenece.

1.6.6. Además contará con un módulo de Actualización de Datos.

Se va a realizar procedimientos que se van a encargar de Actualizar los datos de los alumnos que han sido previamente ingresados, este tipo de modificaciones se lo hace ingresando el código del alumno para así poder modificar dichos datos.

1.7. Beneficios del proyecto.

El beneficio principal que se obtendrá de la implementación de nuestro módulo es el acceso rápido y oportuno de la información de cada uno de los datos de los alumnos de la Carrera de Ingeniería en Sistemas Computacionales sin necesidad de visitar las instalaciones de la Institución, es decir para obtener dicha información simplemente lo puede hacer desde su hogar, trabajo o desde cualquier Cyber que tenga acceso al Internet sin importar la ubicación geográfica.

1.8. Cronograma

De acuerdo al cronograma realizado por nuestro grupo de trabajo en donde involucra el análisis, diseño e implementación del módulo será definida en la siguiente tabla:

Actividad	Día	Fecha Inicio	Fecha Fin
Propedéutico	1d	sáb 21/01/06	sáb 21/01/06
Problemática a resolver	2d	lun 23/01/06	mar 24/01/06
Objetivos del proyecto	1d	mié 25/01/06	mié 25/01/06
Alcance del proyecto	2d	jue 26/01/06	vie 27/01/06
Presentación	1d	sáb 28/01/06	sáb 28/01/06
Corrección de alcance	11d	dom	vie 10/02/06
Presentación de la Corrección	1d	sáb 11/02/06	sáb 11/02/06
Análisis del sistema	22d	lun 13/02/06	vie 10/03/06
Análisis de necesidades	5d	lun 13/02/06	vie 17/02/06
Desarrollo de presupuestos	5d	sáb 18/02/06	jue 23/02/06
Calcular fechas de entrega	2d	vie 24/02/06	lun 27/02/06
Analizar recursos	4d	mar 28/02/06	vie 03/03/06
Analizar metodologías	6d	sáb 04/03/06	vie 10/03/06

Actividad	Día	Fecha Inicio	Fecha Fin
Diseño del sistema	25d	lun 13/03/06	mar 11/04/06
Especificaciones preliminares	5d	lun 13/03/06	vie 17/03/06
Diseño de variables	2d	sáb 18/03/06	do 19/03/06
Diseño de validaciones	2d	lun 20/03/06	mar 21/03/06
Diseño de tablas	4d	mié 22/03/06	lun 27/03/06
Diseño de relaciones	4d	mar 28/03/06	vie 31/03/06
Diseño de módulos	4d	sáb 01/04/06	mié 05/04/06
Especificar funcionamiento	4d	jue 06/04/06	mar 11/04/06
Presentación de Análisis y Diseño	1d	lun 17/04/06	lun 17/04/06
Implementación			
Prototipos de Interfaces de Usuario		Vie 28/04/06	Vie 19/05/06
Modelo de Implementación		Lun 22/05/06	Mar 03/10/06
Modelo de Implementación		Lun 22/05/06	Mar 03/10/06
Integración con demás módulos		Mie 03/10/06	Dom 15/10/06

1.9. Recursos necesarios para el proyecto.

De acuerdo a la tecnología actual necesitaremos recursos importantes para la realización de nuestro sistema, entre ellos tenemos al Hardware, Software y el recurso humano, los cuales los detallamos a continuación:

1.9.1. Hardware

El Hardware, es un dispositivo electrónico físico que será usado para desarrollar el módulo a implementar, el cual siempre está disponible desde el lunes 24 de enero del 2006, hasta la culminación del proyecto.

Como mínimo 2 computadores personales con las siguientes características:

- Procesador Pentium IV de 3 GHz. De 64 bits.
- 512 MB de memoria RAM.
- Disco duro de 80 GB.
- Unidad de CD-Writer.
- Mainboard con Video / Red.
- Monitor, Teclado y Mouse.
- Regulador UPS de 650 w.
- Un servidor de Linux.

1.9.2. Software

El Software (Programas a usar), serán las herramientas intangibles realizadas en un lenguaje de programación, las cuales van a ser usadas para la implementación del módulo, las herramientas para el análisis y diseño están disponibles desde el Lunes 24 de Enero del

2006, hasta la culminación del proyecto, pero el software para la implementación va estar disponible desde el Domingo 30 de Abril del 2006, hasta la culminación del proyecto.

Las herramientas disponibles las detallaremos a continuación:

1.9.2.1. IReport.- Nos permite crear reportes y mostrarlos en archivos .pdf, además de exportar datos a Excel o mostrarlos en un Browser (navegador), entre otros.

Lo primero que se debe de hacer es crear una conexión entre iReport y una base de datos, la Versión 0.5.1 de iReport ya proporciona drivers JDBC para establecer conexiones con bases de datos como MySQL y Access.

Procedemos a configurar iReport para establecer la conexión con la base de datos. Si la conexión fue exitosa, se debe guardar dicha conexión.

1.9.2.1.1. Creación del Reporte

En iReport, se tiene la opción para trabajar por proyecto, el cual puede contener varios reportes o uno solo según lo amerite.

1.9.2.1.2. Secciones de un Reporte en iReport.

A continuación se explicará de manera breve, las secciones que componen a un reporte en iReport, véase en la (fig. 1-3).

	title	
	pageHeader	
	columnHeader	
	detail	
	columnFooter	
	pageFooter	
	lastPageFooter	
	summary	

Fig. 1-3. Secciones que componen el cuerpo para la realización de un reporte en iReport.

title.- El título de nuestro reporte debe escribirse en esta sección. Solo se mostrará en la primera página del reporte.

pageHeader.- Aparece en la parte superior de cada página. Puede contener información adicional del reporte, descripciones, etc.

columnHeader.- En esta sección se muestran los nombres de los campos que se van a presentar.

Detail.- En esta sección se despliegan los valores correspondientes a los nombres de los campos definidos en la sección anterior. Estos datos pueden obtenerse mediante consultas SQL a una base de datos por ejemplo.

columnFooter.- Puede presentar información de totales para algunos de los campos de la sección detail. Por ejemplo “Total de Empleados: 220”.

pageFooter.- Aparece en la parte inferior de cada página. Este parte puede presentar, la fecha, número de página del reporte.

summary.- Esta sección puede presentar totales de campos de la sección detail. Si se desea incluir algún gráfico en el reporte, debe hacerse en esta sección.

1.9.2.1.3. Diseño del Reporte.

Se muestran a continuación los botones principales para el diseño del reporte de la barra de herramientas y su salida, así como se muestra en la (fig. 1-4):



Fig. 1-4. Barra de herramientas y salida del Reporte.

1.9.2.2. Jdeveloper.- Nos permite el desarrollo de nuestro software bajo un ambiente Web.

Es una herramienta de desarrollo de software. Es uno de los entornos visuales existentes en el mercado que permiten desarrollar aplicaciones en Java. Este tipo de herramientas permite generar código de manera automática, con el consiguiente ahorro de tiempo y esfuerzo.

1.9.2.2.1. Procedimiento a Seguir

Se debe crear una carpeta de trabajo en nuestro equipo para guardar en ella todo lo que generemos durante el proyecto. En ella a su vez haremos cuatro subcarpetas no importa los nombres, pero sí conviene que estén en distintos directorios para tenerlo más organizado, luego se empieza a realizar la codificación, una vez desarrollada la codificación se procede a Compilar, para ello utilizar el comando Project => Make o Project => Rebuild seleccionando el workspace o el proyecto. Existen dos iconos que permiten hacer lo mismo. Con el comando "Make" se compila solamente aquellas clases fuente que han cambiado, y con el comando "Rebuild" se compilan todas, hayan cambiado o no, por lo que en general será más rápido compilar con "Make". Luego debemos Ejecutar la aplicación, para ello utilizamos el comando Run: la opción Run => Run, o bien el icono con forma de semáforo, estando posicionados en el proyecto en la ventana System - Navigator. En el Área de Mensajes de Texto nos muestra cómo lo ejecuta. Nota: si no hubiéramos compilado previamente la aplicación, entonces, lo haría antes de ejecutar, por lo que podríamos habernos ahorrado el paso previo. Si no vemos que se nos muestre el resultado de la ejecución, abrir la ventana de

mensajes (View/Log Window). Si nos fijamos tenemos la línea de comandos que nos ejecuta la aplicación y a continuación en la línea siguiente nos escribe el resultado de la operación.

1.9.2.3. Sistema Operativo.- Linux.

Es un sistema operativo de distribución libre y no cuesta nada el único costo que tiene son los dispositivos físicos de donde se va a obtener el software, mientras que Windows cuesta mucho dinero, es por esa razón nuestra decisión de utilizar Linux, en Linux tenemos una instalación flexible, código abierto lo que nos da cabida para realizar modificaciones de una manera fácil y rápida.

1.9.2.4. Postgres.- Base de datos que vamos a utilizar para guardar toda la información de nuestro proyecto.

Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee

muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Postgres ofrece una potencia sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema.

Postgres se encuentra en la categoría de las Bases de Datos identificadas como objeto-relacionales. Nótese que éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que Postgres fue pionera.

PostgreSQL soporta roles de bases de datos, los que simplifican el manejo de grandes cantidades de usuarios con esquemas complejos de privilegios superpuestos.

Parámetros de entrada/salida (IN/OUT): Las funciones ahora soportan parámetros de entrada (IN), salida (OUT) y entrada/salida

(INOUT), lo cual mejora sustancialmente el soporte de lógica compleja y aplicaciones J2EE y .NET.

1.9.3. Humano

Son los operadores de las herramientas (Software) a usar para desarrollar la implementación, y son los directos responsables y autores del mismo.

- Daniel Almeida Palma.
- Stalin Chilán Quimis.
- Lubert Quiroz Pita.

Están disponibles desde el lunes 24 de enero del 2006, hasta la culminación del proyecto.

1.10. Metodología del proyecto

1.10.1. Metodología del Análisis.

El objetivo de esta metodología es la definición de las tareas a realizar, y las técnicas a emplear durante la actividad de análisis. En esta metodología se clasifica en 2 tipos:

- Entregables
- No entregables o internos

1.10.1.1. Los productos entregables.- Son aquellos que se entregan oficialmente, es decir en nuestro caso del módulo de alumno, nuestro producto entregable va a ser el acceso directo a la ejecución del proyecto vía Web.

1.10.1.2. Los no entregables.- Son productos internos al desarrollo que no se entregan al usuario final como el código fuente que es el que contiene las instrucciones necesarias para que el proyecto funcione adecuadamente.

Para nuestro análisis recopilamos los requerimientos, buscamos las alternativas de solución, escogimos la alternativa más factible la cual consistía en realizar el ingreso, consulta, modificación y eliminación de los alumnos de la Carrera de Ingeniería en Sistemas computacionales.

1.10.1.3. Tarea 1: Definir Problemática.- Falta de Información automatizada vía Web de los alumnos de la Carrera, es decir carecen de una base de datos en donde se pueda consultar todos los detalles acerca de ellos.

1.10.1.4. Tarea 2: Solución de la Problemática

Como solución a la problemática antes mencionada ofreceremos tener toda la información de los Alumnos de la Carrera de Ingeniería de Sistemas Computacionales vía Web (información virtual) en el momento que lo necesite, es decir disponibles a cualquier hora y desde cualquier lugar mediante una conexión a Internet y un Navegador Web, para así poder realizar consultas de toda índole, a más de consultas el estudiante tendrá la opción de realizar modificaciones y eliminaciones en el momento que éste lo necesite.

1.10.1.5. Tarea 3: Desarrollar prototipos

El desarrollo de prototipos consiste en realizar pantallas las cuales van a ser usadas para que interactúe con el usuario, una vez que están desarrollados y aprobados los prototipos en base al análisis realizado se procede al siguiente paso el cual es el diseño.

1.10.2. Metodología del diseño.

La metodología del diseño del software se obtiene de acuerdo al modelo de análisis. Es partimos de la información obtenida en la fase de análisis,

donde la información puede representarse como un flujo continuo que sufre una serie de transformaciones conforme va de la entrada a la salida, los pasos a seguir para el análisis son los siguientes:

1.10.2.1. Tarea 1.- Diseño de datos

El análisis de la estructura de datos sobre la estructura del programa hace que el diseño de datos tenga una gran influencia en la calidad del software. Los datos bien diseñados pueden conducir a una mejor estructura de programa y a una modularidad efectiva, en nuestro módulo de alumno en esta etapa del diseño de datos realizamos todas las validaciones necesarias para que nuestro proyecto acepte datos reales y precisos en el momento de la ejecución.

1.10.2.2. Tarea 2.- Diseño Arquitectónico

El objetivo principal del diseño arquitectónico es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Mezcla la estructura de programas y la estructura de datos y define las relaciones que facilitan el flujo de los datos a lo largo del programa.

Para nuestro módulo alumno una vez que tenemos la información del diseño de datos empezamos a realizar el diagrama de eventos, luego las entidades, luego el diagrama entidad relación, realizamos nuestro

diagrama de flujo estructurado y creamos las tablas que se van a usar en nuestro sistema.

1.10.2.3. Tarea 3.- Diseño de Interfaz.- El diseño de interfaz refleja los prototipos de las ventanas, las cuales van a ser las que sirven para que el usuario interactúe con el módulo, aquí se especifica cada uno de los accesos en forma gráfica, en esta parte se especifica de principio a fin como interactuar con el sistema.

CAPITULO II

2. ANALISIS

El análisis es un método, procedimiento, conjunto o arreglo de elementos para realizar un objetivo, la función del análisis es desarrollar un producto o servicio el cual puede ser lucrativo o con fines benéficos.

Un [Análisis](#) de Sistema se lleva a cabo teniendo en cuenta los siguientes objetivos:

- Necesidades de los usuarios.
- Evaluar que conceptos tiene el usuario del sistema para establecer su viabilidad.
- Análisis Técnico y económico.
- Funciones de los recursos: Hardware, Software, personal, base de datos, y otros elementos del Sistema.
- Restricciones de presupuestos y planificación temporal.

2.2. Necesidades de los usuarios.

Es el primer paso del análisis del sistema, se identifican las metas globales, se analizan las perspectivas del usuario, sus necesidades y requerimientos, este análisis se lo dividen en tres partes:

- Reconocimiento del problema.
- Evaluación y Síntesis.
- Especificación.

2.7.3. Reconocimiento del problema.

Como problema principal tenemos la falta de Información automatizada vía Web de los alumnos de la Carrera, es decir carecen de una base de datos en donde se pueda consultar todos los detalles acerca de ellos, por eso este módulo es de mucha importancia para nuestro sistema informático, debido a que es la parte inicial a la cual vamos a estar accedando en el día a día.

2.7.4. Evaluación y síntesis.

Al evaluar el problema llegamos a la conclusión que debemos obtener toda la información de los Alumnos de la institución educativa a la cual

se le implantará este módulo, para así poder realizar consultas de cualquier índole.

2.7.5. Especificación.

El módulo contará con los siguientes puntos:

- El módulo nos permitirá Ingresar, consultar, actualizar y eliminar los datos de los alumnos.
- Información acerca de los exámenes tomados de las materias aprobadas.
- Poder tener consultas de temas de ayuda relacionados a las materias (Syllabus).
- Nos permitirá realizar reportes de los alumnos.
- Tener acceso a una información histórica los alumnos.
- El módulo será parametrizable con algunos de sus atributos.

3.7. Evaluar que conceptos tiene el usuario del sistema para establecer su viabilidad.

Según encuestas realizadas a los alumnos el desarrollo del módulo será viable, ya que cada usuario podrá tener acceso de manera fácil y rápida a consultas de materias aprobadas o que está cursando en la actualidad, además tener ayudas de temas de exámenes y otros contenidos referentes a la carrera.

La viabilidad y el análisis de riesgos están relacionados, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce, sin embargo se deben tomar en cuenta lo siguiente:

- Viabilidad económica.
- Viabilidad técnica.
- Viabilidad Legal.

2.2.1. Viabilidad económica.

Es una evaluación de los costos de desarrollo, comparados con los ingresos netos o beneficios obtenidos del producto o Sistema desarrollado.

Para la realización de este módulo no obtendremos ningún beneficio económico, debido a que es propiedad de la carrera y ella será la encargada de sacarle provecho a este sistema en el momento que lo creyera necesario.

2.2.2. Viabilidad Técnica.

Es un estudio de funciones y restricciones que puedan afectar la realización de un sistema.

Según estudios realizados no tenemos restricciones, porque contamos con hardware y software necesarios para comenzar con el desarrollo del módulo.

2.2.3. Viabilidad Legal.

Es determinar cualquier posibilidad de infracción, violación o responsabilidad legal en que se podría incurrir al desarrollar el Sistema.

En cuanto a Viabilidad legal no tendríamos ninguna clase de restricciones por el motivo que las herramientas que se usarán son libres de licencias, es decir de tecnología Open Source.

2.3. Análisis Técnico y económico.

En el Análisis Técnico, se evalúa los principios técnicos del Sistema y al mismo tiempo recoge información adicional sobre el rendimiento, fiabilidad, características de mantenimiento y productividad. Los resultados obtenidos del análisis técnico son la base para determinar si continuar o abandonar el proyecto.

2.4. Funciones de los recursos: Hardware, Software, [personal](#), [base de datos](#), y otros elementos del Sistema.

Cada recurso queda especificado mediante cuatro características:

- Descripción del Recurso.
- Informes de disponibilidad.
- Fecha cronológica en la que se requiere el recurso.
- Tiempo durante el que será aplicado el recurso.

2.4.1. El Hardware.- (El computador), es el dispositivo electrónico que será usado para desarrollar el módulo a implementar, el cual siempre está disponible desde el Lunes 24 de Enero del 2006, hasta la culminación del proyecto.

2.4.2. El Software.- (Programas a usar), serán las herramientas con las cuales se llevará a cabo la implementación del módulo, las herramientas para el análisis y diseño están disponibles desde el Lunes 24 de Enero del 2006, hasta la culminación del proyecto, pero el software para la implementación va estar disponible desde el Domingo 30 de Abril del 2006, hasta la culminación del proyecto.

2.4.3. Personal: Son los operadores de las herramientas (Software) a usar para desarrollar la implementación.

- Daniel Almeida Palma.
- Stalin Chilán Quimis.
- Lubert Quiroz Pita.

Están disponibles desde el Lunes 24 de Enero del 2006, hasta la culminación del proyecto

2.4.4. Base de Datos: Colección de información organizada a las que se accede por medio del Software, va estar disponible desde el Domingo 30 de Abril del 2006, hasta la culminación del proyecto.

2.4.5. Documentación: Manuales o formularios, que detallan el empleo y operación del Programa, se lo ira desarrollando en el transcurso del desarrollo del módulo y va estar disponible en su totalidad en la culminación del proyecto.

2.4.6. Procedimientos: Pasos que definen el uso específico de cada uno de los elementos del Sistema y las reglas de su manejo.

2.5. Restricciones de presupuestos y planificación temporal.

2.5.1. Restricciones de presupuestos.

En las restricciones de presupuesto por el momento no se refleja ningún inconveniente debido a que se van a usar herramientas Open Source y económicamente no vamos a tener muchos gastos, con

respecto al Hardware ya contamos con un computador de similares características que detallaremos mas adelante y con el Software se lo podrá adquirir de una forma fácil y de bajo costo.

2.6. Casos de Uso

Se emplean para visualizar el comportamiento del sistema, el diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema, ya que solo especifica como deben comportarse y no como están implementadas las partes que define. Un caso de uso especifica un requerimiento funcional, es decir indica que es lo que debe pasar al realizar una acción. Los casos de uso son una representación de cómo el usuario también llamado Actor interactúa con el sistema.

En el diagrama nos encontramos con diferentes figuras que pueden mantener diversas relaciones entre ellas:

2.6.1. Actores: Un actor es alguien o algo que interactúa con el sistema, es quien utiliza el sistema. Por la frase "interactúa con el sistema" se debe entender que el actor envía o recibe del sistema unos mensajes o intercambia información con el sistema. En pocas palabras, el actor lleva a cabo los casos de uso. Un actor puede ser una persona u otro sistema que se comunica con el sistema a modelar, se representan por un muñeco, véase en la (fig. 2-1).



Fig. 2-1. Representación de un Usuario.

2.6.2. Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad, véase en la (fig. 2-2).



Fig. 2-2. Representación de un Caso de uso.

Los casos de uso pueden tener relaciones con otros casos de uso. Sus relaciones son:

2.6.3. Inclusión: Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso, véase en la (fig. 2-3).



Fig. 2-3. Representación de una Inclusión.

2.6.4. Extensión: Una relación de un caso de Uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.

2.6.5. Herencia: Es la típica relación de herencia, así se muestra en la (Fig. 2-4).



Fig. 2-4. Representación de una Herencia.

2.6.6. Communicates: Comunica un actor con un caso de uso, o con otro actor.

Parte del sistema (System boundary): Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

2.7. Elaboración de los casos de uso del proyecto.

Usuario final: ingreso, modificación y eliminación de alumnos

Nombre:	Modulo de Alumno (Ingreso, consulta, modificación y eliminación)
Autor:	Abel Alarcón
Fecha:	05/10/2006
Descripción:	<p>Permite Ingresar a un nuevo alumno a la Base de Datos de la Carrera.</p> <p>Permite Modificar los alumnos de la Base de Datos de la Carrera.</p> <p>Permite Eliminar los alumnos de la Base de Datos de la Carrera.</p>
Actores:	Usuario logoneado a nuestro modulo
Precondiciones:	El usuario debe haberse logoneado en el sistema.
Flujo Normal:	<p>El actor pulsa sobre el menú de alumnos para que aparezca un sub menú y poder hacer el ingreso, consulta, modificación o eliminación del alumno.</p> <p>El sistema muestra cajas de texto para introducir los datos del alumno.</p> <p>El actor introduce los datos necesarios.</p> <p>El sistema comprueba si los datos son validos y los almacena.</p>
Flujo Alternativo:	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	Los datos han sido almacenados en el sistema.

Saltándome los campos evidentes como nombre, autor, fecha y descripción; los actores son aquellos que interactúan con el sistema. Las precondiciones son los hechos que se han de cumplir para que el flujo de evento se pueda llevar a cabo. Luego tenemos el flujo de eventos, que corresponde a la ejecución normal y exitosa del caso de uso. Los flujos alternativos son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados. Por último, las poscondiciones son los hechos que se ha de cumplir si el flujo de eventos normal se ha ejecutado correctamente.

De forma que un caso de uso es un documento como el anteriormente presentado. Los casos de uso se pueden detallar más o menos dependiendo de la necesidad del problema.

2.8. Diagrama de Casos de Usos

En los diagramas de casos de uso los muñecos son los actores y los eclipses son los documentos de casos de uso. Así que se dibuja un muñeco por actor y un eclipse por cada caso de uso y los enlazas con líneas cuando haya una relación entre ellos.

Con esto consigues una visión general de cómo los diferentes actores interactúan con los distintos casos de uso, véase en la (fig. 2-5)

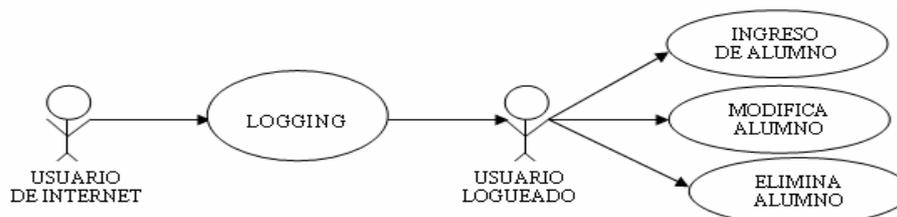


Fig. 2-5. Diagrama de los Casos de Usos de Ingresos, Modificación y Eliminación.

Usuario final: consulta de alumnos

Nombre:	Modulo de Alumno (consulta por nombre, por cedula, por código)
Autor:	Abel Alarcón
Fecha:	05/10/2006
Descripción:	<p>Permite Consultar un nuevo alumno a la Base de Datos de la Carrera de acuerdo a 3 opciones:</p> <ul style="list-style-type: none"> Por nombre. Por cedula. Por código.
Actores:	Usuario logoneado a nuestro modulo
Precondiciones:	El usuario debe haberse logoneado en el sistema.
Flujo Normal:	<p>El actor pulsa sobre el menú de alumnos para que aparezca un sub menú y poder hacer la consulta por nombre, cedula, y código.</p> <p>El sistema muestra cajas de texto para introducir los datos del alumno a buscar.</p> <p>El actor introduce los datos necesarios.</p> <p>El sistema comprueba si los datos son validos y los almacena.</p>
Flujo Alternativo:	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	Los datos han sido consultados en el sistema.

Diagrama de Casos de Usos

El diagrama de casos de usos para consultas se explica en la (fig. 2-6)

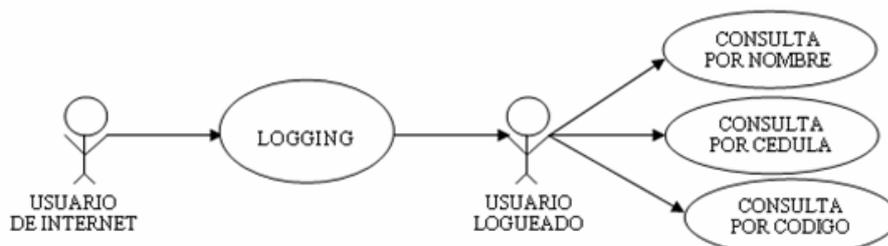


Fig. 2-6. Diagrama de los Casos de Usos para Consultas de un Alumno.

Usuario administrador: ingresa, consulta, actualiza y elimina provincias, ciudades, parroquias, colegios, especialización.

Nombre:	Modulo de Alumno (Ingreso, consulta, modifica y elimina Provincias, Ciudades, Parroquias, Colegio y especialización)
Autor:	admin
Fecha:	05/10/2006
Descripción:	Permite Ingresar, consultar, actualizar y eliminar datos como: Provincias. Ciudades. Parroquias. Colegios. Especialización.
Actores:	Administrador logoneado a nuestro modulo.
Precondiciones:	Debe ser usuario administrador validado con su respectiva contraseña.
Flujo Normal:	El actor pulsa sobre el menú de alumnos para que aparezca un sub menú y poder hacer el ingreso, consulta, modificación o eliminación de provincias, ciudades, parroquias, colegio y especialidad. El sistema muestra cajas de texto para introducir los datos necesarios. El actor introduce los datos necesarios. El sistema comprueba si los datos son validos y los almacena.
Flujo Alternativo:	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que los corrija.
Poscondiciones:	El proceso ha sido realizado con éxito.

Diagrama de Casos de Usos

El diagrama de casos de usos de forma general se explica en la (fig. 2-7)

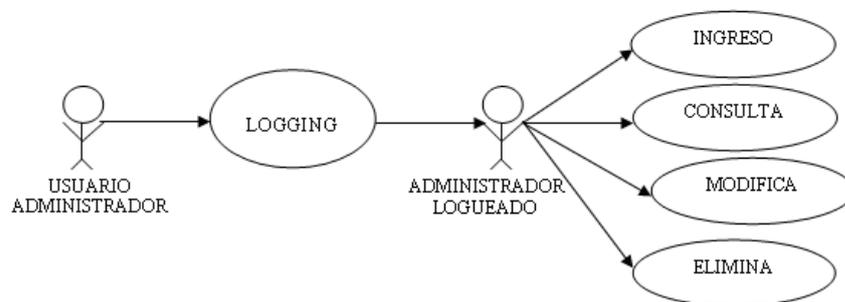


Fig. 2-7. Diagrama de los Casos de Usos de manera general .

CAPITULO III

3. DISEÑO

3.7. Especificaciones preliminares del diseño.

El Diseño de Sistemas define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un modelo a seguir para la futura implementación del módulo.

En el diseño partimos de la representación del flujo de la información obtenida en la fase de análisis, donde la información puede representarse como un flujo continuo.

3.7. Diseño de variables.

En este punto se definen todas las variables a usar por nuestro módulo, es decir el punto de partida del diseño, se base en el análisis previamente realizado.

El impacto de la estructura de datos sobre la estructura del programa hace que el diseño de datos tenga una gran influencia en la calidad del software. Los datos bien diseñados pueden conducir a una mejor estructura de programa.

Entre las variables a usar en el módulo alumno tenemos las siguientes:

- codigo
- apellido
- pasaporte
- telefono1
- e_mail
- sexo
- nombre trabajo
- num_afiliacion
- provincia
- nombre padre
- nombre conyuge
- cedula
- direccion
- telefono
- celular
- estado_civil
- colegio
- direcciontrabajo
- cargo
- ciudad
- nombre madre
- numero afiliacion
- nombre
- ced_militar
- fax
- fec_nacimiento
- nacionalidad
- especialización
- telefono
- forma contacto
- parroquia
- direccion

3.7. Diseño de validaciones.

3.3.1. Validación de código.- El código es un identificador único debido a ello va a ser generado automáticamente, será de tipo int4 es decir entero de 4 dígitos, debido a ello no va haber coincidencia de códigos en ningún momento.

3.3.2. Validación del número de cédula.- Para nuestra implementación el número de cédula de identidad va a estar conformada por diez dígitos y para verificar si este es o no correcto se aplica un algoritmo llamado módulo 10, el cual tiene un dígito verificador que es el encargado de comprobar la veracidad del mismo, en caso que éste sea falso se procede a rechazar todo tipo de acceso al módulo.

3.3.3. Validación de longitud de caracteres de las variables.- Se implementará un procedimiento, el cual verifica que la variable tenga la longitud correspondiente de sus caracteres, (Ejemplo: El número de teléfono debe tener una longitud de 7 caracteres)

3.3.4. Validación de tipo de datos de las variables.- Para validar los datos se desarrollará un procedimiento, en donde los tipos de datos pueden ser numéricos, alfabéticos, alfanuméricos y fecha de acuerdo al caso que lo amerite, todo esto con la finalidad de que no exista conflicto entre tipo de variables.

3.3.5. Validación del sexo.- El campo sexo tiene una relación directa con cedula militar, ya que si el sexo es masculino se debe ingresar obligatoriamente el número de identificación militar, si el sexo es femenino el campo identificación militar quedará deshabilitado.

3.3.6. Validación del estado civil.- Para el estado civil tenemos un combo en donde tenemos cuatro opciones las cuales son:

- Soltero (a)
- Casado (a)
- Divorciado(a)
- Viudo(a)
- Unión Libre

Al escoger Casado(a) se debe ingresar obligatoriamente el nombre del cónyuge, en cualquiera de los otros casos no es obligatorio el ingreso de este dato.

3.7. Diseño de entidades.

“Es cualquier objeto del mundo real con existencia propia, sobre el cual queremos tener información en una base de datos”. Una entidad puede ser un objeto con existencia física o un objeto con existencia conceptual. Debemos elegir nombres que comuniquen, hasta donde sea posible, el significado de cada entidad.

Las entidades a usar en nuestro módulo son:

- persona

- alumno.
- datos_laborales.
- Colegio
- especializacion
- provincia
- ciudad
- parroquia
- materiaexamen
- sugerencias

3.4.1. Tipo de entidades.

Tenemos 2 tipos de entidades que son las siguientes:

- **Fuertes (o regulares)**, son aquellas que tienen existencia por sí mismas (Por ejemplo, alumno).
- **Débiles**, depende de otro tipo de entidad. (Por ejemplo, datos_laborales y persona dependen de alumno. La desaparición de un alumno de la base de datos hace que desaparezcan también todos los datos del mismo). Estas entidades normalmente no tienen suficientes atributos para formar una clave primaria.

En donde la entidad fuerte es alumno representado en un rectángulo simple, y las débiles son las restantes representada en un rectángulo doble. (fig. 3-1).

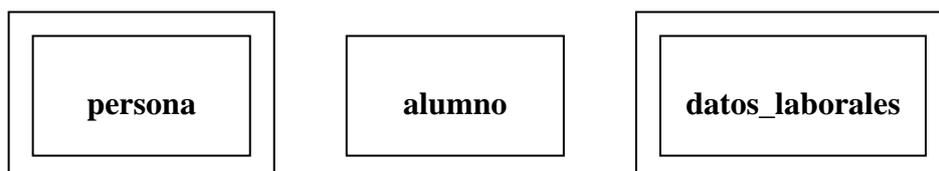


Fig. 3-1. Gráfico de las entidades datos_personales, alumno, datos_laborales

Cada una de estas entidades va estar compuesta por atributos que se detallaran mas adelante.

3.7. Diseño de atributos.

Cada entidad tiene propiedades específicas, llamadas **atributos**, que la describen. Los atributos se representan por elipses que están conectadas a su entidad o relación mediante una línea recta.

Los atributos pueden ser:

3.5.1. Simples o compuestos: Los compuestos están formados por un conjunto de atributos, mientras que los simples no se pueden dividir.

3.5.2. Monovaluados o multivaluados: Los monovaluados sólo pueden tener un valor para una entidad particular, mientras que los

multivaluados pueden tener más de un valor. Los multivaluados se representan mediante una elipse con trazado doble.

3.5.3. Almacenados o derivados: Los derivados son atributos cuyo valor para una entidad particular puede obtenerse en función de los valores almacenados en otros atributos. Se representan mediante una elipse con trazo discontinuo.

Los atributos que usaremos en las entidades descritas anteriormente, serán detallados en los gráficos posteriores.

3.5.4. Entidad alumno.- La entidad alumno es la principal del módulo a desarrollar en donde los atributos a usar son: codigoalumno, nombrepadre, nombremadre, codigospecializacion, codigocolegio, nombreconyuge, direccionmadre, estado, véase en la (fig. 3-2).

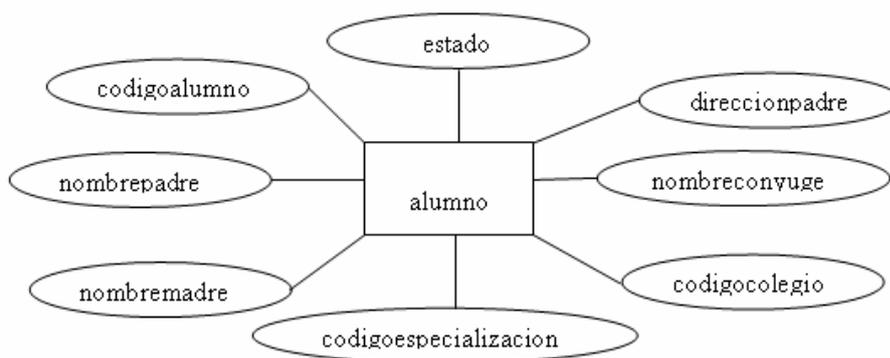


Fig. 3-2. Gráfico de la entidad alumno y sus respectivos atributos.

3.5.5. Entidad persona.- Esta entidad es de tipo débil por lo cual depende de la entidad alumno, y se relaciona por medio del atributo codigo.

Los atributos a usar en esta entidad son: codigoalumno (atributo derivado), apellidos, nombres, pasaporte, cedula, direccion, cedulamilitar, telefonocasa, celular, fechanacimiento, email, estadocivil, fax, nacionalidad, sexo, formacontacto, estado, véase en la (fig. 3-3).

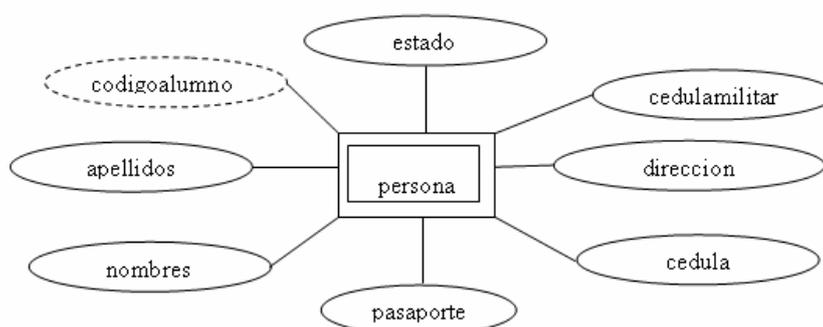


Fig. 3-3. Gráfico de la entidad datos_personales y sus respectivos atributos.

3.5.6. Entidad datos_laborales.- Esta entidad también es de tipo débil es decir depende de la entidad alumno, y se relaciona por medio del atributo codigo.

Los atributos a usar en esta entidad son: codigoalumno (atributo derivado), empresa, direcciontrabajo, telefonotrabajo, numeroafiliacion, cargo, estado. (fig. 3-4).

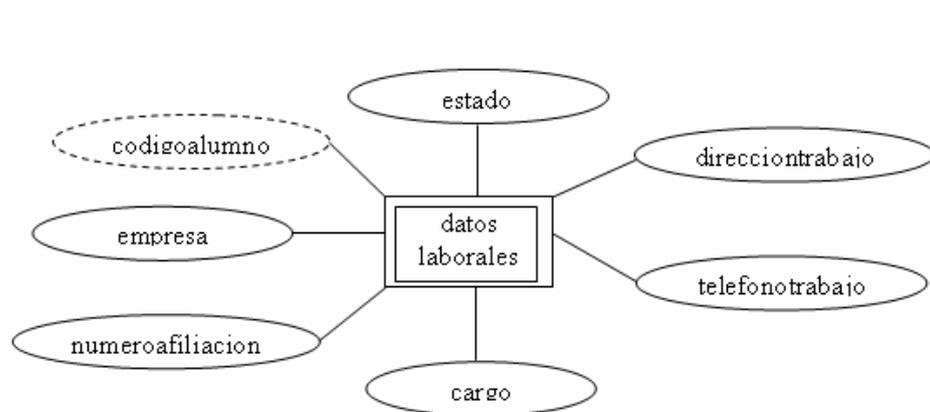


Fig. 3-4. Gráfico de la entidad datos_laborales y sus respectivos atributos.

3.7. Vínculo o relación de las entidades.

Se puede definir como una correspondencia o conexión entre dos o más entidades. En los diagramas se representa gráficamente como un rombo y sus nombres son verbos.

En la (fig. 3-5), nos muestra claramente la relación de las entidades personas y datos_laborales con la entidad alumno, donde describe que un alumno tiene datos personales y también puede tener datos laborales.

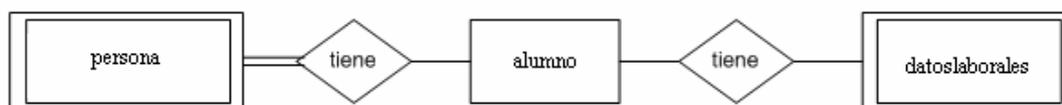


Fig. 3-5. Gráfico de las relaciones entre las entidades: datos_personales, alumno, datos_laborales

3.6.1. Correspondencia de cardinalidad.

Expresa el número máximo de entidades que están relacionadas con una única entidad del otro conjunto de entidades que interviene en la relación.

3.6.1.1. Tipo de cardinalidad.

Entre los tipos de cardinalidades tenemos las siguientes:

- **1:1 (Una a una):** La cardinalidad máxima en ambas direcciones es 1.
- **1:N (Una a muchas):** La cardinalidad máxima en una dirección es 1 y en la otra muchos.
- **N:M (Muchas a muchas):** La cardinalidad máxima en ambas direcciones es muchos.

En la (fig. 3-6), se describe la correspondencia de la cardinalidad entre las entidades a usar, en donde un alumno tiene un dato personal, y dicho alumno puede tener cero o más datos laborales.

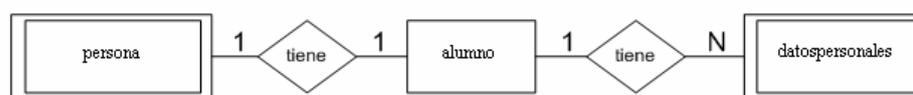


Fig. 3-6. Gráfico de la correspondencia de cardinalidad entre las entidades: datos_personales, alumno, datos_laborales

3.6.1.2. Tipos de participación de las entidades en una relación.

3.6.1.2.1. Opcional (parcial): No todas las ocurrencias de una entidad tienen que estar relacionadas con alguna de la otra entidad y se la representa mediante una línea con trazo sencillo. (Por ejemplo, no todo alumno posee datos laborales (fig. 3-7). En este caso ambas entidades participan parcialmente en la relación).

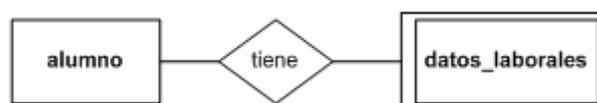


Fig. 3-7. Gráfico de la participación de cardinalidad entre las entidades: datos_laborales y alumno.

3.6.1.2.2. Obligatoria (total): Todas las ocurrencias de una entidad deben estar relacionadas con alguna de la entidad con la que esta relacionada. Se dice también, que existe una participación total de ese conjunto de entidades en el conjunto de relaciones, y se representa mediante una línea con trazo doble. (Por ejemplo, todo alumno tiene datos personales

(fig. 3-8). En este caso ambas entidades participan de forma total en la relación).

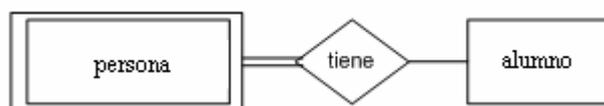


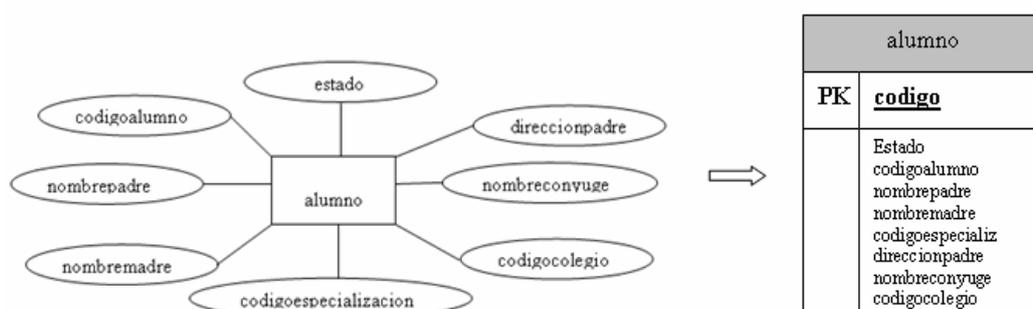
Fig. 3-8. Gráfico de la participación de cardinalidad entre las entidades: datos_personales y alumno.

3.7. Diseño de tablas.

Tanto el modelo Entidad-Relación (E-R) como el modelo de Base Datos relacional son representaciones abstractas y lógicas del desarrollo del mundo real. Ya que ambos modelos emplean principios de diseño similares, de tal forma que se puede convertir un diseño Entidad-Relación (E-R) en un diseño de base de datos relacional directamente, siguiendo una serie de normas que podemos resumir de la siguiente forma:

3.7.1. Para las entidades.

- En las **entidades fuertes** se genera una tabla con los atributos de una entidad. La clave primaria de la tabla es la misma que la de la entidad del modelo E-R.



FigFig. 3-9. Gráfico de la conversión de la entidad alumno del modelo E-R a una tabla de BD Relacional.

La (fig. 3-9), vemos como la entidad alumno se convierte en la tabla alumno y sus atributos se convierten en campos de la tabla alumno, es de esa manera como se empieza a formar nuestra base de datos.

- En las **entidades débiles**, se genera una tabla con los atributos de la entidad débil, más la clave primaria de la entidad fuerte.

En las (figs. 3-10 y 3-11), notamos que adicionalmente de la transformación de las entidades a tablas se heredan datos de las entidades fuertes como por ejemplo código, este campo va a ser el enlace entre la entidad fuerte y débil.

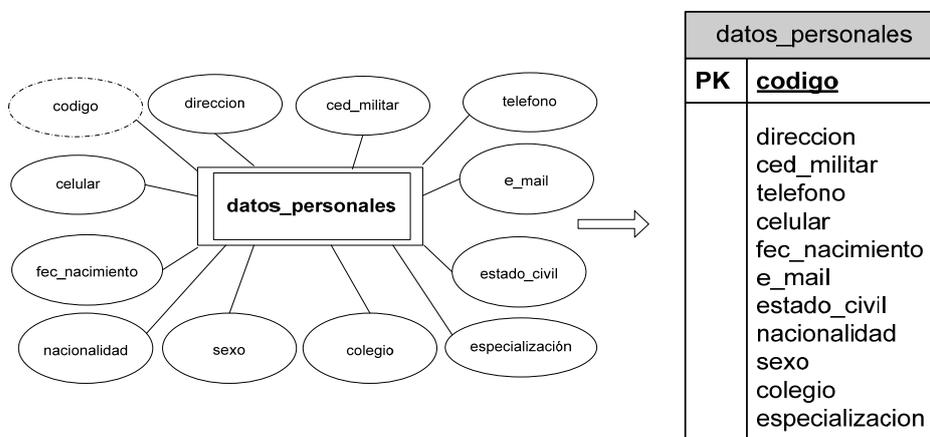


Fig. 3-10. Gráfico de la conversión de la entidad datos_personales del modelo E-R a una tabla de BD Relacional.



Fig. 3-11. Gráfico de la conversión de la entidad datos_laborales del modelo E-R a una tabla de BD Relacional.

3.7.2. Para las relaciones.

Si la relación es del **tipo 1:1** y es obligatorio (**total**) el tipo de participación de ambas entidades, solo es necesario crear una tabla con los atributos de las entidades que participan en la relación, y como clave primaria se puede tornar cualquiera de las claves de las entidades (fig. 3-12).

persona	
PK	<u>codigo</u>
	cedula nombre direccion ced_militar telefono celular fec_nacimiento e_mail estado_civil nacionalidad sexo colegio especializacion

Fig. 3-12. Tabla de la relación entre las entidades: alumnos y datos_personales.

Cuando la relación es del tipo **1:N**, y la entidad del lado N es de participación optativa (**parcial**) se necesitan tres tablas: una para representar cada entidad y una para representar la relación (fig. 3-13).

alum_labor	
PK	<u>codigo_a</u>
PK	<u>codigo_l</u>

Fig. 3-13. Tabla de la relación entre las entidades: alumnos y datos_laborales.

3.7. Diseño de procedimientos.

El diseño de procedimientos se realiza después de que se ha establecido la estructura de los datos, en este proceso se especifican los detalles de los procedimientos.

Los elementos a utilizar son:

- La secuencia.
- La condición.

- La repetición.

Estas tres construcciones son fundamentales en la programación estructurada.

En la (fig. 3-14), nos detalla claramente el diagrama de flujo de datos generalizado en donde debemos capturar los datos de entrada el cual va a ser el código del estudiante para así poder ingresa, consultar, modificar, y eliminar registros de los alumnos.

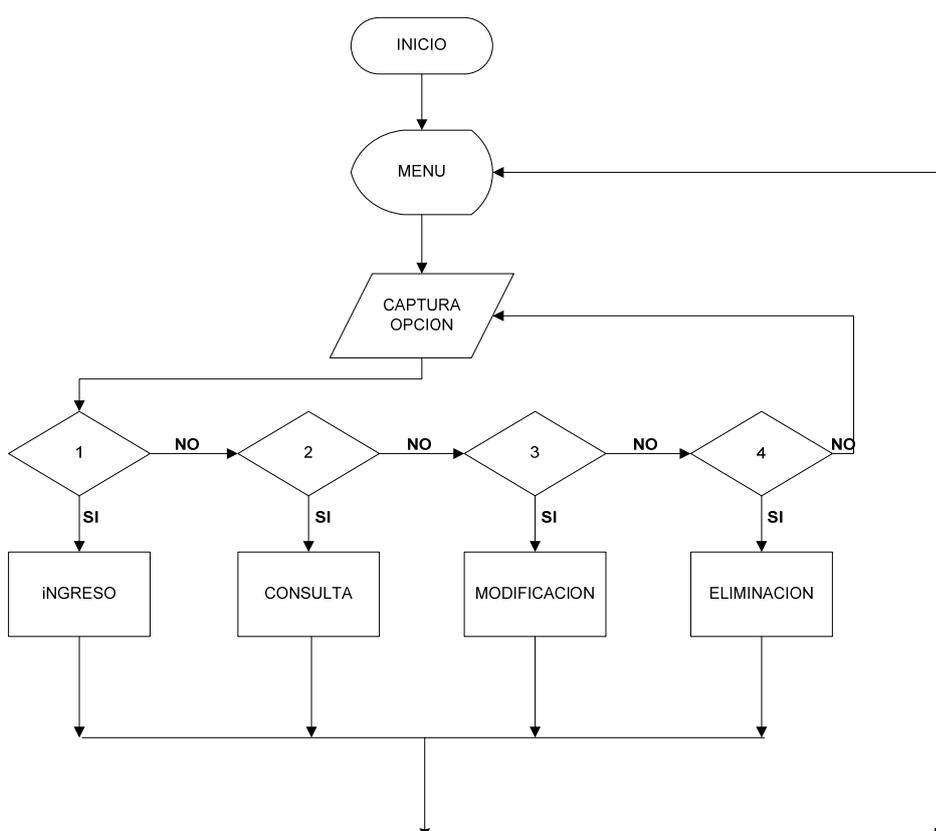


Fig. 3-14. Diagrama de Flujos de Datos Generales.

En la (fig. 3-15), vemos el diagrama de flujo de datos del procedimiento de ingresos en donde se genera automáticamente el código del nuevo alumno a ingresar, luego empezamos a ingresar los datos y una vez ingresado se los valida para ser almacenados en la base de datos.

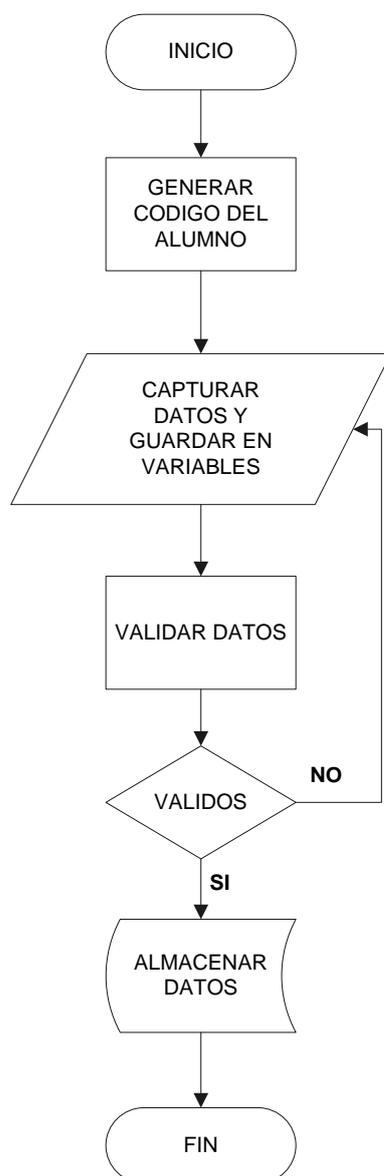


Fig. 3-15. Diagrama de Flujos de Datos del procedimiento de Ingreso.

En la (fig. 3-16), vemos el diagrama de flujo de datos del procedimiento de modificación, en donde ingresamos el código del alumno a modificar, una vez encontrado el alumno empezamos a modificar sus registros.

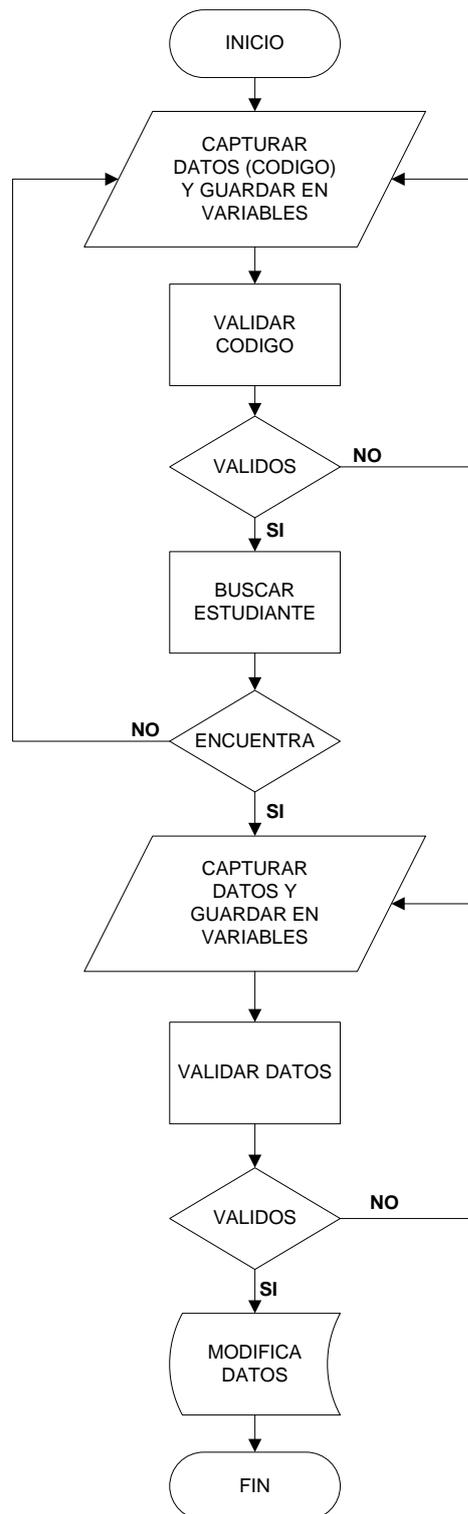


Fig. 3-16. Diagrama de Flujos de Datos del procedimiento de Modificación.

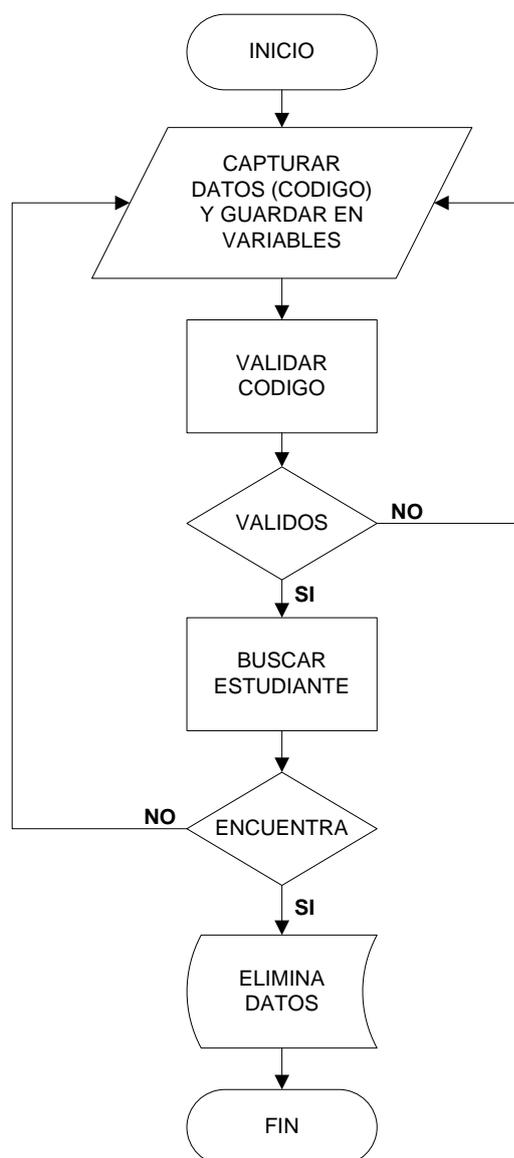


Fig. 3-17. Diagrama de Flujos de Datos del procedimiento de Eliminación.

En la (fig. 3-17), vemos el diagrama de flujo de datos del procedimiento de eliminación, en donde ingresamos el código del alumno a eliminar, una vez encontrado el alumno empezamos a eliminar sus registros.

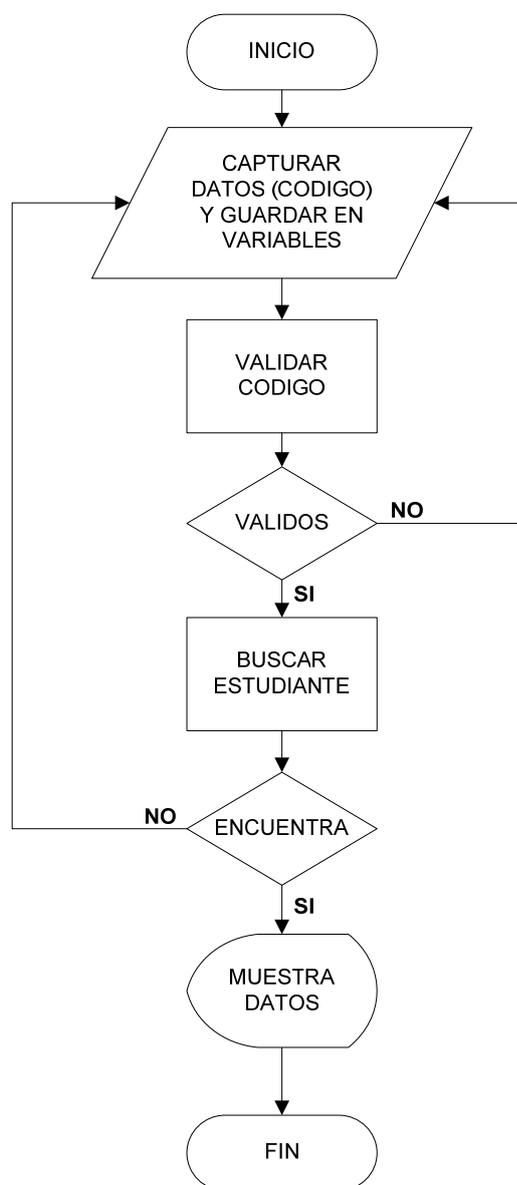


Fig. 3-18. Diagrama de Flujos de Datos del procedimiento de Consultas.

En la (fig. 3-18), vemos el diagrama de flujo de datos del procedimiento de consulta, en donde ingresamos el código del alumno a consultar, una vez encontrado el alumno empezamos a consultar todos sus registros.

3.7. Diseño de salidas.

El diseño de salidas refleja los prototipos de las ventanas, las cuales van a ser las que sirven para que el usuario interactúe con el módulo.

3.9.1. Pantalla de Bienvenida al sistema de integración académico.- Cuando el usuario ingresa al sistema por medio de un browser le aparece la pantalla de bienvenida en donde debe ingresar su clave y contraseña para poder acceder a dicho sistema, si no es un usuario registrado no podrá acceder a dicha pagina, a continuación mostraremos un ejemplo de la pagina en ejecución. (fig. 3-19).

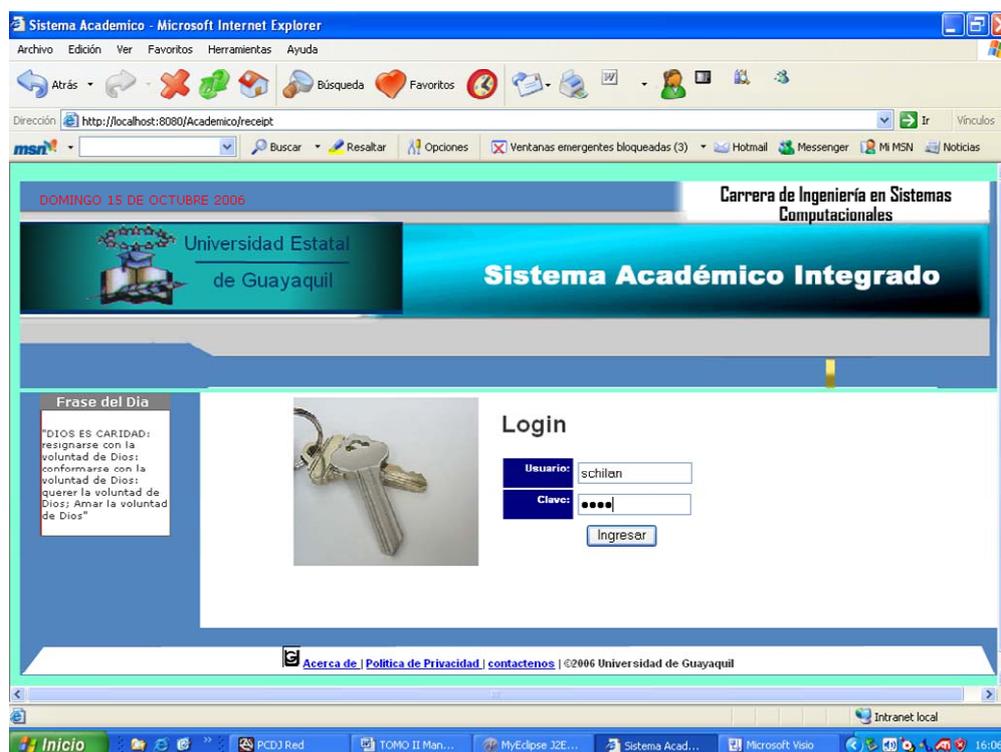


Fig. 3-19. Interfaz de Bienvenida al Sistema Integrado de la Carrera de Ingeniería en Sistemas Computacionales.

Existen 2 tipos de usuario el usuario normal y el usuario administrador, este último se va a encargar de darle el mantenimiento respectivo a la página.

3.9.2. Ingreso del usuario al sistema académico integrado.- Una vez ingresado el usuario aparece una página de configuración el cual nos permite modificar la apariencia Web de la página a la conveniencia de cada usuario, sobre el lado izquierdo se encontrará un menú para poder ingresar a cada uno de los módulos que conforman parte del sistema, véase en la (fig. 3-20).

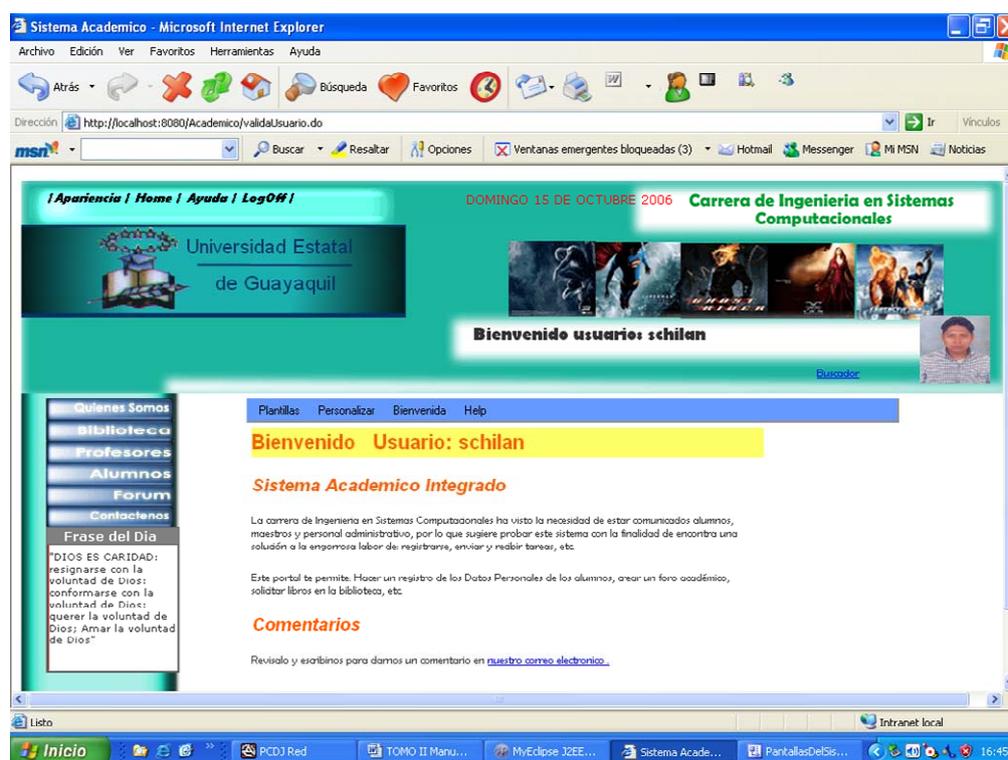


Fig. 3-20. Interfaz de Bienvenida y Menú Principal del Sistema Integrado.

3.9.3. Pantalla de Bienvenida.- Cuando el usuario ingresa al módulo le aparecerá una pantalla de bienvenida, en la parte superior de la misma tendremos un menú el cual nos ayudara a acceder a cada una de sus opciones.

En este menú se podrá realizar un ingreso de los alumnos al sistema, también una consulta por nombre, por código y por la cédula de los alumnos ya registrados, una modificación o actualización de sus datos, una eliminación del sistema de un alumno y una visualización de diversos reportes sobre lo que se quiera saber de los alumnos pertenecientes a la Carrera de Sistemas, a continuación mostraremos la página de bienvenida del modulo alumno, así como se muestra en la (fig. 3-21).

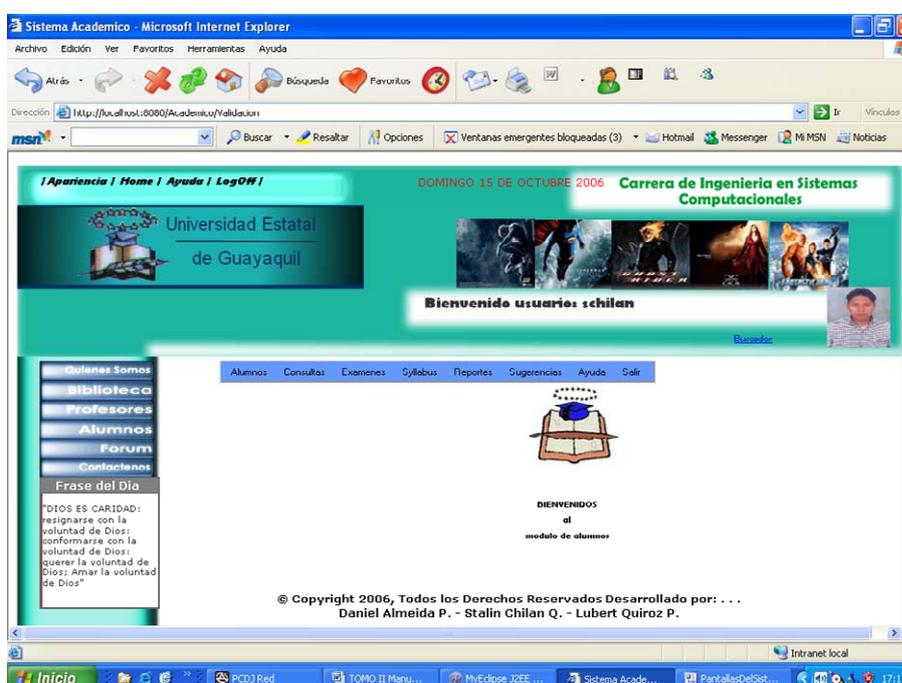


Fig. 3-21. Interfaz de Bienvenida al Módulo de Alumnos con su respectivo Menú Principal.

3.9.4. Ingreso de un Alumno.- Al escoger del menú la primera opción aparecerá una nueva pantalla en donde podrá ingresar un nuevo alumno al sistema, como se muestra en la (fig. 3-22).

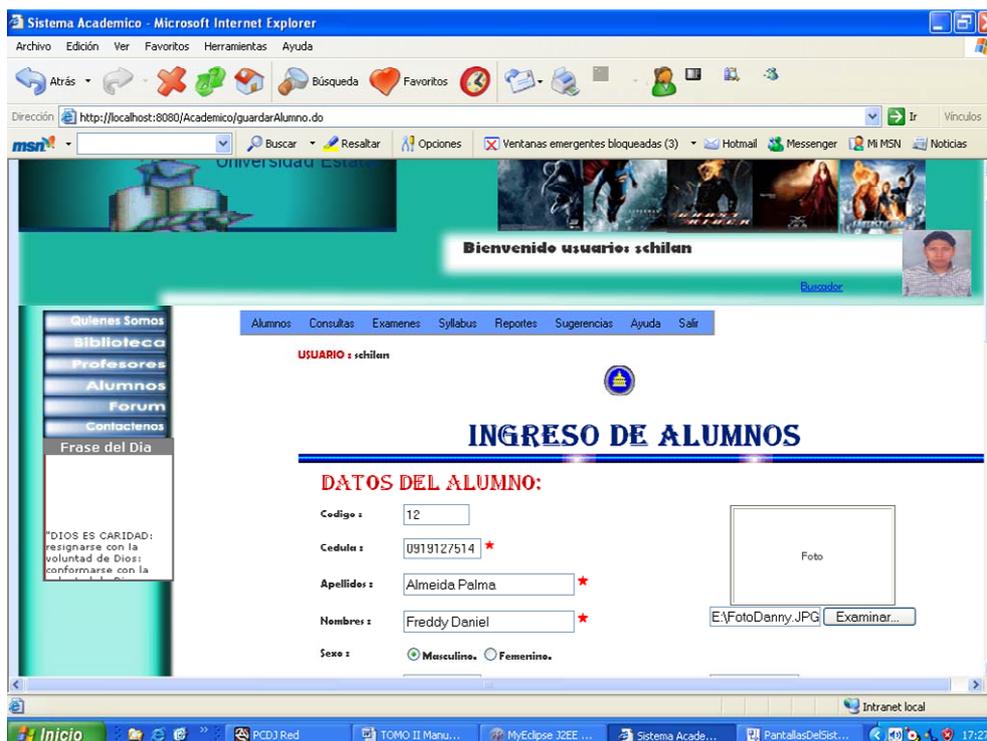


Fig. 3-22. Interfaz de Ingreso de Alumno.

Entre los campos validados tenemos el código de manera se genera automáticamente y no tendrá que ingresarlo, los campos con asteriscos indicarán los campos que necesariamente tendrán que llenarse, el campo de la cédula está validado de manera que cuando se ingrese éste tiene que ser un número válido porque de lo contrario le aparecerá un mensaje de error, luego ingresa los apellidos, los

nombres, el sexo, si es masculino podrá ingresar el numero de la cédula militar por que de lo contrario automáticamente se desactivará la caja de texto, el pasaporte, la dirección, teléfonos, celular, fax, la nacionalidad, el estado civil, el colegió donde se graduó, la forma de contacto que es con quien se lo pueda contactar, si el usuario se decide a guardar solamente esos datos ingresados aparecerá en la parte superior de la pagina un mensaje de todos los errores cometidos, ya sea porque no ingresó correctamente la cédula o porque no ingresó todos los campos obligatorios, tal como se muestra en la (fig. 3-23).

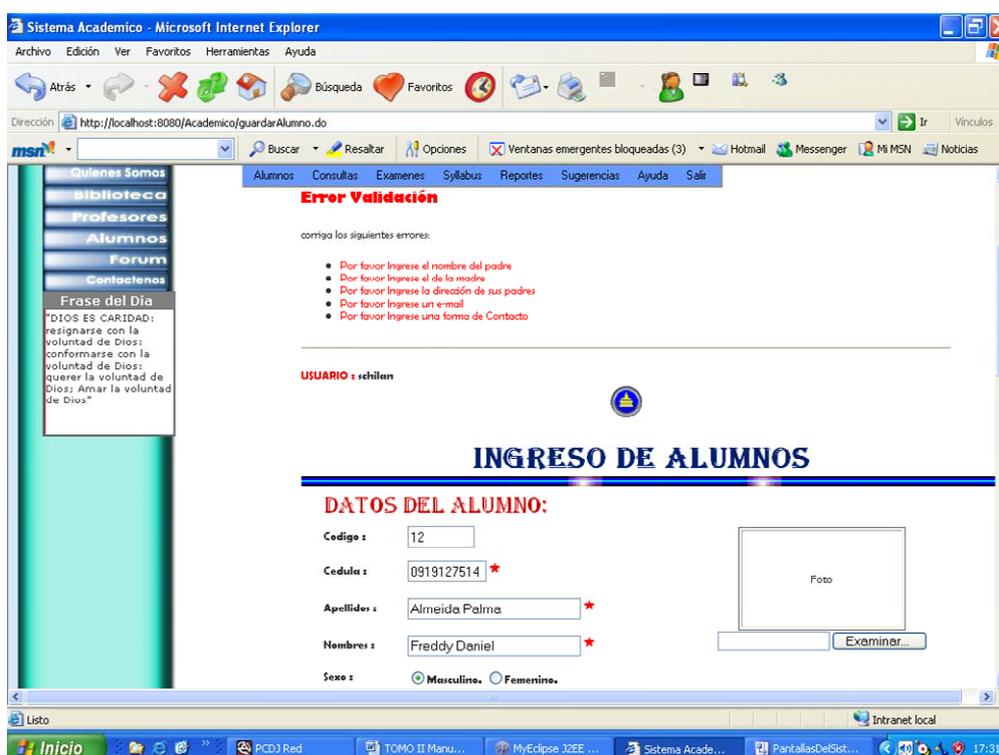


Fig. 3-23. Interfaz de Ingreso cuando no se han completado todos los campos obligatorios.

Se continúa ingresando la provincia, aquí cuando se seleccione una provincia automáticamente se llenará el combo de las ciudades de esa provincia que seleccionó, de igual manera pasa con las parroquias se llenará el combo cuando seleccione una ciudad específica, podrá seleccionar su fecha de nacimiento, tendrá que llenar los campos con el nombre de sus padres, la dirección donde residen, su e-mail, en caso de ser casado el nombre de su cónyuge, luego son opcionales los Datos Laborales aquí podrá ingresar el nombre de la empresa, la dirección, el teléfono, el número de afiliación al Seguro Social y el cargo en caso de que tenga algún trabajo; si lleno bien con los datos especificados anteriormente podrá seleccionar el botón Guardar como se muestra en la (fig. 3-24).

Sistema Academico - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Búsqueda Favoritos

Dirección <http://localhost:8080/Academico/guardarAlumno.do>

msn

Nombre del Padre : Daniel Almeida *

Nombre de la Madre : Cecilia Palma *

Dirección de los Padres : Saucés 0 *

E-mail : fdanielalmeida@hotmail.com *

Nombre del Cónyuge :

DATOS LABORALES:

Nombre de la Empresa : Coop. Grel. Vermeza

Dirección del Trabajo : Cuencia y Cacique Alvarez

Teléfono del Trabajo : 2400120

Número de Afiliación :

Cargo : Técnico

(*) Todos los campos con asteriscos son obligatorios

Guardar

Intranet local

Inicio PC02 Ped TOMO II M... MyClique J... Sistema Al... Pantallas... pgAdmin III 17:44

Fig. 3-24. Interfaz de Ingreso mostrando el botón de Guardar.

Luego de esto le aparecerá una nueva pantalla en donde le especifica que sus datos han sido guardados con éxito, tal como se muestra en la (fig. 3-25).

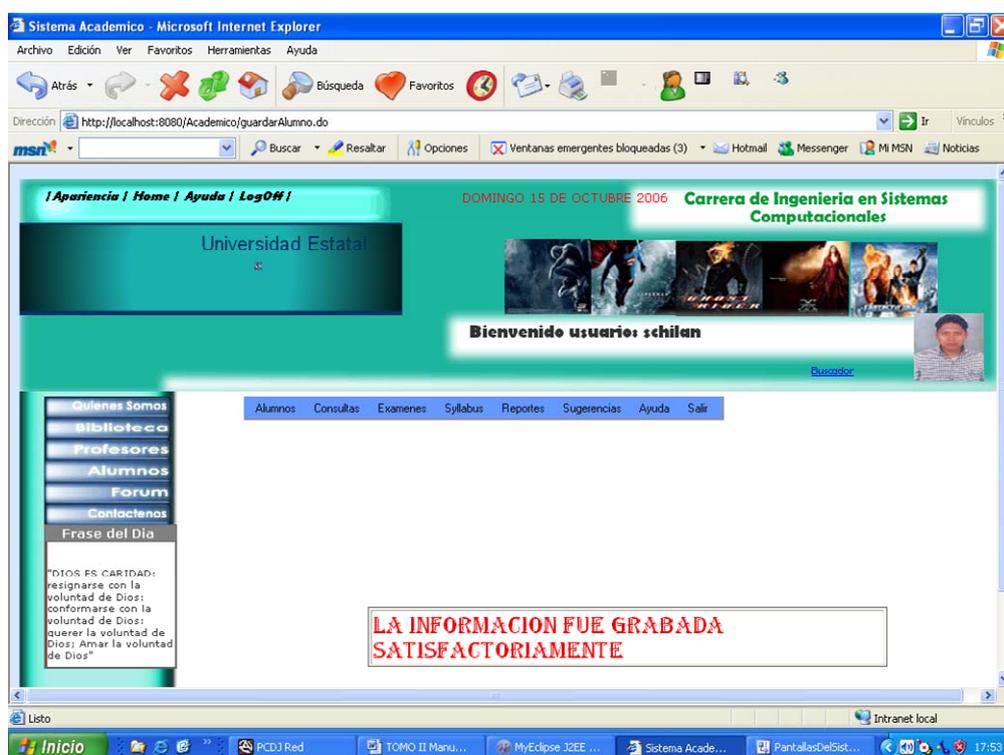


Fig. 3-25. Interfaz de Confirmación de datos guardados satisfactoriamente.

3.9.5. Consulta de un Alumno por Nombre.- Si seleccionamos del menú Consulta por nombre aparecerá una pantalla en donde podrá ingresar entre asteriscos un apellido completo o sólo una letra y al

presionar en el botón Buscar nos mostrará un listado de todos los apellidos encontrados en la base que tengan relación con la palabra ingresada, tal como se muestra en la (fig. 3-26).

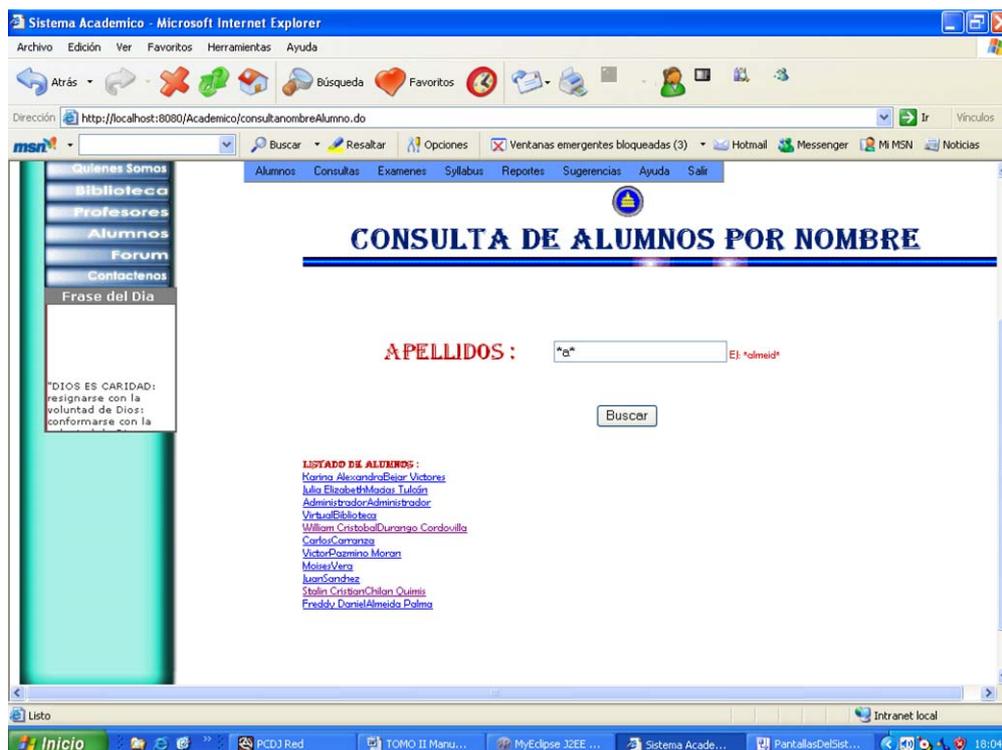


Fig. 3-26. Interfaz de Consulta por nombre de Alumno.

Luego para visualizar todos los datos referentes a un alumno determinado lo que se hace es un click sobre cualquier nombre de la lista encontrada, ya que cada uno de ellos se muestran de manera de link y ésta nos llevará a una nueva ventana presentándonos todos sus datos personales, tal como se muestra en la (fig. 3-27).

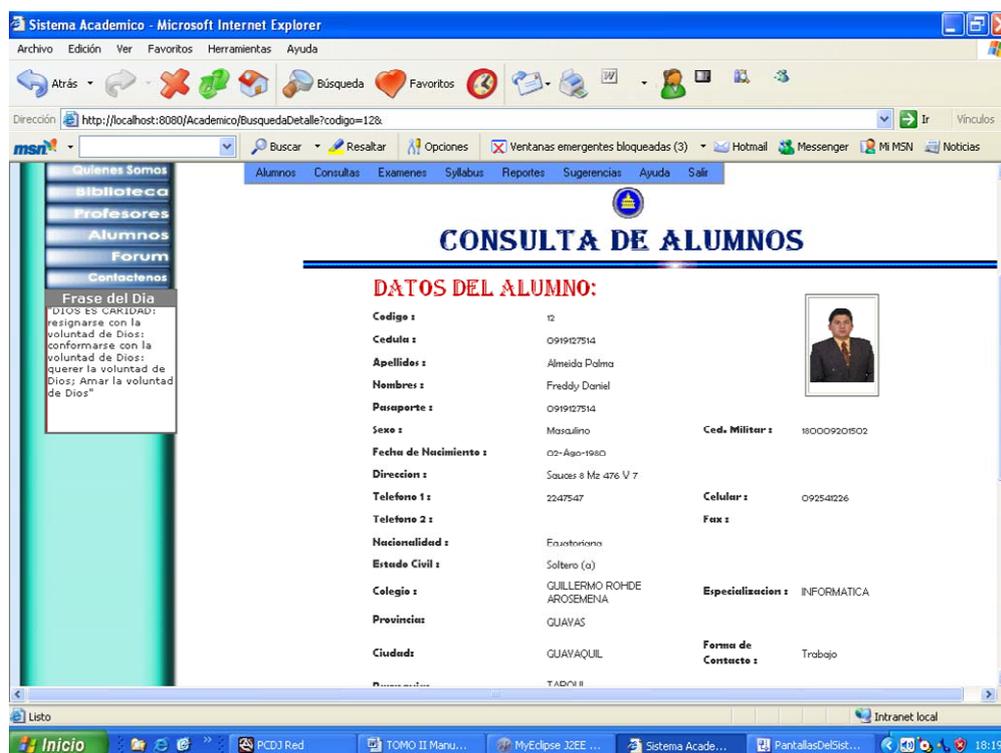


Fig. 3-27. Interfaz en donde se muestran los datos encontrados de un Alumno.

Una vez que se mostraron los datos podemos regresar a la página principal haciendo un click sobre el icono que se encuentra en la parte superior de la pantalla o sino puede realizar cualquier otra acción seleccionando alguno del menú principal.

3.9.6. Consulta de un Alumno por Código.- Si seleccionamos del menú Consulta por código aparecerá una pantalla en donde podrá ingresar el código correspondiente a cualquier alumno, si el código es válido y al presionar el botón de Buscar nos presentará todos los datos correspondientes a ese alumno tal como se muestra en la (fig. 3-28).

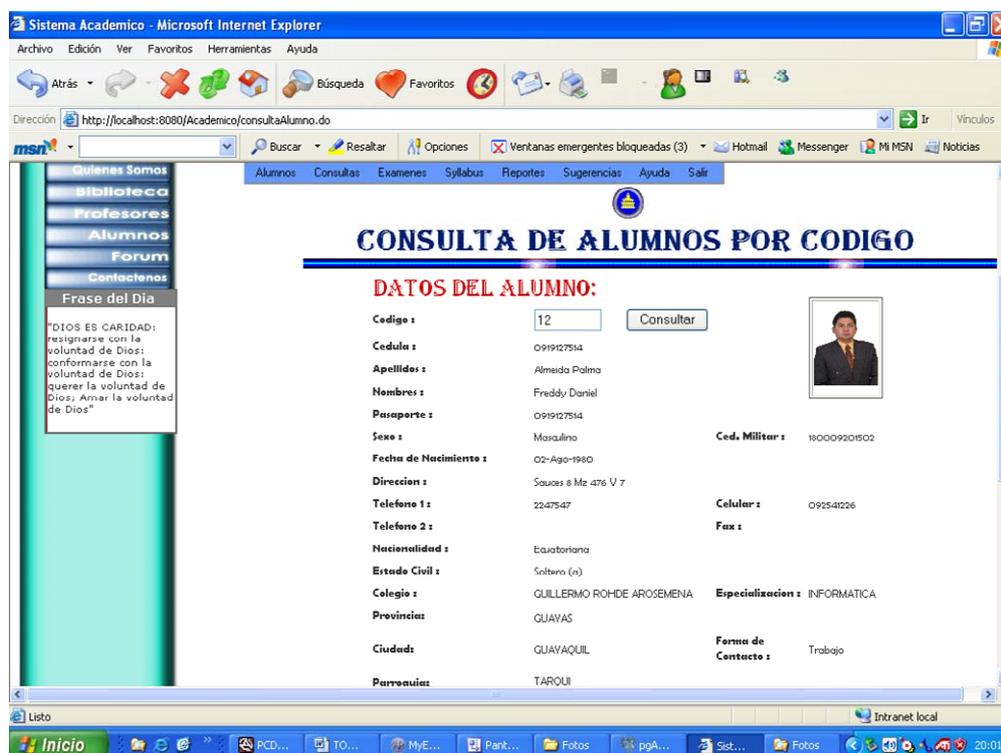


Fig. 3-28. Interfaz de Consulta por Código de Alumno.

De la misma manera una vez presentados los datos podemos regresar a la página principal haciendo un clic sobre el icono de la parte superior de la pantalla o sino puede realizar cualquier otra acción seleccionando alguno del menú principal.

3.9.7. Consulta de un Alumno por Cedula.- Si seleccionamos del menú Consulta por cedula aparecerá una pantalla en donde podrá ingresar el numero de cedula correspondiente a cualquier alumno, si el numero es válido y al presionar el botón de Buscar nos presentará todos los datos correspondientes a ese alumno tal como se muestra en la (fig. 3-29).



Fig. 3-29. Interfaz de Consulta por la Cédula de un Alumno.

En la (fig. 3-30), tenemos la interfaz gráfica de modificación, el único campo habilitado va a ser el del campo código en dicho campo se debe ingresar el código del alumno a modificar, una vez encontrado el alumno se habilitaran los demás campos y se deshabilitará este (campo código), para así poder realizar dicha modificación, y al presionar el botón modificar esta información será modificada en la base de datos.



Fig. 3-30. Interfaz de Modificación de los Datos de un Alumno.

Pero no todos los datos se podrán actualizar solo algunos entre estos mencionamos los siguientes: la dirección, los teléfonos, el celular, la forma de contacto, la dirección donde residen sus padres, el e-mail, de la misma manera podrá actualizar algunos de sus datos laborales como es el nombre de la empresa donde trabaja, la dirección, el número de teléfono y el cargo en que se desempeña, así como se muestra en la (fig. 3-31).

Sistema Academico - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost:8080/Academico/actualizacionAlumno.do>

Nombre de la Madre : *

Dirección de los Padres : *

E-mail :

Nombre del Conyuge :

DATOS LABORALES:

Nombre de la Empresa :

Dirección del Trabajo :

Telefono del Trabajo :

Numero de Afiliación :

Cargo:

(*) Todos los campos con asteriscos son obligatorios

Fig. 3-31. Interfaz de Modificación de los Datos de un Alumno en la parte del botón de Actualizar.

3.9.8. Eliminación de Alumno.- En la (fig. 3-32), tenemos la interfaz gráfica de eliminación, el único campo habilitado va a ser el del campo código en dicho campo se debe ingresar el código del alumno a eliminar, una vez encontrado el alumno se mostraran los datos a eliminar, y al presionar el botón eliminar esta información será anulada de la base de datos.

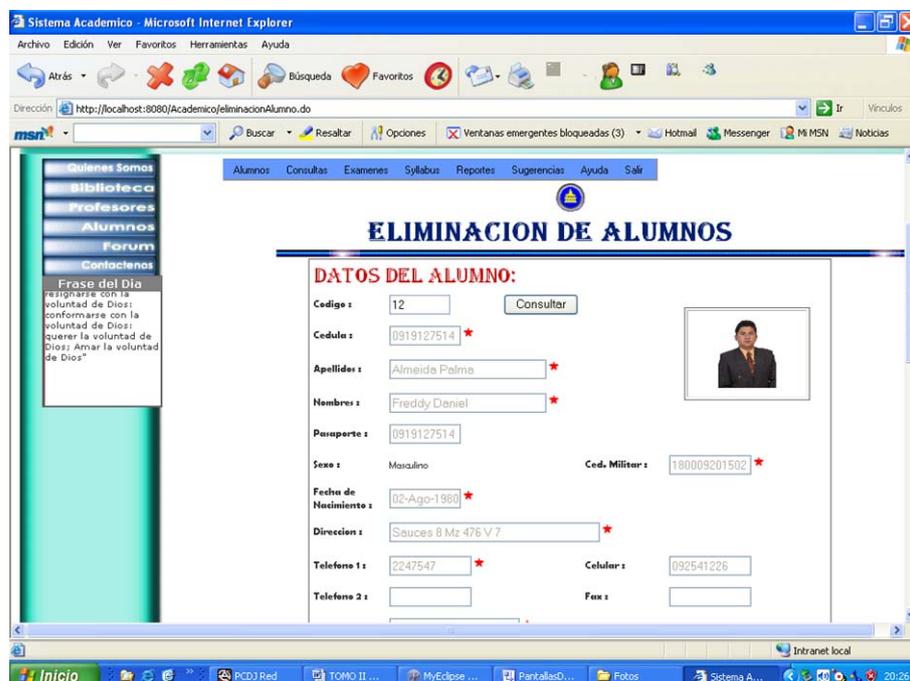


Fig. 3-32. Interfaz de Eliminación de los Datos de un Alumno.

Revisando la información del alumno en la parte inferior de la pantalla se encontrará el botón Eliminar, tal como se muestra en la (fig. 3-33).

Sistema Academico - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección http://localhost:8080/Academico/eliminacionAlumno.do

msn Buscar Resaltar Opciones Ventanas emergentes bloqueadas (3) Hotmail Messenger Mi MSN Noticias

Nombre de la Madre : Cecilia Palma *

Dirección de los Padres : Sauces 8 *

E-mail : danielalmeida@hotmail.com

Nombre del Computo :

DATOS LABORALES:

Nombre de la Empresa : Coop. Gral. Vermeza

Dirección del Trabajo : Cuenca y Cacique Alvarez

Telefono del Trabajo : 2400120

Numero de Afiliación :

Cargos : Técnico

(*) Todos los campos con asteriscos son obligatorios

Eliminar

Intranet local

Inicio PCDJ Red TOMO II ... MyEclipse ... PantallasD... Fotos Sistema A... 20:28

Fig. 3-33. Interfaz de Eliminación de los Datos de un Alumno en la parte del botón de Eliminar.

CAPITULO IV

4. DESARROLLO Y PRUEBA DEL SISTEMA

4.1.- Desarrollo del Sistema

4.1.1. Creación de la base de datos

Las tablas son utilizadas en base de datos para representar la información de forma más compacta y de fácil acceso. Estas tablas están compuestas por filas y columnas. Las filas de una tabla equivalen a los registros los cuales contienen los valores de los objetos, y las columnas equivalen a los campos los cuales contienen los atributos de los objetos.

La estructura de las tablas muestra la forma como se almacenan los datos y en el caso de estar relacionadas con otras tablas cuales serian los campos que se relacionan, de dicha relación nacen los conceptos de campos foráneos y primarios.

4.1.1.2 Clave Primaria

La clave primaria es la de menor tamaño en cuanto a número de campos que la componen y es el identificador único para una tabla. Con la clave primaria nunca existen dos filas de una tabla con el mismo valor.

4.1.1.3. Clave Foránea

Una clave foránea es un atributo de una tabla o adición de atributos ya existentes a una tabla; ésta inclusive puede ser una clave primaria de otra tabla. Sin embargo los atributos de una clave foránea no necesariamente tienen que formar parte de la clave primaria de la tabla a la que pertenecen.

La base de datos se la ha realizado en una herramienta llamada Postgres, la base de datos que usaremos le hemos dado el nombre de Académico, y esta compuesta por las siguientes tablas:

4.1.2. Descripción de la Base de Datos

4.1.2.1. Tabla: Persona

codigopersona, nombres, pasaporte, cedulamilitar, sexo, fechanacimiento, nacionalidad, estadocivil, direccion, telefonocasa, otrotelefono, fax, celular, email, formacontacto, estado.

Esta tabla almacenará toda la información de la persona y se la debe crear antes que la del alumno.

Campo	Tipo	Longitud
codigopersona	Int4	10
nombres	Varchar	50
apellidos	Varchar	50
pasaporte	Varchar	50
cedula	Varchar	10
cedulamilitar	Varchar	15
sexo	Varchar	1
fechanacimiento	Varchar	14
nacionalidad	Varchar	15
estadocivil	Varchar	1
foto	Bytea	100
direccion	Varchar	50
telefonocasa	varchar	10
otrotelefono	varchar	10
fax varchar(10),	varchar	10
celular	varchar	10
email	varchar	50
formacontacto	varchar	20
estado	varchar	2

4.1.2.2. Tabla: Alumno

Se encarga de almacenar toda la información de una persona la cual lo ilustraremos en la siguiente tabla:

codigoalumno, nombrepadre, nombremadre,
 codigospecializacion, codigocolégio, nombreconyuge,
 direccionpadres, estado.

Campo	Tipo	Longitud
codigoalumno,	Int4	10
nombrepadre	Varchar	50
nombremadre	Varchar	50
codigospecializacion	Varchar	10
codigocolégio	Varchar	10
nombreconyuge	Varchar	50
direccionpadres	Varchar	50
estado	Varchar	2

4.1.2.3. Tabla: datoslaborales

Se encarga de almacenar toda la información de las personas que trabajan.

codigoalumno, nombreempresa, direccion, telefono,
 numeroafiliacion, cargo, estado.

Campo	Tipo	Longitud
codigoalumno	Int4	10
nombreempresa	Varchar	50
direccion	Varchar	50
telefono	Varchar	10
numeroafiliacion	Varchar	10
cargo	Varchar	50
estado	Varchar	2

4.2. Objetivo.

La estructura de las tablas muestra detalladamente los campos almacenados y la relación que existe entre cada una de las tablas para comprender el flujo de los datos.

4.3. Definición Estructura de Datos.

4.3.1. Tabla Alumno

alumno	
PK	<u>codigo</u>
	Estado codigoalumno empresa numeroafiliacion cargo direcciontrabajo telefonotrabajo

Esta tabla almacenara la información necesaria de los alumnos.

4.3.2. Tabla Persona

persona	
PK	<u>codigo</u>
	cedula nombre direccion ced_militar telefono celular fec_nacimiento e_mail estado_civil nacionalidad sexo colegio especializacion

Aquí se almacenaran los diferentes datos personales de cada uno de los alumnos ingresados, esta tabla también podría llamarse PERSONA, esta tabla es usada también por profesores debido a que los profesores también son personas al igual que los alumnos.

4.3.3. Tabla Datos_Laborales

Datos laborales	
PK	<u>codigo</u>
	Estado codigoalumno empresa numeroafiliacion cargo direcciontrabajo telefonotrabajo

Esta tabla almacenara todos los datos laborales del alumno, como empresa, dirección, teléfono, número de afiliación, cargo.

4.3.4. Tabla Alumn_Perso

alum_perso	
PK	<u>codigo</u>
	cedula nombre direccion ced_militar telefono celular fec_nacimiento e_mail estado_civil nacionalidad sexo colegio especializacion

Esta es la relación entre alumno y persona, dichas tablas están relacionadas para saber cuales son los las personas que también son alumnos, por que pueden existir personas que no sean alumnos, pueden existir personas que sean solo profesor.

alum_labor	
PK	<u>codigo a</u>
PK	<u>codigo I</u>

Esta es la relación entre alumno y persona, dichas tablas están relacionadas para saber cuales son los alumnos que trabajan, por que pueden existir alumnos que no trabajen.

Los campos q llevan las siguientes tablas son:

4.4. Diccionario de Datos.

4.4.1. Tabla persona

La tabla persona esta compuesta por los siguientes campos:

codigopersona, apellidos, nombres, pasaporte, cedula, cedulamilitar, sexo, fechanacimiento, nacionalidad, estadocivil, direccion, codigoubicaciongeografica, telefonocasa, otrotelefono, fax, celular, email, formacontacto, estado.

codigoalumno int4 (10).- Es una clave primaria el cual va atener el código de la persona dicho código no puede repetido, este código va a tener relación con alumno y datos personales.

apellidos varchar (50).- Almacena el apellido de la persona.

nombres: varchar (50).- Guarda el nombre de la persona.

pasaporte: varchar (50).- Almacena el numero de pasaporte, no es un campo obligatorio.

Cedula: varchar (10).- Almacena la cedula de ciudadanía la cual tiene que ser única y validada para que no acepte un numero invalido.

Cedulamilitar varchar (15).- Almacena la cedula militar este campo es obligatorio para las personas de sexo masculino mayores de 18 años.

fechanacimiento varchar (14).- Almacena la fecha de nacimiento del usuario.

Sexo varchar (1).- Este campo sexo se solamente de 2 caracteres, donde va a guardar "M" si es masculino y "F" si es femenino.

Nacionalidad varchar (15).- Almacena la nacionalidad este es un campo parametrizable.

Estadocivil varchar (1).- Es un campo donde existen 5 opciones y va a guardar "S" si es soltero, "C" Si es casado, "V" si es viudo, "D" si es divorciado y "U" si es unión libre.

Codigoubicaciongeografica varchar (10).- Este es un campo que esta relacionado con las tablas país, provincia, ciudad y parroquia para saber la ubicación geográfica, se la usa para saber el lugar de nacimiento.

Fax varchar (10).- Este es un campo que esta relacionado con las tablas país, provincia, ciudad y parroquia para saber la ubicación geográfica, se la usa para saber el lugar de nacimiento.

Celular varchar (10).- Campo donde se va a guardar el numero de teléfono de la casa de una persona.

Email varchar (50).- Campo donde se va a guardar el numero de fax de una persona.

Formacontacto varchar (20).- En este campo tenemos la información que nos especifica cual es la vía mas rápida de contacto para el alumno.

Estado varchar (2).- Es un campo que va a tener 2 caracteres, para identificar se el alumno ha sido ingresado el campo va a ser "I", si el campo no ha sido ingresado va a tener "E".

Direccion varchar (50).- Guarda la dirección es un campo bastante largo y puede ser modificado.

telefonocasa varchar (10).- Campo donde se va a guardar el numero de teléfono de la casa de una persona.

foto bytea.- Campo donde se va a guardar cualquier imagen, en este caso se almacenará la foto de los alumnos.

4.4.2. Tabla alumno

La tabla alumno esta compuesta por los siguientes campos:

codigoalumno, nombrepadre, nombremadre, direccionpadre,
nombreconyuge, codigocolegio, codigoespecializacion,
codigopersona, estado.

Nombrepadre varchar (50).- Este campo hace referencia al nombre del padre de cada uno de los alumnos ingresados.

Nombremadre varchar (50).- Este campo hace referencia al nombre de la mamá de cada uno de los alumnos ingresados.

Direccionpadre varchar (50).- Este campo almacena la dirección, es un campo bastante largo y puede ser modificado.

Nombreconyuge varchar (50).- En el caso que sea casado sale el nombre del conyuge de lo contrario me devuelve null.

Codigocolegio varchar (10).- Se guarda el código del colegio al cual perteneció el alumno.

Codigospecializacion varchar (10).- Almacena la especialización en la cual se incorporó el alumno al terminar el segundo nivel educativo.

4.4.3. Tabla datoslaborales

La tabla datoslaborales está compuesta por los siguientes campos:

codigoalumno, nombreempresa, direccion, telefono, numeroafiliacion, cargo, estado.

Nombreempresa varchar (50).- Este campo es opcional, aquí se va a almacenar el nombre del trabajo al cual pertenece.

Dirección varchar (50).- Este campo es opcional, aquí se va almacenar la dirección del trabajo al cual pertenece.

Telefono varchar (10).- Este campo es opcional, aquí se va almacenar el numero telefónico del trabajo al cual pertenece.

Numeroafiliacion varchar (10).- En el caso que este afiliado al seguro social (IESS), va a obtener un numero de afiliación el cual va a ser almacenado en este campo.

Cargo varchar (50).- Este campo es opcional, aquí se va almacenar el cargo que ocupa en el trabajo al cual pertenece.

4.5. Script de la Tablas de Nuestra base de Datos

4.5.1 Tabla persona

```
CREATE TABLE persona
(
  codigopersona int4(10) NOT NULL,
  apellidos varchar(50) NOT NULL,
  nombres varchar(50) NOT NULL,
  pasaporte varchar(50),
  cedula varchar(10),
  cedulamilitar varchar(15),
  sexo varchar(1) NOT NULL,
  fechanacimiento varchar(14) NOT NULL,
  nacionalidad varchar(15) NOT NULL,
  estadocivil varchar(1) NOT NULL,
  foto varchar(100),
  direccion varchar(50) NOT NULL,
  telefonocasa varchar(10),
  otrotelefono varchar(10),
  fax varchar(10),
  celular varchar(10),
  email varchar(50),
  formacontacto varchar(20) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT persona_pkey PRIMARY KEY (codigopersona),
)
WITHOUT OIDS;
ALTER TABLE persona OWNER TO postgres;
```

4.5.2. Tabla alumno

```

CREATE TABLE alumno
(
  codigoalumno varchar(10) NOT NULL,
  nombrepadre varchar(50) NOT NULL,
  nombremadre varchar(50) NOT NULL,
  direccionpadre varchar(50) NOT NULL,
  nombreconyuge varchar(50),
  codigocolegio varchar(10) NOT NULL,
  codigospecializacion varchar(10) NOT NULL,
  codigopersona varchar(10) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT alumno_pkey PRIMARY KEY (codigoalumno),
  CONSTRAINT alumno_codigoalumno_fkey1 FOREIGN KEY (codigoalumno)
    REFERENCES persona (codigopersona) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT alumno_codigocolegio_fkey FOREIGN KEY (codigocolegio)
    REFERENCES colegio (codigocolegio) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT alumno_codigospecializacion_fkey
    FOREIGN KEY (codigospecializacion)
    REFERENCES especializacion (codigospecializacion) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE alumno OWNER TO postgres;

```

4.5.3. Tabla datos laborales

```

CREATE TABLE datoslaborales
(
  codigoalumno varchar(10) NOT NULL,
  nombreempresa varchar(50) NOT NULL,
  direccion varchar(50) NOT NULL,
  telefono varchar(10) NOT NULL,
  numeroafiliacion varchar(10) NOT NULL,
  cargo varchar(50) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT datoslaborales_pkey PRIMARY KEY (codigoalumno),
  CONSTRAINT datoslaborales_codigoalumno_fkey
    FOREIGN KEY (codigoalumno)
    REFERENCES alumno (codigoalumno) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE datoslaborales OWNER TO postgres;

```

4.6. Estándares de la Lógica del Negocio.

Clases DAO y BO y relaciona con tablas

Tenemos como archivos principales para que el módulo de alumno funcione correctamente las clases DAO y BO, cada tabla está compuesta por un archivo llamado con el mismo nombre y acompañado por un BO o un DAO, Ejm:

La tabla Alumnos va a tener 2 archivos llamados alumnoBO y alumnoDAO, el alumnoBO es el que va a establecer la conexión de la base de datos ayudado con una tabla llamada BASE la cual va a realizar la conexión, una vez que tenemos la conexión, vamos a utilizar la clase alumnoDAO la cual va a tener las sentencias SQL, es decir es la que va a interactuar con la Base de Datos en este caso con POSTGRES.

Además tenemos más clases BO y DAO las cuales detallamos a continuación:

AlumnoDAO	AlumnoBO
ColegioDAO	ColegioBO
ParroquiaDAO	ParroquiaBO
CiudadDAO	CiudadBO
EspecializaciónDAO	EspecializaciónBO
ProvinciaDAO	Provinciano
MigrarDAO	MigrarBO

ReporteDAO

ReporteBO

SugerenciaDAO

SugerenciaBO

Para mejor apreciación vamos a citar el siguiente ejemplo:

4.6.1. ALUMNOBO

Este código es el que encarga de definir las clases que se están utilizando.

```
package com.cisc.bo;
import com.cisc.bean.Alumno;
import com.cisc.dao.AlumnoDao;
import com.cisc.util.BaseAlumno;
import java.sql.Connection;
import java.util.Vector;
```

Este código define al constructor de la clase.

```
public class AlumnoBO {
    private Connection con = null;
```

Este código se encarga de realizar la conexión a la Base de Datos y desde aquí va a llamar al procedimiento Guardar el cual se encuentra en la clase alumnoDao que es el que interactúa directamente con la Base de Datos realizando los insert, update, etc.

```

public boolean Guardar (Alumno unAlumno) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Guardar(unAlumno);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
}

```

Este código se encarga de crear la conexión para realizar la consulta a la Base de Datos por medio del método buscarCodigo que se encuentra en AlumnoDAO.

```

public Alumno BuscarCodigo (String unCodigo) {
    Alumno valorDevolver = null;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.BuscarCodigo(unCodigo);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
}

```

Este código es el que se encarga de generar automáticamente un código para ingresar una nueva persona este código es generado por medio del método generaCodigo que pertenece a la clase DAO.

```

public String GeneraCodigo () {
    String valorDevolver = null;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDao = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDao.GeneraCodigo();
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}

```

Este código es que se encarga de consultar y a la vez actualizar la Base de Datos todos los datos ingresados por medio del método actualiza de la clase DAO.

```

public boolean Actualizar (Alumno unAlumno) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Actualizar(unAlumno);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            system.out.println(ee);
        }
    }
    return valorDevolver;
}

```

Este código es el que se encarga de eliminar de la Base de Datos todos los datos ingresados desde una Página con extensión Jsp llamada Ingreso Alumno.

El registro no se elimina directamente de la Base de la Datos, ya que existe un campo estado que es el que nos ayuda a saber si el alumno esta activo o inactivo, todo ello lo hace con la ayuda del método elimina de la clase DAO.

```
public boolean Eliminar (String unCodigo) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConexion();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Eliminar(unCodigo);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
```

Una vez realizado la interconexión se llama a cada uno de los métodos los cuales se van a encargar de ingresar, actualizar, etc., todo ello se lo realiza con la ayuda de los métodos DAO que veremos a continuación.

4.6.2. ALUMNODAO

Este código es el que encarga de definir las clases que se están utilizando.

```
package com.cisc.dao;

import com.cisc.bean.Alumno;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.lang.String;
import java.util.Vector;
```

Este código define al constructor de la clase.

```
public class AlumnoDao {
    private Connection con = null;

    public AlumnoDao (Connection unaCon)
    {
        con = unaCon;
    }
}
```

Este código es que se encarga de guardar a la Base de Datos todos los datos ingresados en una Página con extensión Jsp llamada Ingreso Alumno.

```

String sql1 = "INSERT INTO alumno (codigoalumno,nombrepadre,
nombremadre,direccionpadre,nombreconyuge,
codigocollegio,codigospecializacion,codigopersona
,estado) values (?, ?, ?, ?, ?, ?, ?, ?, ?)";

PreparedStatement unPs1 = con.prepareStatement(sql1);
unPs1.setString(1, unAlumno.getCodigoalumno());
unPs1.setString(2, unAlumno.getNombrepadre());
unPs1.setString(3, unAlumno.getNombremadre());
unPs1.setString(4, unAlumno.getDireccionpadres());
unPs1.setString(5, unAlumno.getNombreconyuge());
unPs1.setString(6, unAlumno.getCodigocollegio());
unPs1.setString(7, unAlumno.getCodigospecializacion());
unPs1.setString(8, unAlumno.getCodigopersona());
unPs1.setString(9, unAlumno.getEstado());
int i1 = unPs1.executeUpdate();

String sql2 = "INSERT INTO datoslaborales (codigoalumno,nombreempresa,
direccion,telefono,numeroafiliacion,cargo,estado)
values (?, ?, ?, ?, ?, ?, ?)";
PreparedStatement unPs2 = con.prepareStatement(sql2);
unPs2.setString(1, unAlumno.getCodigoalumno());
unPs2.setString(2, unAlumno.getNombretrabajo());
unPs2.setString(3, unAlumno.getDirecciontrabajo());
unPs2.setString(4, unAlumno.getTelefonotrabajo());
unPs2.setString(5, unAlumno.getNumeroafiliacion());
unPs2.setString(6, unAlumno.getCargo());
unPs2.setString(7, unAlumno.getEstado());
int i2 = unPs2.executeUpdate();

```

```

String sql3 = "INSERT INTO persona (codigopersona,apellidos,nombres,
pasaporte,cedula,cedulamilitar,sexo,fechanacimiento,
nacionalidad,estadocivil,direccion,codigoubicaciongeografica,
telefonocasa,otrotelefono,fax,celular,email,formacontacto,estado)
values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
PreparedStatement unPs3 = con.prepareStatement(sql3);
unPs3.setString(1, unAlumno.getCodigoalumno());
unPs3.setString(2, unAlumno.getApellidos());
unPs3.setString(3, unAlumno.getNombres());
unPs3.setString(4, unAlumno.getPasaporte());
unPs3.setString(5, unAlumno.getCedula());
unPs3.setString(6, unAlumno.getCedulamilitar());
unPs3.setString(7, unAlumno.getSexo());
unPs3.setString(8, unAlumno.getFecha());
unPs3.setString(9, unAlumno.getNacionalidad());
unPs3.setString(10, unAlumno.getEstadocivil());
unPs3.setString(11, unAlumno.getDireccion());
unPs3.setString(12, unAlumno.getCodigoubicacion());
unPs3.setString(13, unAlumno.getTelefono1());
unPs3.setString(14, unAlumno.getTelefono2());
unPs3.setString(15, unAlumno.getFax());
unPs3.setString(16, unAlumno.getCelular());
unPs3.setString(17, unAlumno.getEmail());
unPs3.setString(18, unAlumno.getFormacontacto());
unPs3.setString(19, unAlumno.getEstado());
int i3 = unPs3.executeUpdate();

```

Este código es el que se encarga de generar automáticamente un código para ingresar una nueva persona desde una Pagina con extensión Jsp llamada Ingreso Alumno.

```

public String GeneraCodigo() throws Throwable
{
    String valorDevolver = new String();
    String sql = "SELECT max(codigopersona) as codigo FROM persona";
    PreparedStatement unPs = con.prepareStatement(sql);
    ResultSet unRs = unPs.executeQuery();
    while (unRs.next()){
        valorDevolver = new String();
        valorDevolver = (unRs.getString("codigo"));
    }

    unRs.close();
    unPs.close();
    int valorDevolver1 = new Integer(valorDevolver).intValue() + 1;
    valorDevolver = new String ("0000" + valorDevolver1);
    return valorDevolver;
}

```

Este código es que se encarga de consultar a la Base de Datos todos los datos ingresados desde una Pagina con extensión Jsp llamada Ingreso Alumno.

```

Alumno unAlumno = new Alumno();
String sql = "SELECT a.codigopersona as codpersona,apellidos,nombres,pasaporte,cedula,
cedulamilitar,sexo,fechanacimiento,nacionalidad,estadocivil,a.direccion
as direc,codigoubicaciongeografica,telefonocasa,otrotelefono,fax,celular,
email,formacontacto,\n" +
" b. codigoalumno as codalumno,nombrepadre,nombremadre,direccionpadre,nombreconyuge,
codigocolegio,codigospecializacion,\n" +
" nombreempresa,c.direccion as dir,telefono,numeroafiliacion,cargo\n" +
"From persona as a, alumno as b, datoslaborales as c\n" +
"where b.codigoalumno = a.codigopersona and b.codigoalumno = c.codigoalumno and
b.codigoalumno = ?";
PreparedStatement unPs = con.prepareStatement(sql);
unPs.setString(1,unCodigo);
ResultSet unRs = unPs.executeQuery();
while (unRs.next()){
    unAlumno.setCodigoalumno(unRs.getString("codpersona"));
    unAlumno.setNombres(unRs.getString("nombres"));
    unAlumno.setApellidos(unRs.getString("apellidos"));
    unAlumno.setPasaporte(unRs.getString("pasaporte"));
    unAlumno.setCedula(unRs.getString("cedula"));
    unAlumno.setCedulamilitar(unRs.getString("cedulamilitar"));
    unAlumno.setSexo(unRs.getString("sexo"));
    unAlumno.setFecha(unRs.getString("fechanacimiento"));
    unAlumno.setNacionalidad(unRs.getString("nacionalidad"));
    unAlumno.setEstadocivil(unRs.getString("estadocivil"));
    unAlumno.setDireccion(unRs.getString("direc"));
    unAlumno.setCodigoubicacion(unRs.getString("codigoubicaciongeografica"));
    unAlumno.setTelefono1(unRs.getString("telefonocasa"));
    unAlumno.setTelefono2(unRs.getString("otrotelefono"));
    unAlumno.setFax(unRs.getString("fax"));
    unAlumno.setCelular(unRs.getString("celular"));
    unAlumno.setEmail(unRs.getString("email"));
    unAlumno.setFormacontacto(unRs.getString("formacontacto"));
    unAlumno.setNombrepadre(unRs.getString("nombrepadre"));
    unAlumno.setCodigoalumno(unRs.getString("codalumno"));
    unAlumno.setNombremadre(unRs.getString("nombremadre"));
    unAlumno.setDireccionpadres(unRs.getString("direccionpadre"));
    unAlumno.setNombreconyuge(unRs.getString("nombreconyuge"));
    unAlumno.setCodigocolegio(unRs.getString("codigocolegio"));
    unAlumno.setTelefonoTrabajo(unRs.getString("telefono"));
    unAlumno.setNumeroafiliacion(unRs.getString("numeroafiliacion"));
    unAlumno.setCargo(unRs.getString("cargo"));
}

unRs.close();
unPs.close();
return unAlumno;

```

Este código es que se encarga de consultar y a la vez actualizar la Base de Datos todos los datos ingresados desde una Pagina con extensión Jsp llamada Ingreso Alumno.

```
String sql1 = "UPDATE alumno\n" +
"set nombrepadre = ?, nombremadre = ?, direccionpadre = ?, nombrecony
"where codigoalumno = ?";
PreparedStatement unPs1 = con.prepareStatement(sql1);
unPs1.setString(1, unAlumno.getNombrepadre());
unPs1.setString(2, unAlumno.getNombremadre());
unPs1.setString(3, unAlumno.getDireccionpadre());
unPs1.setString(4, unAlumno.getNombreconyuge());
unPs1.setString(5, unAlumno.getCodigocolegio());
unPs1.setString(6, unAlumno.getCodigospecializacion());
unPs1.setString(7, unAlumno.getCodigopersona());
unPs1.setString(8, unAlumno.getCodigoalumno());
int i1 = unPs1.executeUpdate();

String sql2 = "UPDATE datoslaborales\n" +
"set nombreempresa = ?, direccion = ?, telefono = ?, numeroafiliacion
"where codigoalumno = ?";
PreparedStatement unPs2 = con.prepareStatement(sql2);
unPs2.setString(1, unAlumno.getNombretrabajo());
unPs2.setString(2, unAlumno.getDirecciontrabajo());
unPs2.setString(3, unAlumno.getTelefonotrabajo());
unPs2.setString(4, unAlumno.getNumeroafiliacion());
unPs2.setString(5, unAlumno.getCargo());
unPs2.setString(6, unAlumno.getCodigoalumno());
int i2 = unPs2.executeUpdate();

unPs1.close();
unPs2.close();
unPs3.close();

if (i1==1 && i2==1 && i3==1)
    return true;
else
    return false;
```

```
String sql3 = "UPDATE persona \n" +
"set apellidos = ?, nombres = ?, pasaporte = ?, cedula = ?,
cedulamilitar = ?, sexo = ?, fechanacimiento = ?,
nacionalidad = ?, estadocivil = ?, \n" +
" direccion = ?, codigoubicaciongeografica = ?,
telefonocasa = ?, otrotelefono = ?, fax = ?,
celular = ?, email = ?, formacontacto = ?\n" +
"where codigopersona = ?";
PreparedStatement unPs3 = con.prepareStatement(sql3);
unPs3.setString(1, unAlumno.getApellidos());
unPs3.setString(2, unAlumno.getNombres());
unPs3.setString(3, unAlumno.getPasaporte());
unPs3.setString(4, unAlumno.getCedula());
unPs3.setString(5, unAlumno.getCedulamilitar());
unPs3.setString(6, unAlumno.getSexo());
unPs3.setString(7, unAlumno.getFecha());
unPs3.setString(8, unAlumno.getNacionalidad());
unPs3.setString(9, unAlumno.getEstadocivil());
unPs3.setString(10, unAlumno.getDireccion());
unPs3.setString(11, unAlumno.getCodigoubicacion());
unPs3.setString(12, unAlumno.getTelefono1());
unPs3.setString(13, unAlumno.getTelefono2());
unPs3.setString(14, unAlumno.getFax());
unPs3.setString(15, unAlumno.getCelular());
unPs3.setString(16, unAlumno.getEmail());
unPs3.setString(17, unAlumno.getFormacontacto());
unPs3.setString(18, unAlumno.getCodigoalumno());
int i3 = unPs3.executeUpdate();
```

Este código es el que se encarga de eliminar de la Base de Datos todos los datos ingresados desde una Pagina con extensión Jsp llamada Ingreso Alumno.

El registro no se elimina directamente de la Base de la Datos, ya que existe un campo estado que es el que nos ayuda a saber si el alumno esta activo o inactivo.

```

public boolean Eliminar (String unCodigo) throws Throwable
{
    String sql3 = "UPDATE persona set estado = 'E' where codigopersona = ?";
    PreparedStatement unPs3 = con.prepareStatement(sql3);
    unPs3.setString(1,unCodigo);
    int i3 = unPs3.executeUpdate();

    String sql1 = "UPDATE alumno set estado = 'E' where codigoalumno = ?";
    PreparedStatement unPs1 = con.prepareStatement(sql1);
    unPs1.setString(1,unCodigo);
    int i1 = unPs1.executeUpdate();

    String sql2 = "UPDATE datoslaborales set estado = 'E' where codigoalumno = ?";
    PreparedStatement unPs2 = con.prepareStatement(sql2);
    unPs2.setString(1,unCodigo);
    int i2 = unPs2.executeUpdate();

    unPs1.close();
    unPs2.close();
    unPs3.close();

    if (i1==1 && i2==1 && i3==1)
        return true;
    else
        return false;
}

```

Todos estos procesos antes mencionados son los que se utilizaran para la realización de nuestro módulo.

4.7 Seguridades

Podemos entender como seguridad una característica de cualquier sistema informático que nos indica que ese sistema está libre de

peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para la mayoría de los expertos el concepto de seguridad en la informática es un tópico porque no existe un sistema 100% seguro. Para que un sistema se pueda definir como seguro debemos de dotar de tres características al mismo:

- Integridad
- Confidencialidad
- Disponibilidad

4.7.1. Seguridad de sistema operativo

Además, con la generalización de las conexiones con Internet y el rápido desarrollo del software, la seguridad se está convirtiendo en una cuestión cada vez más importante. Ahora, la seguridad es un requisito básico, ya que la red global es insegura por definición. Mientras sus datos estén almacenados en un soporte informático, mientras sus datos vayan desde un sistema a otro sistema y a través de un medio físico, Internet, por ejemplo, puede pasar por ciertos puntos durante el camino proporcionando a otros usuarios la posibilidad de interceptarlos, e incluso alterar la información

contenida. Incluso algún usuario de su sistema puede modificar datos de forma maliciosa para hacer algo que nos pueda resultar perjudicial. Con el acceso masivo y barato a Internet se han reducido notablemente los costes de un atacante para asaltar un sistema en red, a la vez que ha aumentado paralelamente el número de potenciales atacantes.

A nadie le gustaría que desconocidos abran su correo privado, que miren en sus cajones, que se hagan copias de las llaves de su escritorio o de la tarjeta de crédito. Pues todo esto es aplicable en la misma medida a las redes telemáticas.

Nuestro sistema es seguro ya que nadie puede ingresar al mismo si no es un usuario calificado, al ingresar debe escribir la clave y contraseña para poder hacer uso de la opciones del menú, además contamos con un usuario especial llamado admin, el cual se encarga de dar mantenimiento a nuestro modulo alumno, para tener seguridad se deben seguir las siguientes recomendaciones:

- Decidir cuáles servicios se necesita, y limitar el sistema para ellos. Esto incluye desactivación / desinstalación de servicios innecesarios.
- Limitar usuarios y permisos

- Usamos herramientas apropiadas, para garantizar que el uso desautorizado se detecte, de tal manera que se puede tomar las medidas apropiadas, en este caso no podrán ingresar a nuestro Sistema Académico.

4.7.2. Seguridad física

Las primeras medidas de seguridad que necesita tener en cuenta son las de seguridad física de sus sistemas. Hay que tomar en consideración quienes tienen acceso físico a las máquinas y si realmente deberían acceder.

El nivel de seguridad física que necesita en su sistema depende de su situación concreta. Un usuario doméstico no necesita preocuparse demasiado por la protección física, salvo proteger su máquina de un niño o algo así. En una oficina puede ser diferente.

Linux proporciona los niveles exigibles de seguridad física para un sistema operativo:

- Un arranque seguro
- Posibilidad de bloquear las terminales
- Por supuesto, las capacidades de un sistema multiusuario real.

4.7.3. Seguridades de red

La seguridad de las conexiones en red merecen en la actualidad una atención especial, incluso por medios de comunicación no especializados, por el impacto que representan los fallos ante la opinión pública.

El propio desarrollo tanto de Linux, como de la mayoría del software que lo acompaña, es de fuentes abiertas. Podemos ver y estudiar el código. Esto tiene la ventaja de que la seguridad en Linux no sea una mera apariencia, sino que el código está siendo escrutado por muchas personas distintas que rápidamente detectan los fallos y los corrigen con una velocidad asombrosa.

Si además comprendemos los mecanismos que se siguen en las conexiones en red, y mantenemos actualizados nuestros programas, podemos tener un nivel de seguridad y una funcionalidad aceptables.

Tampoco tienen las mismas necesidades de seguridad un equipo doméstico, con conexiones esporádicas a Internet, que un servidor conectado permanentemente y que actúe como pasarela entre una intranet e Internet.

4.7.4. Seguridades del administrador

A menudo, el mayor enemigo del sistema es el propio administrador del sistema, sí, tiene todos los privilegios y cualquier acción puede ser irreversible y hacerle perder posteriormente mucho más tiempo que el que hubiera perdido por realizar las tareas de forma segura. Puede borrar cualquier fichero e incluso destruir el propio sistema, mientras que un usuario «normal» sólo puede perjudicarse a sí mismo. Por estos motivos, conseguir privilegios de administrador es la meta de cualquier ataque.

En nuestro sistema existe un solo administrador el cual se va a encargar de darle mantenimiento a las provincias, ciudades, parroquias, colegios y especializaciones.

4.8 ETAPA DE PRUEBAS

4.8.1. Prueba de especificaciones: (código)

En esta etapa se deben cumplir con las especificaciones planteadas por el Modulo y cómo debe desempeñarlas, no es una prueba completa, debido a que aún el módulo no está concluido, pero si el programa cumple estas especificaciones no

fallará, en este instante se ha cumplido con la primera fase de la prueba.

Ej: Hemos probado el ingreso del alumno pero sin las debidas validaciones ya que el objetivo primordial era ingresar dicha información a la Base de Datos.

4.8.2. Prueba Unitaria:

En esta etapa hemos probado en forma independiente todos los procedimientos que conforman el módulo, tales como ingreso, consulta, modificación y eliminación, a esta etapa de prueba también se lo denomina prueba de programas.

Se realiza esta prueba independiente, para localizar errores en la lógica y en la codificación, se trata de evitar aquellos errores que surgen en el momento de la interacción.

Podemos llevar a cabo esta prueba de dos maneras la Ascendente y la descendente.

4.8.2.1. La prueba Ascendente.- Se lleva a cabo de abajo hacia arriba, es decir comenzando con los módulos más pequeños y de menor nivel uno a la vez.

4.8.2.2. La prueba descendente.- Es la inversa de la ascendente, se comienza por los módulos de arriba hacia los de abajo, pero como no se llega a probar los módulos inferiores se utilizan los que se llaman módulos esclavos, simplemente para mandar mensajes de OK ó de ERROR para los módulos superiores, los módulos de menor nivel no se prueban, por eso a veces se combina la prueba ascendente con la descendente.

4.8.3. Prueba de Sistemas

No prueba el Software, sino la integración de cada uno de los procedimientos pertenecientes a dicho modulo, también hace pruebas para encontrar discrepancias entre el sistema y su objetivo original, y la documentación del mismo. La preocupación es la compatibilidad de los módulos individuales.

Ej. Error en el tipo de un campo que da una salida irreal, este ejemplo se da en el módulo de consulta de alumnos al darnos un dato que no es acorde con la realidad.

4.8.4. Pruebas especiales de Sistemas

4.8.4.1. Prueba de Carga Pico:

Es donde se prueba si el sistemas es o no calificado como multiusuario, esta prueba se lo hace más a menudo en los sistemas en línea tipo Bancario, es una prueba donde se buscan situaciones reales, para este caso tenemos un servidor en Linux el cual nos va ayudar a realizar este proceso.

4.8.4.2. Prueba de almacenamiento:

Cuando se diseña el sistema se analizó el espacio que puede llegar a ocupar, buscando cierta compatibilidad con el espacio de Disco que se va a utilizar en la computadora que se va a correr. En este caso nuestro servidor de Linux, justamente esto se tiene que probar antes de la puesta en marcha.

4.8.4.3. Prueba de desempeño en tiempo

Este proceso se pone en práctica antes de la puesta en marcha para determinar cuánto tomará recibir la respuesta a una consulta u obtener una copia de respaldo de un archivo o copia de la base de datos para llevar un historial, la idea es no tener

que padecer la pérdida de datos reales por el tiempo de respuesta.

4.8.4.4. Prueba de Recuperación:

Cuando se crea una situación de pérdida ó de falla de datos donde los usuarios se vean obligados a volver a cargar y recuperar una información, se puede determinar si los procedimientos de recuperación son adecuados.

Ej. Al Ingresar los datos de los alumnos tenemos datos que son obligatorios y si esos datos no son ingresados nos muestra mensajes de error y el usuario debe ingresar la información correspondiente sin perderse los datos ya ingresados.

4.8.4.5. Prueba de procedimiento

Se debe tener mucho énfasis en la creación de los manuales de procedimiento, ya que un buen diseño de ellos es insustituible, sobre todo siendo lo más críticos posible en los detalles del manejo del sistema incluyendo todo esto en la documentación.

Lo que en realidad se necesita es ejercer la prueba de Recursos Humanos. Donde se observa en el usuario las acciones y reacciones que va teniendo en el manejo del Sistema.

Ej. En el ingreso de datos nos cuidamos de pantallas en blanco donde quizás el usuario no sabe que hacer. Se debe anticipar a las preguntas de los usuarios cuando interactúan con el sistema.

Otra prueba importante es ver como el usuario carga los datos en el sistema, toda esta etapa de prueba la hemos realizado al presentar cada avance a cada uno de nuestros instructores los cuales han sido los usuarios finales.

4.8.7. Diseño de Datos de prueba

4.8.5.1. Utilización de datos reales de prueba.

Estos datos son los que se extraen de usuarios finales. Se les piden a los usuarios que carguen datos como si estuvieran en realidad realizando sus actividades normales.

Conseguir datos reales suficientes no siempre se logra y a veces no se ven todas las combinaciones o formatos que pueden llegar a introducirse.

Ej. En el ingreso de alumnos necesitamos datos reales ya que tenemos un procedimiento que nos valida el numero de cédula si es o no correcto, al ingresar un número de cédula incorrecto el sistema no lo acepta, por ello debemos ingresar datos que sean coherentes y reales.

4.8.5.2. Utilización de datos artificiales de prueba.

Estos datos se crean únicamente para propósitos de prueba, dado que se arman para probar todo aquello que con los datos reales no se pudo. Estas pruebas la realizamos nosotros mismo al momento que estamos desarrollando cada uno de los procesos.

4.8.8. Bibliotecas de prueba

Estas bibliotecas sirven para probar el sistema cuando ya está avanzado, es como un mantenimiento mientras el sistema va evolucionando y los programas se van modificando y se deben volver a probar.

Por la tanto las Bibliotecas de datos se deben mantener a través de la vida del sistema, ya que conforme se realiza cada cambio se

tengan disponibles otra vez datos confiables para probar el sistema.

Ej. En el momento de hacer el ingreso de los alumno se va a ir realizando esta biblioteca de pruebas ya que cada vez que se van haciendo nuevas validaciones se va modificando el sistema y hay que volver a realizar la prueba hasta que el módulo este concluido.

CAPITULO V

5. IMPLEMENTACION DEL SISTEMA

5.1. Implementación de Sistemas

Cuando se definen aplicaciones, se pueden advertir problemas de configuración, conexión y validación. Estos resultados de validación pueden proporcionar valiosa información sobre problemas que podrían pasar desapercibidos hasta el momento de la implementación, y por lo tanto se podrían evitar costosos cambios más adelante.

5.2. Objetivo Implementación de Sistemas

Poner en práctica lo establecido en todo el proceso de codificación.

1. Seguir el cronograma de implantación.
2. Determinar la capacitación.
3. Poner en práctica lo establecido en las necesidades.

5.2.1. Seguir el cronograma de implantación.- Una vez que empezamos a implementar el software debemos regirnos al cronograma establecido al inicio del proyecto para saber si lo estamos implementando en el momento preciso y no tener inconvenientes en el plazo de la entrega.

5.2.2. Determinar la capacitación.- En esta etapa de la implementación se debe capacitar a los usuarios que van a ser los que se encarguen de interactuar con el sistema para que éste desarrolle como se esperaba y no tenga problemas al momento que se lo este ejecutando.

5.2.3. Implementar lo establecido en las necesidades.- Una vez que el sistema se está implementando se debe considerar los requerimientos para el cual fue creado, es decir que cumpla con todos los requerimientos que se hicieron al iniciar el proyecto, y en el caso que no cumpla con alguno de ellos se debe volver a implementar hasta que este se encuentre acorde con los requerimientos.

5.3. Elementos Físicos

Dentro de nuestro proyecto tenemos a los equipos que vamos a necesitar para la implementación, los cuales detallamos a continuación:

El **Hardware**, es un dispositivo electrónico físico que será usado para desarrollar el módulo a implementar, el cual siempre está disponible desde el lunes 24 de enero del 2006, hasta la culminación del proyecto.

Como mínimo 2 computadores personal con las siguientes características:

- Computador que sirve como navegador.
- Servidor Linux para cargar la aplicación en Internet, el cual va a trabajar en conjunto con Apache.

5.4. Elemento Lógico

Los elementos lógicos son aquellos programas que utilizamos para poder realizar nuestro proyecto, Software como Linux, el cual va a levantar la aplicación en Apache Tomcat el cual es el servidor de Internet, también utilizamos JDeveloper que fue donde se programó el

software, y utilizamos MyEclipse el cual interactúa directamente con Apache.

5.5. Elementos Humanos

Son los operadores de las herramientas (Software) a usar para desarrollar la implementación, y son los directos responsables y autores del mismo.

- Daniel Almeida Palma.
- Stalin Chilán Quimis.
- Lubert Quiroz Pita.

CAPITULO VI

6. RECOMENDACIONES Y CONCLUSIONES

6.1 Recomendaciones

6.1.1. Hardware

- Procesador Pentium IV de 3 GHz. De 64 bits.
- 512 MB de memoria RAM.
- Disco duro de 80 GB.
- Unidad de CD-Writer.
- Mainboard con Video / Red.
- Monitor, Teclado y Mouse.
- Regulador UPS de 650 w.

6.1.2. Software

- Sistema Operativo Linux
- Programación en JSP, SERVLETS, XML
- Base de Datos POSTGRESS

6.1.3. Cableado

- Existe conexión inalámbrica vía Internet.

6.1.4. Puesta en marcha

La puesta en marcha es el proceso que determina la diferencia entre el éxito y el fracaso en la implantación. Consiste de 6 actividades básicas:

- Consultoría
- Programación
- Instalación
- Capacitación
- Registro de datos iniciales
- Lanzamiento.

En la consultoría elaboramos el proyecto y tenemos que presentarlo como tesis de grado para la obtención de nuestro título como Ingeniero en Sistemas Computacionales, y culmina con la presentación de la misma ante un jurado delegado por la Carrera.

Las actividades de programación, capacitación y registro de datos iniciales son de preparación de la puesta en marcha, que la efectuamos en el proceso de lanzamiento. Durante éstas actividades

de preparación se sigue operando con los métodos y procedimientos que tenga actualmente establecidos, de manera que sus operadores no son sometidos a un stress innecesario.

Es hasta el lanzamiento que migramos al nuevo sistema.

Garantía.- En el caso que exista algún inconveniente con el software los únicos encargados de revisarlos o darle un mantenimiento en el caso que éste lo amerite son las personas que la han desarrollado, ellos son los directos responsables del modulo alumno.

Consultaría	Objetivos	Contribuir con la misión, objetivos y estrategias de la carrera. Plan de puesta en marcha
	Medios	Entrevistas con el alumnos, recopilación de información. Análisis de documentos y reportes vigentes Presentación de evaluaciones de la herramienta
	Entregables	Propuesta del proyecto.
Programación	Objetivos	Desarrollar y modificar de acuerdo a los requerimientos.
Instalación	Objetivo	Montar y configurar el software requerido en el servidor.
	Medios	Instalación y configuración del sistema operativo del servidor.

		Instalación y configuración de Base de datos en el servidor
Capacitación	Objetivos	<p>Que el personal conozca los nuevos métodos y procedimientos de los procesos a seguir.</p> <p>Que los usuarios sean capaces de utilizarlo correctamente</p> <p>Que se cuente con personal capaz de operar el servidor.</p>
Lanzamiento	Objetivo	Comenzar la operación del sistema conforme a las especificaciones establecidas en el proyecto.

6.2. Conclusión

Nuestro Sistema Informático (Software) de la Carrera de Ingeniería en Sistemas Computacionales, específicamente del módulo de alumnos es en la actualidad un hecho, puesto que tenemos información a la mano de cada uno de los alumnos de la carrera simplemente con acceder a las consultas desarrollada en éste módulo, la única condición que se necesita es ingresar a la dirección Web por medio de un Navegador de Internet (browser) con su respectiva conexión a Internet que se lo puede encontrar fácilmente desde cualquier Cyber, Instituciones educativas o desde la comodidad de su casa, siempre y cuando tenga acceso a Internet.

Hemos puesto nuestro mayor esfuerzo para que éste módulo sea concluido con éxito, gracias a la ayuda de cada uno de los directivos y profesores de la carrera que supieron guiarnos impartiéndonos sus conocimientos, y queda demostrado que con esfuerzo y dedicación se pueden cumplir todos los objetivos trazados a lo largo de nuestras vidas.

UNIVERSIDAD DE GUAYAQUIL

Facultad de Ciencias Matemáticas y Físicas

**Carrera de Ingeniería en Sistemas
Computacionales**

“Sistema Académico - Módulo Alumnos”

**MANUAL TECNICO
MANUAL DE USUARIO**

TESIS DE GRADO

Previo a la Obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Autor(es):

**Daniel Almeida Palma
Stalin Chilán Quimis
Lubert Quiroz Pita**

GUAYAQUIL – ECUADOR

Año: 2006

INDICE GENERAL

CAPITULO I

MANUAL TECNICO

1.1. Estándares del Sistema.	1
1.2. Objetivo.	1
1.3. Descripción de estándares.	2
1.4. Diseño de Tablas.	2
1.4.1. Objetivo.	3
1.5. Definición Estructura de Datos.	4
1.5.1. Tabla persona	7
1.5.2. Tabla alumno	7
1.5.3. Tabla datos laborales	7
1.6. Script de las Tablas de Nuestra base de Datos	8
1.6.1 Tabla persona	8
1.6.2 Tabla alumno	8
1.6.3. Tabla datos laborales	9
1.7. Diccionario de Datos.	9
1.8. Estándares de la Lógica del Negocio.	11

CAPITULO II

MANUAL DE USUARIO

2.1. Descripción	23
2.2. Objetivo	23
2.3. Detalle de Acciones y Respuestas del Sistema	24
2.3.1 Pantalla de Bienvenida General	24
2.3.1.1. Ingreso de un Alumno	27
2.3.1.2 Consulta de un Alumno por el Apellido	30
2.3.1.3 Consulta de un Alumno por Código	33
2.3.1.4 Consulta de un Alumno por la Cédula	34
2.3.1.5 Modificación de un Alumno por Código	35
2.3.1.6 Eliminación de un Alumno por Código	38
2.3.1.7 Subir Exámenes	40
2.3.1.8 Descargar Exámenes	41
2.3.1.9 Subir Syllabus	43
2.3.1.10 Descargar Syllabus	45
2.3.1.11 Generación de Reportes para Alumnos	47
2.3.2. Mantenimiento de Atributos	48

CAPITULO III**MANUAL DEL ADMINISTRADOR**

3.1. Descripción	49
3.2. Objetivo	49
3.3. Detalle de Acciones y Respuestas del Sistema	50
3.3.1 Pantalla de Bienvenida General	50
3.3.2 Ingreso de una Provincia	51
3.3.3. Consulta de una Provincia	54
3.3.4. Modificación de una Provincia	55
3.3.5. Eliminación de una Provincia	56
3.3.6. Ingreso de una Ciudad	57
3.3.7. Consulta de una Ciudad	59
3.3.8. Modificación de una Ciudad	60
3.3.9. Eliminación de una Ciudad	61
3.3.10. Ingreso de una Parroquia	62
3.3.11. Consulta de una Parroquia	63
3.3.12. Modificación de una Parroquia	64
3.3.13. Eliminación de una Parroquia	65
3.3.14. Ingreso de un Colegio	66

3.3.15. Consulta de un Colegio	67
3.3.16. Modificación de un Colegio	68
3.3.17. Eliminación de un Colegio	69
3.3.18. Ingreso de una Especialización	70
3.3.19. Consulta de una Especialización	71
3.3.20. Modificación de una Especialización	72
3.3.21. Eliminación de una Especialización	73

CAPITULO I

1. MANUAL TECNICO

1.1. Estándares del Sistema.

Son acuerdos que contienen especificaciones técnicas u otros criterios específicos para ser usados como guías o definiciones de características de nuestro Modulo de Alumnos, sirven para asegurar que materiales procesos y servicios son obtenidos o han sido realizados de acuerdo a sus propósitos.

1.2. Objetivo.

El objetivo de los Estándares desarrollados por nuestro módulo se lo detalla a continuación:

- Creación de la Base de Datos.
- Definir requerimientos obligatorios para el manejo de los sistemas y la generación de informes.

- Para saber cual es la logística que se ha llevado a cabo para la realización del Modulo.
- Es una Asistente para el mantenimiento del mismo.

1.3. Descripción de estándares.

Objetos de la base de datos.

La creación de las Bases de Datos con sus respectivas tablas que van a estar relacionadas entre si, entre las principales tablas tenemos:

Alumno	Ciudad	Colegio
Datos_laborales	Especialización	Pais
Parroquia	Persona	Provincia
Ubicación_Geografica	MateriaExamen	Aux
Sugerencias		

1.4. Diseño de Tablas.

Las tablas son utilizadas en base de datos para representar la información de forma más compacta y de fácil acceso. Estas tablas están compuestas por filas y columnas. Las filas de una tabla equivalen a los registros los cuales contienen los valores de los objetos, y las columnas equivalen a los campos los cuales contienen los atributos de los objetos.

La estructura de las tablas muestra la forma como se almacenan los datos y en el caso de estar relacionadas con otras tablas cuales serian los campos que se relacionan, de dicha relación nacen los conceptos de campos foráneos y primarios.

Clave Primaria

La clave primario es la de menor tamaño en cuanto a número de campos que la componen y es el identificador único para una tabla. Con la clave primaria nunca existen dos filas de una tabla con el mismo valor.

Clave Foránea

Una clave foránea es un atributo de una tabla o adición de atributos ya existentes a una tabla; ésta inclusive puede ser una clave primaria de otra tabla. Sin embargo los atributos de una clave foránea no necesariamente tienen que formar parte de la clave primaria de la tabla a la que pertenecen.

1.4.1. Objetivo.

La estructura de las tablas muestra detalladamente los campos almacenados y la relación que existe entre cada una de las tablas para comprender el flujo de los datos.

1.5. Definición Estructura de Datos.

TABLA ALUMNO

alumno	
PK	<u>codigo</u>
	Estado codigoalumno empresa numeroafiliacion cargo direcciontrabajo telefonotrabajo

Esta tabla almacenara la información necesaria de los alumnos.

TABLA DATOS PERSONALES

persona	
PK	<u>codigo</u>
	cedula nombre direccion ced_militar telefono celular fec_nacimiento e_mail estado_civil nacionalidad sexo colegio especializacion

Aquí se almacenaran los diferentes datos personales de cada uno de los alumnos ingresados, esta tabla también podría llamarse PERSONA, esta tabla es usada también por profesores debido a que los profesores también son personas al igual que los alumnos.

TABLA DATOS LABORALES

Datos laborales	
PK	<u>codigo</u>
	Estado codigoalumno empresa numeroafiliacion cargo direcciontrabajo telefonotrabajo

Esta tabla almacenara todos los datos laborales del alumno, como trabajo, dirección, teléfono, número de afiliación, cargo.

TABLA ALUM_PERSONA

alum_perso	
PK	<u>codigo</u>
	cedula nombre direccion ced_militar telefono celular fec_nacimiento e_mail estado_civil nacionalidad sexo colegio especializacion

Esta es la relación entre alumno y persona, dichas tablas están relacionadas para saber cuales son las personas que también son alumnos, por que pueden existir personas que no sean alumnos, pueden existir personas que sean solo profesor.

alum_labor	
PK	<u>codigo_a</u>
PK	<u>codigo_l</u>

Esta es la relación entre alumno y persona, dichas tablas están relacionadas para saber cuales son los alumnos que trabajan, por que pueden existir alumnos que no trabajen.

Los campos q llevan las siguientes tablas son:

1.5.1. Tabla persona

La tabla persona esta compuesta por los siguientes campos:

codigopersona, apellidos, nombres, pasaporte, cedula, cedulamilitar, sexo, fechanacimiento, nacionalidad, estadocivil, direccion, codigoubicaciongeografica, telefonocasa, otrotelefono, fax, celular, email, formacontacto, estado.

1.5.2. Tabla alumno

La tabla alumno esta compuesta por los siguientes campos:

codigoalumno, nombrepadre, nombremadre, direccionpadre, nombreconyuge, codigocolegio, codigoespecializacion, codigopersona, estado.

1.5.3. Tabla datoslaborales

La tabla datoslaborales esta compuesta por los siguientes campos:

codigoalumno, nombreempresa, direccion, telefono, numeroafiliacion, cargo, estado.

1.6. Script de las Tablas de Nuestra base de Datos

1.6.1 Tabla persona

```
CREATE TABLE persona
(
  codigopersona int4(10) NOT NULL,
  apellidos varchar(50) NOT NULL,
  nombres varchar(50) NOT NULL,
  pasaporte varchar(50),
  cedula varchar(10),
  cedulamilitar varchar(15),
  sexo varchar(1) NOT NULL,
  fechanacimiento varchar(14) NOT NULL,
  nacionalidad varchar(15) NOT NULL,
  estadocivil varchar(1) NOT NULL,
  foto varchar(100),
  direccion varchar(50) NOT NULL,
  telefonocasa varchar(10),
  otrotelefono varchar(10),
  fax varchar(10),
  celular varchar(10),
  email varchar(50),
  formacontacto varchar(20) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT persona_pkey PRIMARY KEY (codigopersona),
)
WITHOUT OIDS;
ALTER TABLE persona OWNER TO postgres;
```

1.6.2 Tabla alumno

```
CREATE TABLE alumno
(
  codigoalumno varchar(10) NOT NULL,
  nombrepadre varchar(50) NOT NULL,
  nombremadre varchar(50) NOT NULL,
  direccionpadre varchar(50) NOT NULL,
  nombreconyuge varchar(50),
  codigocolegio varchar(10) NOT NULL,
  codigoespecializacion varchar(10) NOT NULL,
  codigopersona varchar(10) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT alumno_pkey PRIMARY KEY (codigoalumno),
  CONSTRAINT alumno_codigoalumno_fkey1 FOREIGN KEY (codigoalumno)
    REFERENCES persona (codigopersona) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT alumno_codigocolegio_fkey FOREIGN KEY (codigocolegio)
    REFERENCES colegio (codigocolegio) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT alumno_codigoespecializacion_fkey
    FOREIGN KEY (codigoespecializacion)
    REFERENCES especializacion (codigoespecializacion) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE alumno OWNER TO postgres;
```

1.6.3. Tabla datos laborales

```

CREATE TABLE datoslaborales
(
  codigoalumno varchar(10) NOT NULL,
  nombreempresa varchar(50) NOT NULL,
  direccion varchar(50) NOT NULL,
  telefono varchar(10) NOT NULL,
  numeroafiliacion varchar(10) NOT NULL,
  cargo varchar(50) NOT NULL,
  estado varchar(2) NOT NULL,
  CONSTRAINT datoslaborales_pkey PRIMARY KEY (codigoalumno),
  CONSTRAINT datoslaborales_codigoalumno_fkey
  FOREIGN KEY (codigoalumno)
  REFERENCES alumno (codigoalumno) MATCH SIMPLE
  ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE datoslaborales OWNER TO postgres;

```

1.7. Diccionario de Datos.

Tabla persona

codigoalumno int4 (10).- Es una clave primaria el cual va a tener el código de la persona dicho código no puede repetido, este código va a tener relación con alumno y datos personales.

apellidos varchar (50).- Almacena el apellido de la persona.

nombres: varchar (50).- Guarda el nombre de la persona.

pasaporte: varchar (50).- Almacena el numero de pasaporte, no es un campo obligatorio.

Cedula: varchar (10).- Almacena la cedula de ciudadanía la cual tiene que ser única y validada para que no acepte un numero invalido.

Cedulamilitar varchar (15).- Almacena la cedula militar este campo es obligatorio para las personas de sexo masculino mayores de 18 años.

fechanacimiento varchar (14).- Almacena la fecha de nacimiento del usuario.

Sexo varchar (1).- Este campo sexo se solamente de 2 caracteres, donde va a guardar "M" si es masculino y "F" si es femenino.

Nacionalidad varchar (15).- Almacena la nacionalidad este es un campo parametrizable.

Estadocivil varchar (1).- Es un campo donde existen 5 opciones y va a guardar "S" si es soltero, "C" Si es casado, "V" si es viudo, "D" si es divorciado y "U" si es unión libre.

Codigoubicaciongeografica varchar (10).- Este es un campo que esta relacionado con las tablas país, provincia, ciudad y parroquia para saber la ubicación geográfica, se la usa para saber el lugar de nacimiento.

Fax varchar (10).- Este es un campo que esta relacionado con las tablas país, provincia, ciudad y parroquia para saber la ubicación geográfica, se la usa para saber el lugar de nacimiento.

Celular varchar (10).- Campo donde se va a guardar el numero de teléfono de la casa de una persona.

Email varchar (50).- Campo donde se va a guardar el numero de fax de una persona.

Formacontacto varchar (20).- En este campo tenemos la información que nos especifica cual es la vía mas rápida de contacto para el alumno.

Estado varchar (2).- Es un campo que va a tener 2 caracteres, para identificar se el alumno ha sido ingresado el campo va a ser "I", si el campo no ha sido ingresado va a tener "E".

Direccion varchar (50).- Guarda la dirección es un campo bastante largo y puede ser modificado.

telefonocasa varchar (10).- Campo donde se va a guardar el numero de teléfono de la casa de una persona.

foto bytea.- Campo donde se va a guardar cualquier imagen, en este caso se almacenará la foto de los alumnos.

1.8. Estándares de la Lógica del Negocio.

Tenemos como archivos principales para que el modulo de alumno funcione correctamente las clases DAO y BO, cada tabla esta

compuesta por un archivo llamado con el mismo nombre y acompañado por un BO o un DAO, Ejm:

La tabla Alumnos va a tener 2 archivos llamados AlumnoDAO y alumnoBO, el alumno BO es el q va a establecerla conexión de la base de datos ayudado con una tabla llamada BASE la cual va a realizar la conexión, una vez que tenemos la conexión, vamos a utilizar la clase AlumnoDAO la cual va a tener las sentencias SQL, es decir es la que va a interactuar con la Base de Datos en este caso con POSTGRES.

Además tenemos más clases DAO y BO las cuales detallamos a continuación:

AlumnoDAO	AlumnoBO
ColegioDAO	ColegioBO
ParroquiaDAO	ParroquiaBO
CiudadDAO	CiudadBO
EspecializaciónDAO	EspecializaciónBO
ProvinciaDAO	ProvinciaBO
MigrarDAO	MigrarBO
ReporteDAO	ReporteBO
SugerenciaDAO	SugerenciaBO

Para mejor apreciación vamos a citar el siguiente ejemplo:

ALUMNOBO

Este código es el que encarga de definir las clases que se están utilizando.

```
package com.cisc.bo;

import com.cisc.bean.Alumno;
import com.cisc.dao.AlumnoDao;
import com.cisc.util.BaseAlumno;
import java.sql.Connection;

import java.util.Vector;
```

Este código define al constructor de la clase.

```
public class AlumnoBO {
    private Connection con = null;
```

Este código se encarga de realizar la conexión a la Base de Datos y desde aquí va a llamar al procedimiento Guardar el cual se encuentra en la clase AlumnoDao que es el que interactúa directamente con la realizando los insert, update, etc.

```

public boolean Guardar (Alumno unAlumno) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Guardar(unAlumno);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}

```

Este código se encarga de crear la conexión para realizar la consulta a la Base de Datos por medio del método buscarCodigo que se encuentra en AlumnoDAO

```

public Alumno BuscarCodigo (String unCodigo) {
    Alumno valorDevolver = null;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.BuscarCodigo(unCodigo);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}

```

Este código es el que se encarga de generar automáticamente un código para ingresar una nueva persona este código es generado por medio del método generaCodigo que pertenece a la clase DAO.

```

public string GeneraCodigo () {
    String valorDevolver = null;
    con = BaseAlumno.getConexion();
    AlumnoDao unAlumnoDao = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDao.GeneraCodigo();
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
}

```

Este código es que se encarga de consultar y a la vez actualizar la Base de Datos todos los datos ingresados por medio del método actualiza de la clase DAO.

```

public boolean Actualizar (Alumno unAlumno) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConexion();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Actualizar(unAlumno);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
}

```

Este código es el que se encarga de eliminar de la Base de Datos todos los datos ingresados desde una Página con extensión Jsp llamada Ingreso Alumno.

El registro nose elimina directamente de la Base de la Datos, ya que existe un campo estado que es el que nos ayuda a saber si el alumno esta activo o inactivo, todo ello lo hace con la ayuda del metodo elimina de la clase DAO.

```
public boolean Eliminar (String unCodigo) {
    boolean valorDevolver = false;
    con = BaseAlumno.getConnection();
    AlumnoDao unAlumnoDAO = new AlumnoDao(con);
    try{
        valorDevolver = unAlumnoDAO.Eliminar(unCodigo);
        con.commit();
    }catch(Throwable e){
        System.out.println(e);
        try {
            con.rollback();
        }catch (Throwable ee){
            System.out.println(ee);
        }
    }
    return valorDevolver;
}
```

Una vez realizado la interconexión se llama a cada uno de los métodos los cuales se van a encargar de ingresar, actualizar, etc., todo ello se lo realiza con la ayuda de los métodos DAO que veremos a continuación.

ALUMNODAO

Este código es el que encarga de definir las clases que se están utilizando.

```
package com.cisc.dao;
import com.cisc.bean.Alumno;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.lang.String;
import java.util.Vector;
```

Este código define al constructor de la clase.

```
public class AlumnoDao {
    private Connection con = null;

    public AlumnoDao (Connection unaCon)
    {
        con = unaCon;
    }
}
```

Este código es que se encarga de guardar a la Base de Datos todos los datos ingresados en una Página con extensión Jsp llamada Ingreso Alumno.

```

String sql1 = "INSERT INTO alumno (codigoalumno,nombrepadre,
nombremadre,direccionpadre,nombreconyuge,
codigocolegio,codigospecializacion,codigopersona
,estado) values (?, ?, ?, ?, ?, ?, ?, ?, ?)";

PreparedStatement unPs1 = con.prepareStatement(sql1);
unPs1.setString(1, unAlumno.getCodigoalumno());
unPs1.setString(2, unAlumno.getNombrepadre());
unPs1.setString(3, unAlumno.getNombremadre());
unPs1.setString(4, unAlumno.getDireccionpadre());
unPs1.setString(5, unAlumno.getNombreconyuge());
unPs1.setString(6, unAlumno.getCodigocolegio());
unPs1.setString(7, unAlumno.getCodigospecializacion());
unPs1.setString(8, unAlumno.getCodigopersona());
unPs1.setString(9, unAlumno.getEstado());
int i1 = unPs1.executeUpdate();

String sql2 = "INSERT INTO datoslaborales (codigoalumno,nombreempresa,
direccion,telefono,numeroafiliacion,cargo,estado)
values (?, ?, ?, ?, ?, ?, ?)";
PreparedStatement unPs2 = con.prepareStatement(sql2);
unPs2.setString(1, unAlumno.getCodigoalumno());
unPs2.setString(2, unAlumno.getNombretrabajo());
unPs2.setString(3, unAlumno.getDirecciontrabajo());
unPs2.setString(4, unAlumno.getTelefonotrabajo());
unPs2.setString(5, unAlumno.getNumeroafiliacion());
unPs2.setString(6, unAlumno.getCargo());
unPs2.setString(7, unAlumno.getEstado());
int i2 = unPs2.executeUpdate();

```

```

String sql3 = "INSERT INTO persona (codigopersona,apellidos,nombres,
pasaporte,cedula,cedulamilitar,sexo,fechanacimiento,
nacionalidad,estadocivil,direccion,codigoubicaciongeografica,
telefonocasa,otrotelefono,fax,celular,email,formacontacto,estado)
values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
PreparedStatement unPs3 = con.prepareStatement(sql3);
unPs3.setString(1, unAlumno.getCodigoalumno());
unPs3.setString(2, unAlumno.getApellidos());
unPs3.setString(3, unAlumno.getNombres());
unPs3.setString(4, unAlumno.getPasaporte());
unPs3.setString(5, unAlumno.getCedula());
unPs3.setString(6, unAlumno.getCedulamilitar());
unPs3.setString(7, unAlumno.getSexo());
unPs3.setString(8, unAlumno.getFecha());
unPs3.setString(9, unAlumno.getNacionalidad());
unPs3.setString(10, unAlumno.getEstadocivil());
unPs3.setString(11, unAlumno.getDireccion());
unPs3.setString(12, unAlumno.getCodigoubicacion());
unPs3.setString(13, unAlumno.getTelefono1());
unPs3.setString(14, unAlumno.getTelefono2());
unPs3.setString(15, unAlumno.getFax());
unPs3.setString(16, unAlumno.getCelular());
unPs3.setString(17, unAlumno.getEmail());
unPs3.setString(18, unAlumno.getFormacontacto());
unPs3.setString(19, unAlumno.getEstado());
int i3 = unPs3.executeUpdate();

```

Este código es el que se encarga de generar automáticamente un código para ingresar una nueva persona desde una Página con extensión Jsp llamada Ingreso Alumno.

```

public String GeneraCodigo() throws Throwable
{
    String valorDevolver = new String();
    String sql = "SELECT max(codigopersona) as codigo FROM persona";
    PreparedStatement unPs = con.prepareStatement(sql);
    ResultSet unRs= unPs.executeQuery();
    while (unRs.next()){
        valorDevolver = new String();
        valorDevolver = (unRs.getString("codigo"));
    }

    unRs.close();
    unPs.close();
    int valorDevolver1 = new Integer(valorDevolver).intValue() + 1;
    valorDevolver = new String ("0000" + valorDevolver1);
    return valorDevolver;
}

```

Este código es que se encarga de consultar a la Base de Datos todos los datos ingresados desde una Página con extensión Jsp llamada Ingreso Alumno.

```

Alumno unAlumno = new Alumno();
String sql = "SELECT a.codigopersona as codpersona,apellidos,nombres,pasaporte,cedula,
cedulamilitar,sexo,fechanacimiento,nacionalidad,estadocivil,a.direccion
as direc,codigoubicaciongeografica,telefonocasa,otrotelefono,fax,celular,
email,formacontacto,\n" +
" b.codigoalumno as codalumno,nombrepadre,nombremadre,direccionpadre,nombreconyuge,
codigocolegio,codigospecializacion,\n" +
" nombreempresa,c.direccion as dir,telefono,numeroafiliacion,cargo\n" +
"From persona as a, alumno as b, datoslaborales as c\n" +
"where b.codigoalumno = a.codigopersona and b.codigoalumno = c.codigoalumno and
b.codigoalumno = ?";
PreparedStatement unPs = con.prepareStatement(sql);
unPs.setString(1,unCodigo);
ResultSet unRs= unPs.executeQuery();
while (unRs.next()){
    unAlumno.setCodigoalumno(unRs.getString("codpersona"));
    unAlumno.setNombres(unRs.getString("nombres"));
    unAlumno.setApellidos(unRs.getString("apellidos"));
    unAlumno.setPasaporte(unRs.getString("pasaporte"));
    unAlumno.setCedula(unRs.getString("cedula"));
    unAlumno.setCedulamilitar(unRs.getString("cedulamilitar"));
    unAlumno.setSexo(unRs.getString("sexo"));
    unAlumno.setFecha(unRs.getString("fechanacimiento"));
    unAlumno.setNacionalidad(unRs.getString("nacionalidad"));
    unAlumno.setEstadocivil(unRs.getString("estadocivil"));
    unAlumno.setDireccion(unRs.getString("direc"));
    unAlumno.setCodigoubicacion(unRs.getString("codigoubicaciongeografica"));
    unAlumno.setTelefono1(unRs.getString("telefonocasa"));
    unAlumno.setTelefono2(unRs.getString("otrotelefono"));
    unAlumno.setFax(unRs.getString("fax"));
    unAlumno.setCelular(unRs.getString("celular"));
    unAlumno.setEmail(unRs.getString("email"));
    unAlumno.setFormacontacto(unRs.getString("formacontacto"));
    unAlumno.setNombrepadre(unRs.getString("nombrepadre"));
    unAlumno.setCodigoalumno(unRs.getString("codalumno"));
    unAlumno.setNombremadre(unRs.getString("nombremadre"));
    unAlumno.setDireccionpadre(unRs.getString("direccionpadre"));
    unAlumno.setNombreconyuge(unRs.getString("nombreconyuge"));
    unAlumno.setCodigocolegio(unRs.getString("codigocolegio"));
    unAlumno.setTelefonotrabajo(unRs.getString("telefono"));
    unAlumno.setNumeroafiliacion(unRs.getString("numeroafiliacion"));
    unAlumno.setCargo(unRs.getString("cargo"));
}

unRs.close();
unPs.close();
return unAlumno;

```

Este código es que se encarga de consultar y a la vez actualizar la Base de Datos todos los datos ingresados desde una Pagina con extencion Jsp llamada Ingreso Alumno.

```
String sql1 = "UPDATE alumno\n" +
"set nombrepadre = ?, nombremadre = ?, direccionpadre = ?, nombrecony
"where codigoalumno = ?";
PreparedStatement unPs1 = con.prepareStatement(sql1);
unPs1.setString(1, unAlumno.getNombrepadre());
unPs1.setString(2, unAlumno.getNombremadre());
unPs1.setString(3, unAlumno.getDireccionpadre());
unPs1.setString(4, unAlumno.getNombreconyuge());
unPs1.setString(5, unAlumno.getCodigocolegio());
unPs1.setString(6, unAlumno.getCodigospecializacion());
unPs1.setString(7, unAlumno.getCodigopersona());
unPs1.setString(8, unAlumno.getCodigoalumno());
int i1 = unPs1.executeUpdate();

String sql2 = "UPDATE datoslaborales\n" +
"set nombreempresa = ?, direccion = ?, telefono = ?, numeroafiliacion
"where codigoalumno = ?";
PreparedStatement unPs2 = con.prepareStatement(sql2);
unPs2.setString(1, unAlumno.getNombretrabajo());
unPs2.setString(2, unAlumno.getDirecciontrabajo());
unPs2.setString(3, unAlumno.getTelefonotrabajo());
unPs2.setString(4, unAlumno.getNumeroafiliacion());
unPs2.setString(5, unAlumno.getCargo());
unPs2.setString(6, unAlumno.getCodigoalumno());
int i2 = unPs2.executeUpdate();

unPs1.close();
unPs2.close();
unPs3.close();

if (i1==1 && i2==1 && i3==1)
    return true;
else
    return false;
```

```

String sql3 = "UPDATE persona \n" +
"set apellidos = ?,nombres = ?,pasaporte = ?,cedula = ?,
    cedulamilitar = ?,sexo = ?,fechanacimiento = ?,
    nacionalidad = ?,estadocivil = ?,\n" +
"    direccion = ?,codigoubicaciongeografica = ?,
    telefonocasa = ?,otrotelefono = ?,fax = ?,
    celular = ?,email = ?,formacontacto = ?\n" +
"where codigopersona = ?";
PreparedStatement unPs3 = con.prepareStatement(sql3);
unPs3.setString(1, unAlumno.getApellidos());
unPs3.setString(2, unAlumno.getNombres());
unPs3.setString(3, unAlumno.getPasaporte());
unPs3.setString(4, unAlumno.getCedula());
unPs3.setString(5, unAlumno.getCedulamilitar());
unPs3.setString(6, unAlumno.getSexo());
unPs3.setString(7, unAlumno.getFecha());
unPs3.setString(8, unAlumno.getNacionalidad());
unPs3.setString(9, unAlumno.getEstadocivil());
unPs3.setString(10, unAlumno.getDireccion());
unPs3.setString(11, unAlumno.getCodigoubicacion());
unPs3.setString(12, unAlumno.getTelefono1());
unPs3.setString(13, unAlumno.getTelefono2());
unPs3.setString(14, unAlumno.getFax());
unPs3.setString(15, unAlumno.getCelular());
unPs3.setString(16, unAlumno.getEmail());
unPs3.setString(17, unAlumno.getFormacontacto());
unPs3.setString(18, unAlumno.getCodigoalumno());
int i3 = unPs3.executeUpdate();

```

Este código es el que se encarga de eliminar de la Base de Datos todos los datos ingresados desde una Página con extensión Jsp llamada Ingreso Alumno.

El registro no se elimina directamente de la Base de la Datos, ya que existe un campo estado que es el que nos ayuda a saber si el alumno está activo o inactivo.

```
public boolean Eliminar (String unCodigo) throws Throwable
{
    String sql3 = "UPDATE persona set estado = 'E' where codigopersona = ?";
    PreparedStatement unPs3 = con.prepareStatement(sql3);
    unPs3.setString(1,unCodigo);
    int i3 = unPs3.executeUpdate();

    String sql1 = "UPDATE alumno set estado = 'E' where codigoalumno = ?";
    PreparedStatement unPs1 = con.prepareStatement(sql1);
    unPs1.setString(1,unCodigo);
    int i1 = unPs1.executeUpdate();

    String sql2 = "UPDATE datoslaborales set estado = 'E' where codigoalumno = ?";
    PreparedStatement unPs2 = con.prepareStatement(sql2);
    unPs2.setString(1,unCodigo);
    int i2 = unPs2.executeUpdate();

    unPs1.close();
    unPs2.close();
    unPs3.close();

    if (i1==1 && i2==1 && i3==1)
        return true;
    else
        return false;
}
```

Todos estos procesos antes mencionados son los que se utilizaran para la realización de nuestro módulo.

CAPITULO II

2. MANUAL DE USUARIO

2.1. Descripción

El siguiente manual está orientado para el fácil y correcto manejo del Módulo de Alumnos con la finalidad de que cualquier usuario final esté en la capacidad de poder manejar el software implementado.

2.2. Objetivo

El objetivo de éste manual es que mediante éste podemos especificar de forma detallada el funcionamiento del sistema, como es el de ingresar, consultar, actualizar, eliminar y realizar reportes de alumnos, como también dar un mantenimiento a otros atributos como es el de ingreso de mas provincias, ciudades, parroquias, colegios y especializaciones, además de realizar migraciones o respaldos históricos de la base de datos.

2.3. Detalle de Acciones y Respuestas del Sistema

A continuación se explicará el uso correcto del módulo de alumno del Sistema Académico:

2.3.1 Pantalla de Bienvenida General.- Para acceder al Sistema Académico Integrado, la persona deberá ingresar con un usuario y una clave, tal como se muestra en la (fig. 2-1).

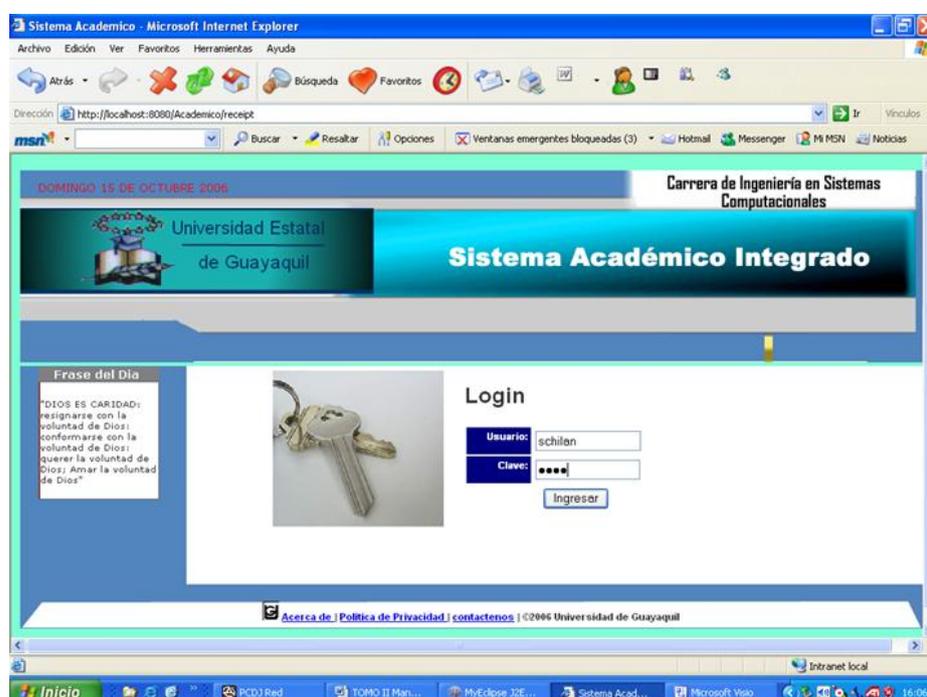


Fig. 2-1. Interfaz de Bienvenida al Sistema Integrado de la Carrera de Ingeniería en Sistemas Computacionales.

Luego le aparecerá un menú sobre el lado izquierdo de la pantalla las distintas opciones que puede realizar este sistema, así como se muestra en la (fig. 2-2).

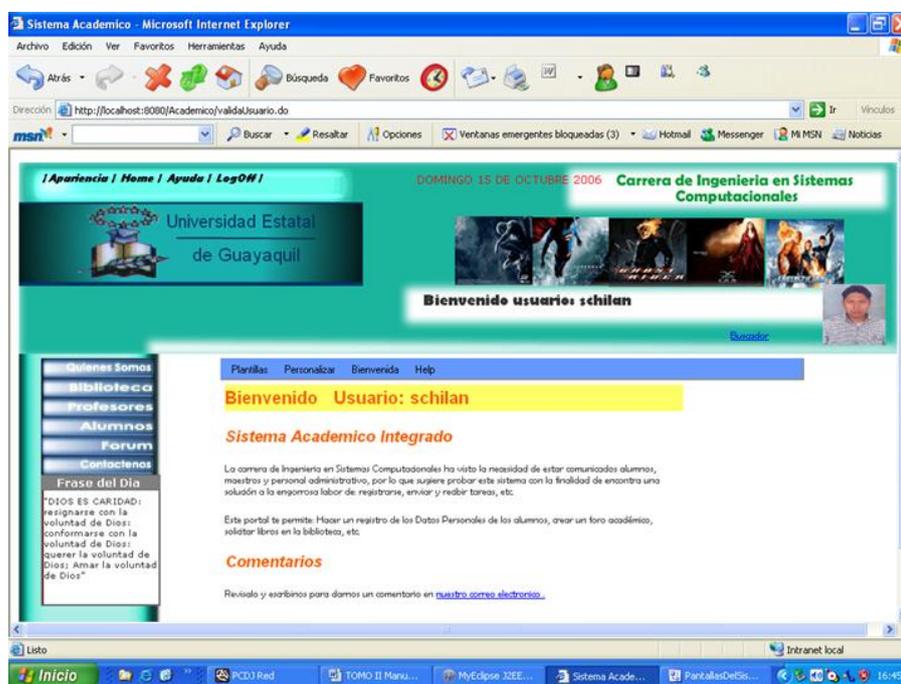


Fig. 2-2. Interfaz de Bienvenida y Menú Principal del Sistema Integrado.

Estando en el sistema podremos seleccionar específicamente del menú desplegado en la parte izquierda Alumnos para ingresar a nuestro módulo. Al hacer link se mostrará una interfaz de Bienvenida al Módulo de Alumnos y un menú en la parte superior sobre las distintas opciones que se pueden realizar sobre los alumnos de la carrera, véase en la (fig. 2-3).



Fig. 2-3. Interfaz de Bienvenida al Módulo de Alumnos con su respectivo Menú Principal.

En este menú se podrá realizar un ingreso de los alumnos al sistema, también una consulta por el apellido, por el código o por el número de cédula de los alumnos ya registrados, una modificación o actualización de sus datos, una eliminación del sistema de un alumno, también se pueden subir y descargar exámenes de las materias ya tomadas, de la misma manera se pueden hacer con los syllabus de las materias, se puede hacer una visualización de diversos reportes sobre los alumnos pertenecientes a la Carrera de Ingeniería en Sistemas y también realizar sugerencias de nuestro sistema.

2.3.1.1 Ingreso de un Alumno.- Al escoger del menú la primera opción aparecerá una nueva pantalla en donde podrá ingresar un nuevo alumno al sistema, como se muestra en la (fig. 2-4).



Fig. 2-4. Interfaz de Ingreso de Alumno.

Entre los campos validados tenemos el código de manera se genera automáticamente y no tendrá que ingresarlo, los campos con asteriscos indicarán los campos que necesariamente tendrán que llenarse, el campo de la cédula está validado de manera que cuando se ingrese éste tiene que ser un número válido porque de lo contrario le aparecerá un mensaje de error, luego ingresa los apellidos, los nombres, el sexo, si es masculino podrá ingresar el numero de la cédula

militar por que de lo contrario automáticamente se desactivará la caja de texto, el pasaporte, la dirección, teléfonos, celular, fax, la nacionalidad, el estado civil, el colegió donde se graduó, la forma de contacto que es con quien se lo pueda contactar, si el usuario se decide a guardar solamente esos datos ingresados aparecerá en la parte superior de la pagina un mensaje de todos los errores cometidos, ya sea porque no ingresó correctamente la cédula o porque no ingresó todos los campos obligatorios, tal como se muestra en la (fig. 2-5).

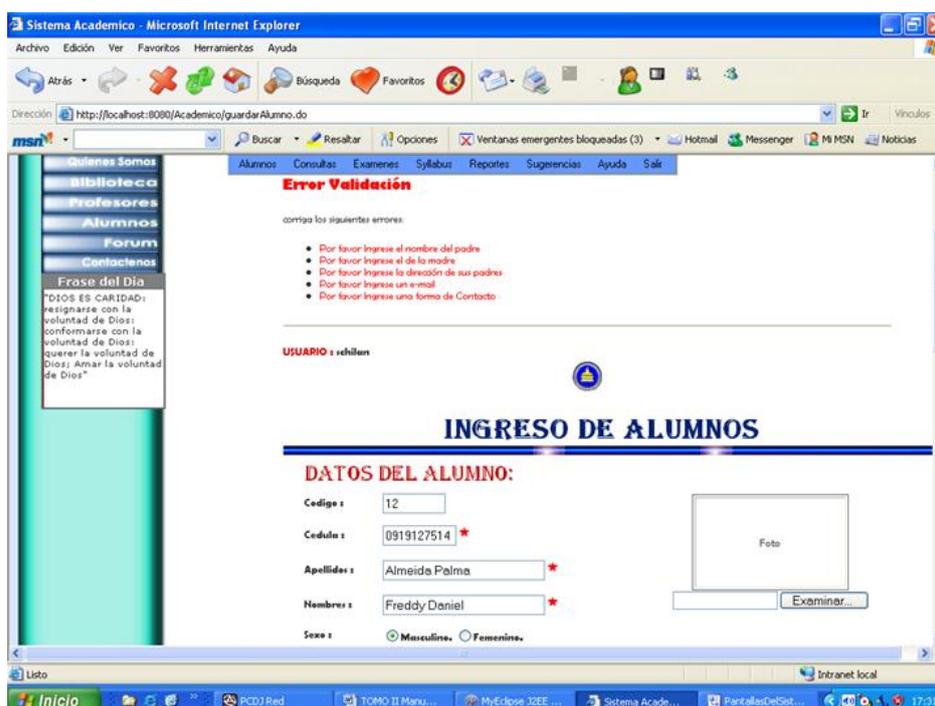


Fig. 2-5. Interfaz de Ingreso cuando no se han completado todos los campos obligatorios.

Se continúa ingresando la provincia, aquí cuando se seleccione una provincia automáticamente se llenará el combo de las ciudades de esa provincia que seleccionó, de igual manera pasa con las parroquias se llenará el combo cuando seleccione una ciudad específica, podrá seleccionar su fecha de nacimiento, tendrá que llenar los campos con el nombre de sus padres, la dirección donde residen, su e-mail, en caso de ser casado el nombre de su cónyuge, luego son opcionales los Datos Laborales aquí podrá ingresar el nombre de la empresa, la dirección, el teléfono, el número de afiliación al Seguro Social y el cargo en caso de que tenga algún trabajo; si lleno bien con los datos especificados anteriormente podrá seleccionar el botón Guardar como se muestra en la (fig. 2-6).

The screenshot shows a web browser window titled 'Sistema Academico - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/Academico/guardarAlumno.do'. The form contains the following fields:

- Nombre del Padre : Daniel Almeida *
- Nombre de la Madre : Cecilia Palma *
- Dirección de los Padres : Sauces 8 *
- E-mail : fdanielalmeida@hotmail.com *
- Nombre del Cónyuge : (empty)

DATOS LABORALES:

- Nombre de la Empresa : Coop. Gral. Vernaza
- Dirección del Trabajo : Cuenca y Cacique Alvarez
- Teléfono del Trabajo : 2400120
- Numero de Afiliación : (empty)
- Cargos : Técnico

(*) Todos los campos con asteriscos son obligatorios

Guardar

Fig. 2-6. Interfaz de Ingreso mostrando el botón de Guardar.

Luego de esto le aparecerá una nueva pantalla en donde le especifica que sus datos han sido guardados con éxito, tal como se muestra en la (fig. 2-7).

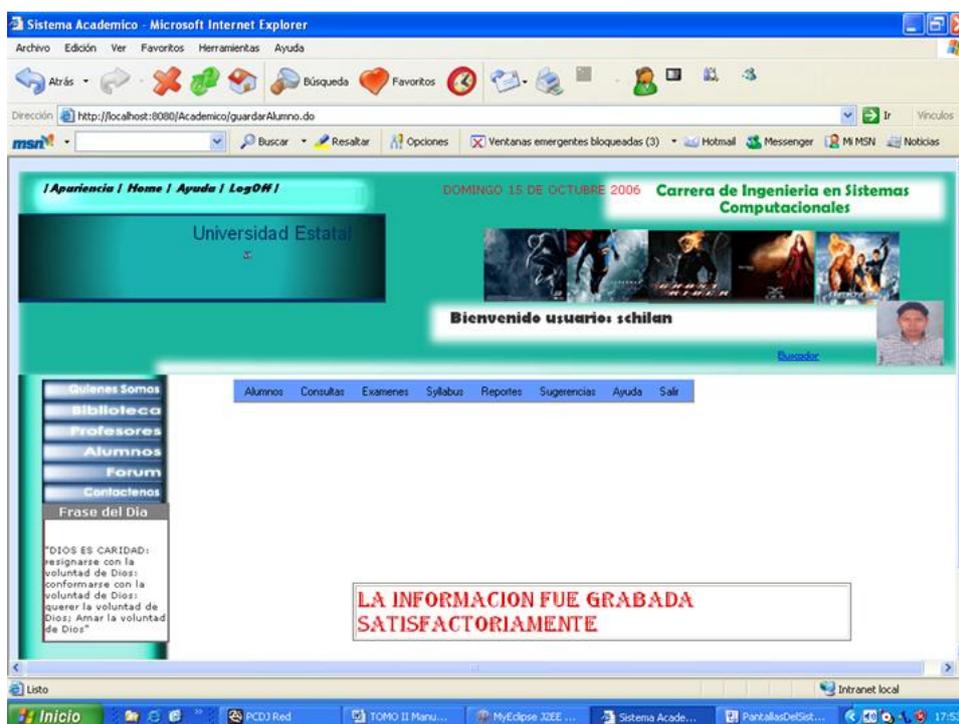


Fig. 2-7. Interfaz de Confirmación de datos guardados satisfactoriamente.

2.3.1.2 Consulta de un Alumno por el Apellido.- Si seleccionamos del menú Consulta por Apellido aparecerá una pantalla en donde podrá ingresar entre asteriscos un apellido completo o sólo una letra y al presionar en el botón Buscar nos mostrará un listado de todos los apellidos encontrados en la base que tengan relación con la palabra ingresada, tal como se muestra en la (fig. 2-8).

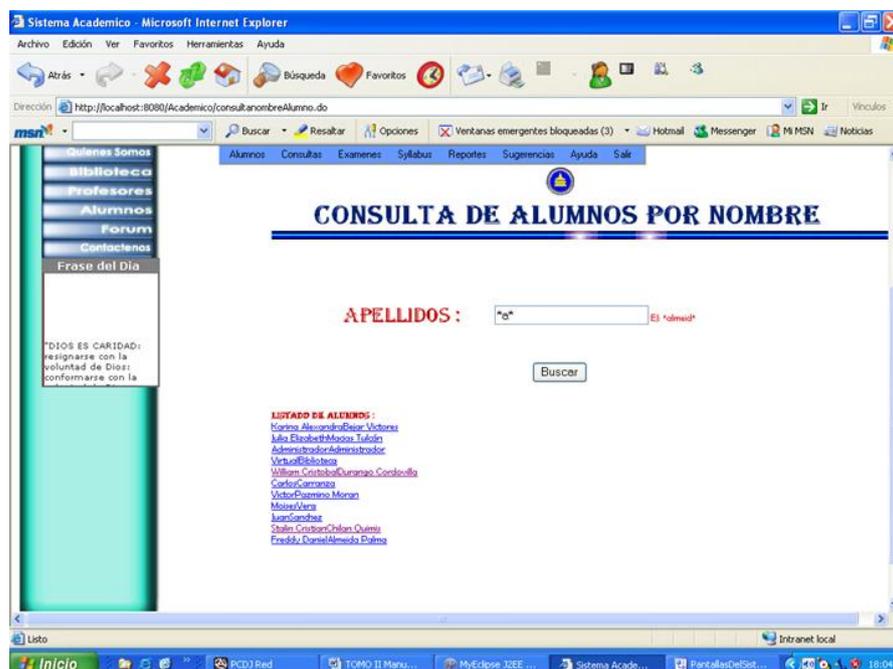


Fig. 2-8. Interfaz de Consulta por nombre de Alumno.

Luego para visualizar todos los datos referentes a un alumno determinado lo que se hace es un click sobre cualquier nombre de la lista encontrada, ya que cada uno de ellos se muestran de manera de link y ésta nos llevará a una nueva ventana presentándonos todos sus datos personales, tal como se muestra en la (fig. 2-9).

The screenshot shows a web browser window titled 'Sistema Academico - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/Academico/BusquedaDetalle?codigo=126'. The page content includes a navigation menu at the top with links like 'Alumnos', 'Consultas', 'Exámenes', 'Syllabus', 'Reportes', 'Sugerencias', 'Ayuda', and 'Salir'. The main heading is 'CONSULTA DE ALUMNOS'. Below this, the section 'DATOS DEL ALUMNO:' lists the following information:

Codigo :	12	
Cedula :	0919127514	
Apellidos :	Almeida Palma	
Nombre :	Freddy Daniel	
Paraparte :	0919127514	
Sexo :	Masculino	Ced. Militar : 180009201502
Fecha de Nacimiento :	02-Ago-1980	
Dirección :	Sauces 9 Mez 476 V 7	Celular : 092549226
Telefono 1 :	2247547	Fax :
Telefono 2 :		
Nacionalidad :	Ecuatoriana	Especialización : INFORMATICA
Estado Civil :	Soltero (a)	
Colegio :	GUILLERMO ROHDE AROSEMENA	Forma de Contacto : Trabajo
Provincia :	QUIAVAS	
Ciudad :	QUIAVAQUIL	

The sidebar on the left contains a 'Frase del Dia' box with the text: 'DIOS ES CARIDAD: resignarse con la voluntad de Dios; conformarse con la voluntad de Dios; querer la voluntad de Dios; Amar la voluntad de Dios'.

Fig. 2-9. Interfaz en donde se muestran los datos encontrados de un Alumno.

Una vez que se mostraron los datos podemos regresar a la página principal haciendo un click sobre el icono que se encuentra en la parte superior de la pantalla o sino puede realizar cualquier otra acción seleccionando alguno del menú principal.

2.3.1.3 Consulta de un Alumno por Código.- Si seleccionamos del menú Consulta por código aparecerá una pantalla en donde podrá ingresar el código correspondiente a cualquier alumno, si el código es válido y al presionar el botón Buscar nos presentará todos los datos correspondientes a ese alumno tal como se muestra en la (fig. 2-10).

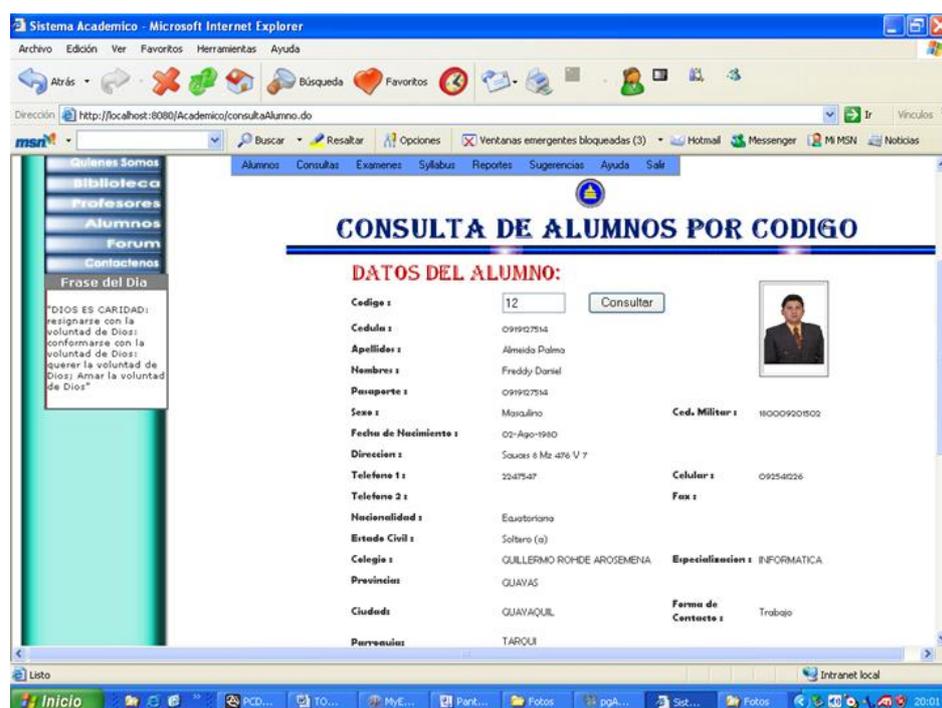


Fig. 2-10. Interfaz de Consulta por Código de Alumno.

2.3.1.4 Consulta de un Alumno por la Cédula.- Si seleccionamos del menú Consulta por Cédula aparecerá una pantalla en donde podrá ingresar la cédula correspondiente a cualquier alumno, si la cédula es válida, se encuentra en la Base y al presionar el botón Buscar nos presentará todos los datos correspondientes a ese alumno tal como se muestra en la (fig. 2-11).



Fig. 2-11. Interfaz de Consulta por la Cédula de un Alumno.

2.3.1.5 Modificación de un Alumno por Código.- Al escoger del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de alumno, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará todos los datos de ese alumno, tal como se muestra en la (fig. 2-12).



Fig. 2-12. Interfaz de Modificación de los Datos de un Alumno.

Pero no todos los datos se podrán actualizar solo algunos entre estos mencionamos los siguientes: la dirección, los teléfonos, el celular, la forma de contacto, la dirección donde residen sus padres, el e-mail, de la misma manera podrá actualizar algunos de sus datos laborales como es el nombre de la empresa donde trabaja, la dirección, el número de teléfono y el cargo en que se desempeña, así como se muestra en la (fig. 2-13).

Sistema Académico - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost:8080/Academico/actualizacionAlumno.do>

msn

Nombre de la Madre : Cecilia Palma *

Dirección de los Padres : Sauces 8 *

E-mail : fdanielalmeida@hotmail.com

Nombre del Conyuge :

DATOS LABORALES:

Nombre de la Empresa : Coop. Gral. Verneza

Dirección del Trabajo : Cuenca y Cacique Alvarez

Telefono del Trabajo : 2400120

Numero de Afiliación :

Cargos : Técnico

(*) Todos los campos con asteriscos son obligatorios

Actualizar

Intranet local

Inicio PC03 Red TOMO II ... MyEclipse ... Pantallas0... Fotos Sistema A... 20:16

Fig. 2-13. Interfaz de Modificación de los Datos de un Alumno en la parte del botón de Actualizar.

Una vez que se ha terminado de actualizar todos los datos, al presionar el botón de Actualizar si alguno de los campos obligatorios se encuentra vacío aparecerá en la parte superior los correspondientes mensajes de errores, caso contrario se mostrará una ventana en donde nos indicará que la información fue actualizada correctamente, tal como se muestra en la (fig. 2-14).

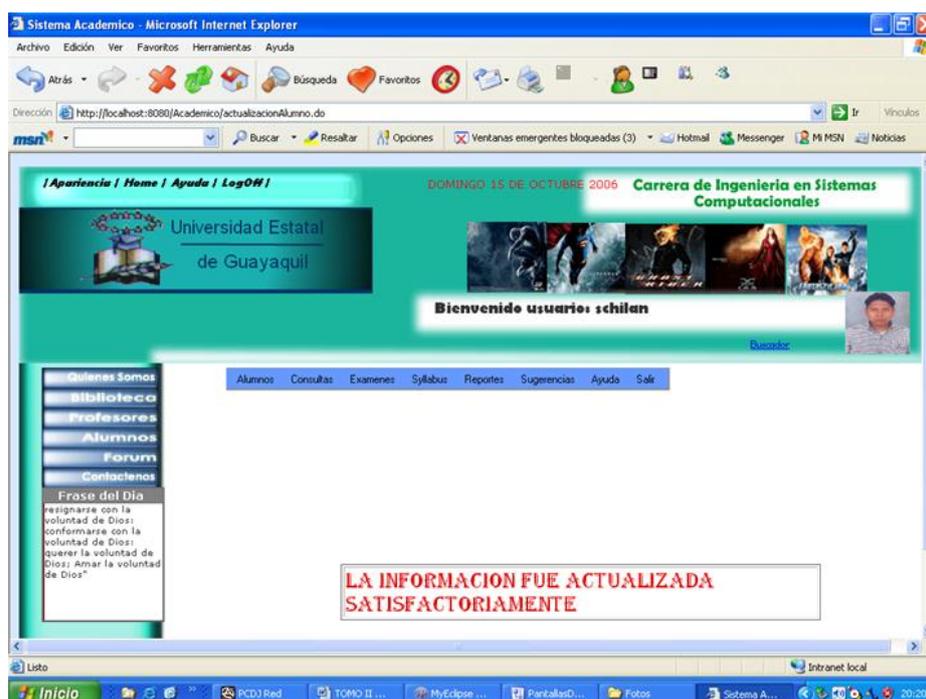


Fig. 2-14. Interfaz de Confirmación de datos actualizados correctamente.

2.3.1.6 Eliminación de un Alumno por Código.- Al escoger del menú principal eliminación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de alumno, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará todos los datos de ese alumno, tal como se muestra en la (fig. 2-15).

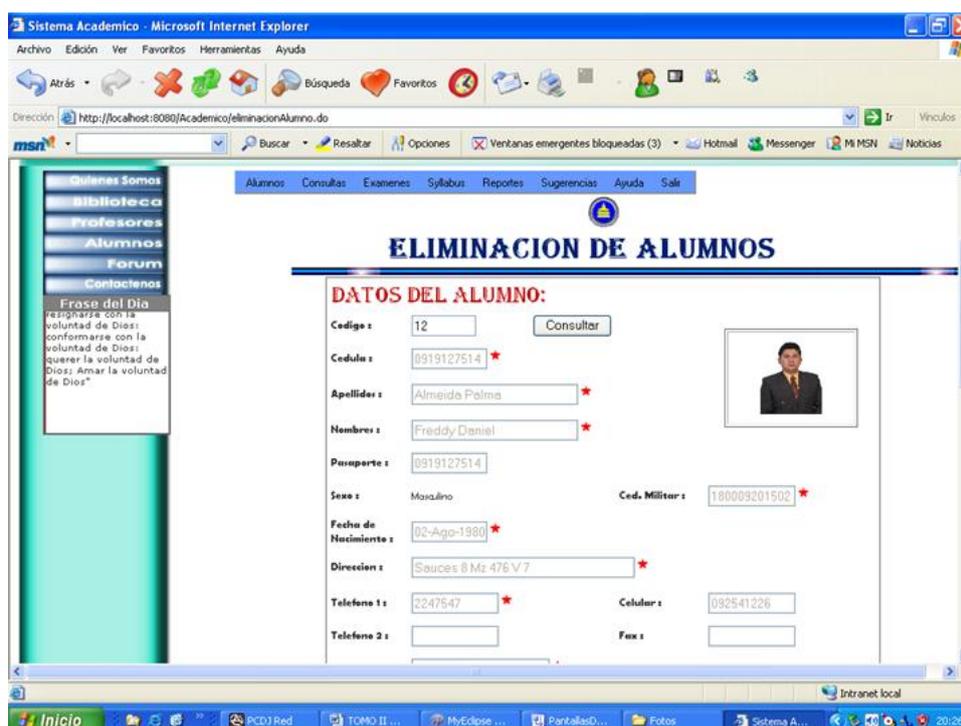


Fig. 2-15. Interfaz de Eliminación de los Datos de un Alumno.

Revisando la información del alumno en la parte inferior de la pantalla se encontrará el botón Eliminar, tal como se muestra en la (fig. 2-16).

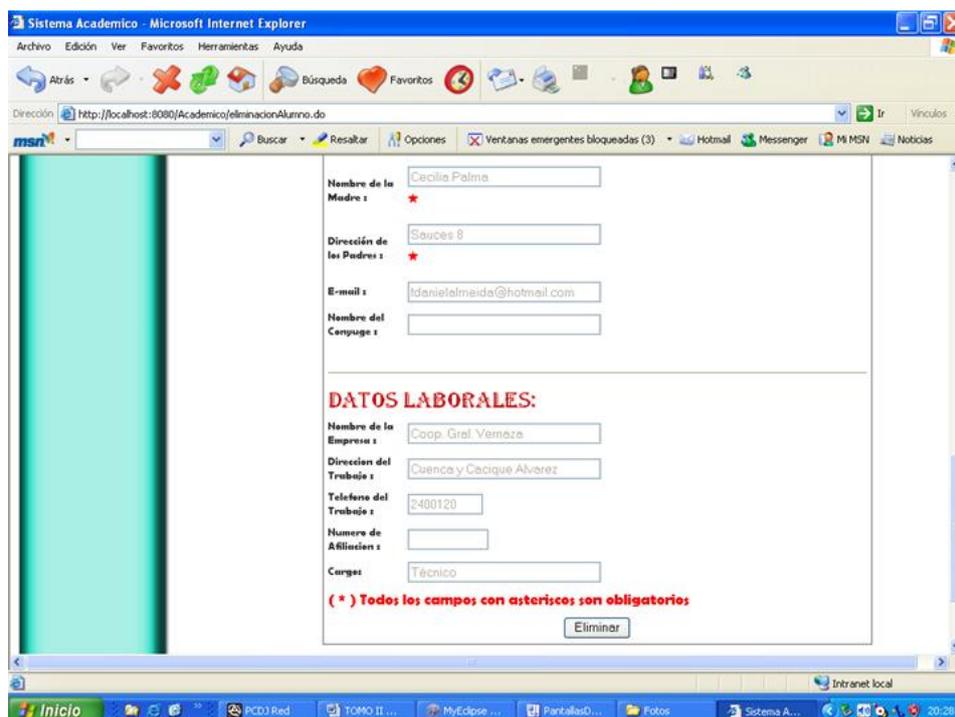


Fig. 2-16. Interfaz de Eliminación de los Datos de un Alumno en la parte del botón de Eliminar.

Al presionar el botón eliminar todos los datos correspondientes a ese alumno serán eliminados de la Base de Datos pero solamente de una manera lógica, luego de esto nos mostrará una pantalla de confirmación de que la acción fue realizada con éxito.

2.3.1.7 Subir Exámenes.- Al escoger del menú principal Subir exámenes, éste nos mostrará una interfaz en donde permitirá seleccionar de un combo alguna materia y una caja de texto en donde podrá direccionar algún documento que se encuentren exámenes de esa materia de periodos lectivos, tal como se muestra en la (fig. 2-17).

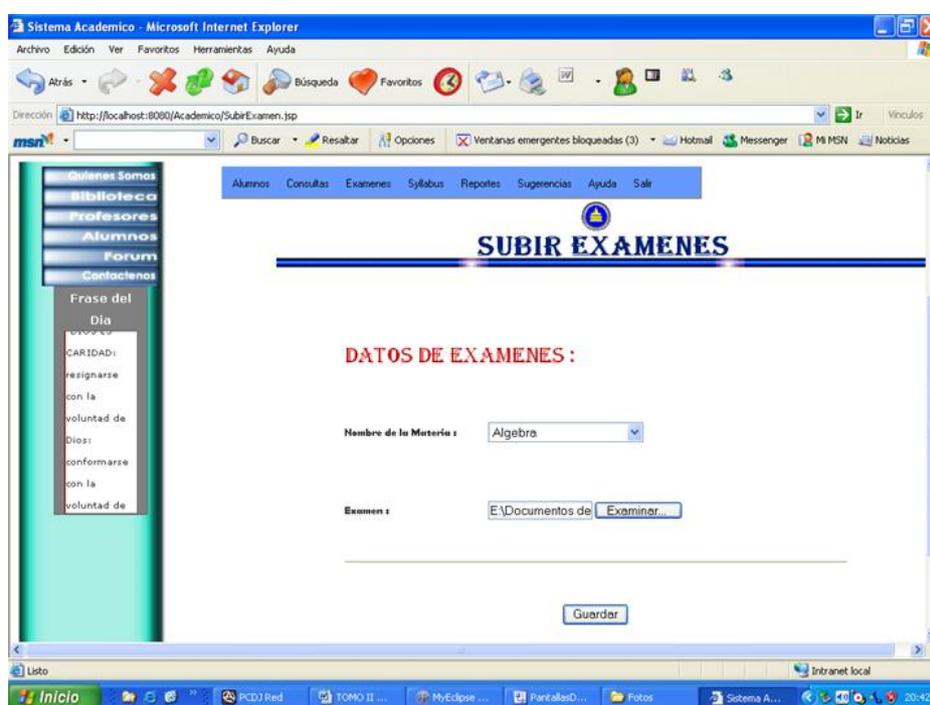


Fig. 2-17. Interfaz para Subir Exámenes.

Al presionar el botón Guardar éste se almacenará en la base de datos y nos mostrará un mensaje de que el documento ha sido guardado correctamente, tal como se muestra en la (fig. 2-18).

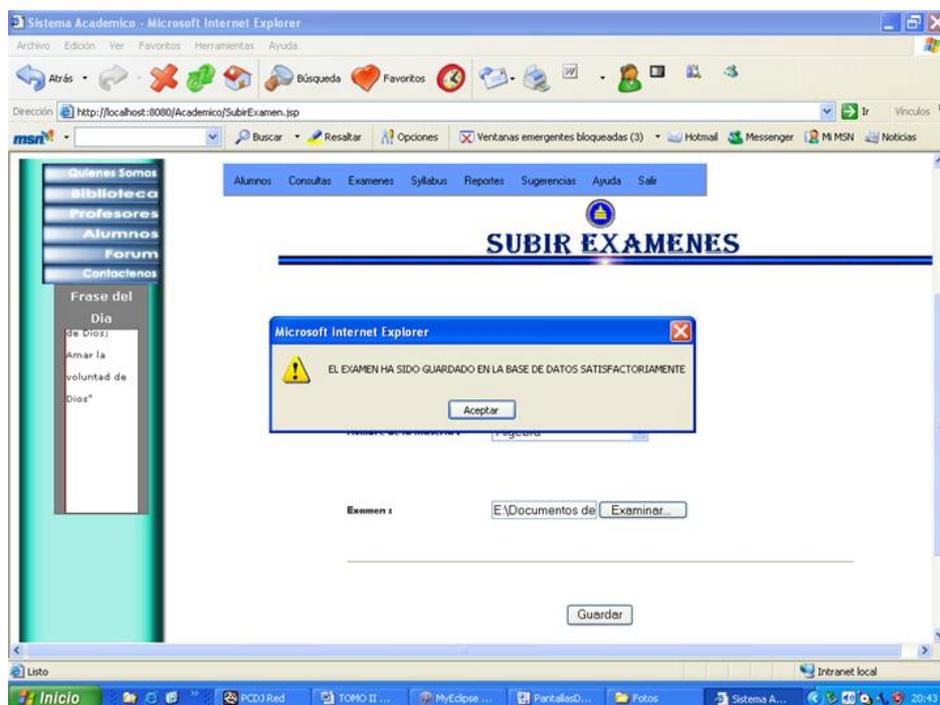


Fig. 2-18. Alerta de confirmación de Examen subido a la base de datos .

2.3.1.8 Descargar Exámenes.- Al escoger del menú principal Descargar exámenes, éste nos mostrará una interfaz en donde permitirá seleccionar de un combo alguna materia específica y poder bajar de la base cualquier documento que esté en la base, véase en la (fig. 2-19).

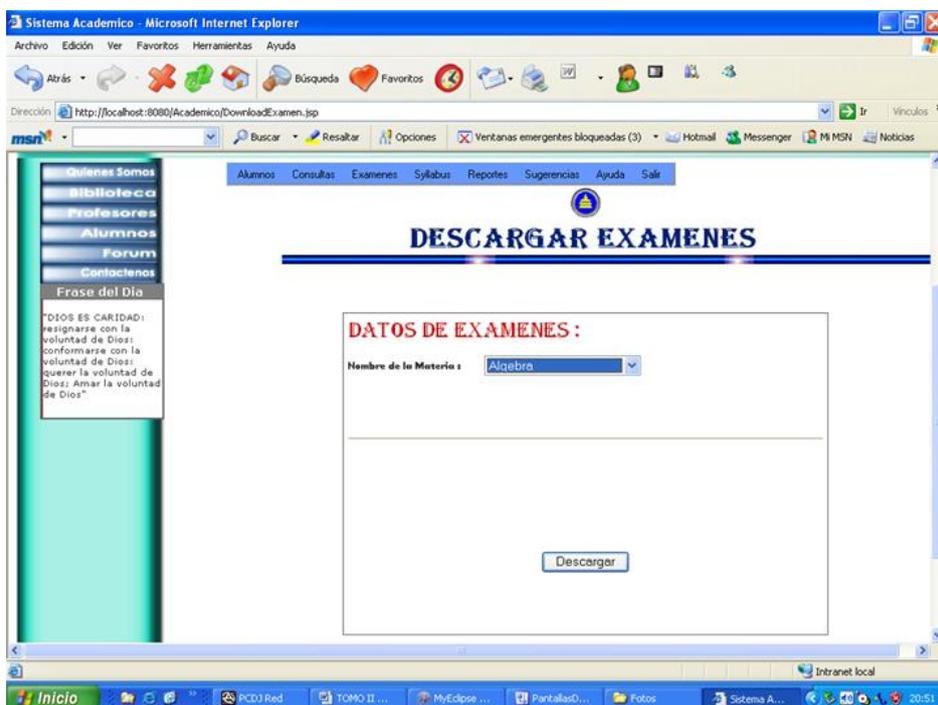


Fig. 2-19. Interfaz para Descargar Exámenes.

Al presionar el botón Descargar éste será bajado en una ruta específica, y así de ésta manera se podrá visualizar el documento, mostrándonos un mensaje de que el documento ha sido descargado con éxito, véase en la (fig. 2-20).

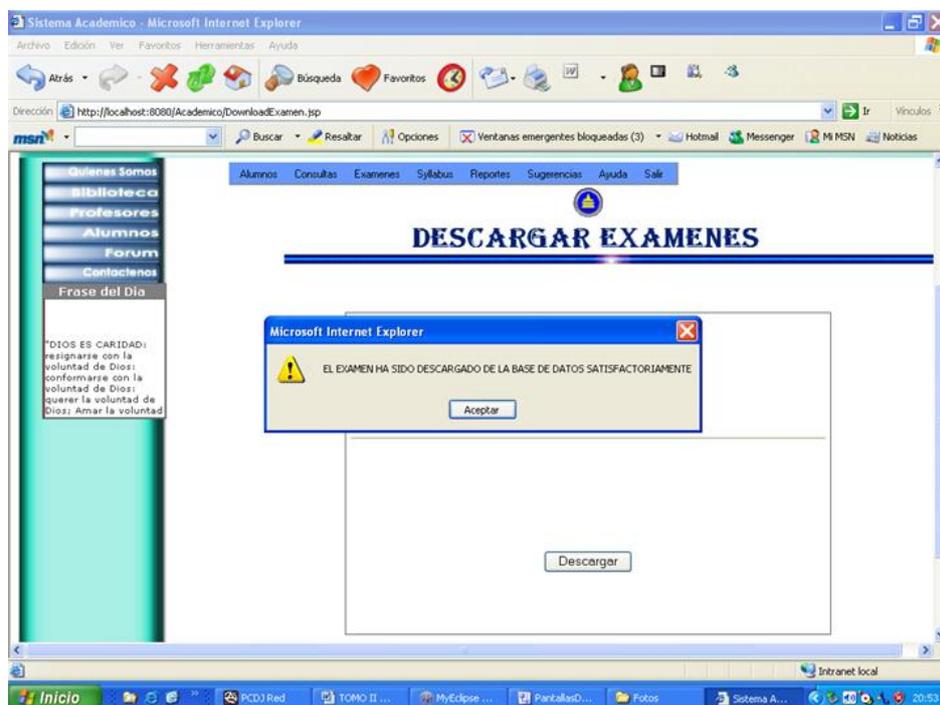


Fig. 2-20. Alerta de confirmación de Examen Descargado de la base de datos .

2.3.1.9 Subir Syllabus.- Al escoger del menú principal Subir Syllabus, éste nos mostrará una interfaz en donde permitirá seleccionar de un combo alguna materia y una caja de texto en donde podrá direccionar algún documento que se encuentren syllabus, cabe recalcar que éstas son ayudas o guías de esa materia seleccionada, tal como se muestra en la (fig. 2-21).

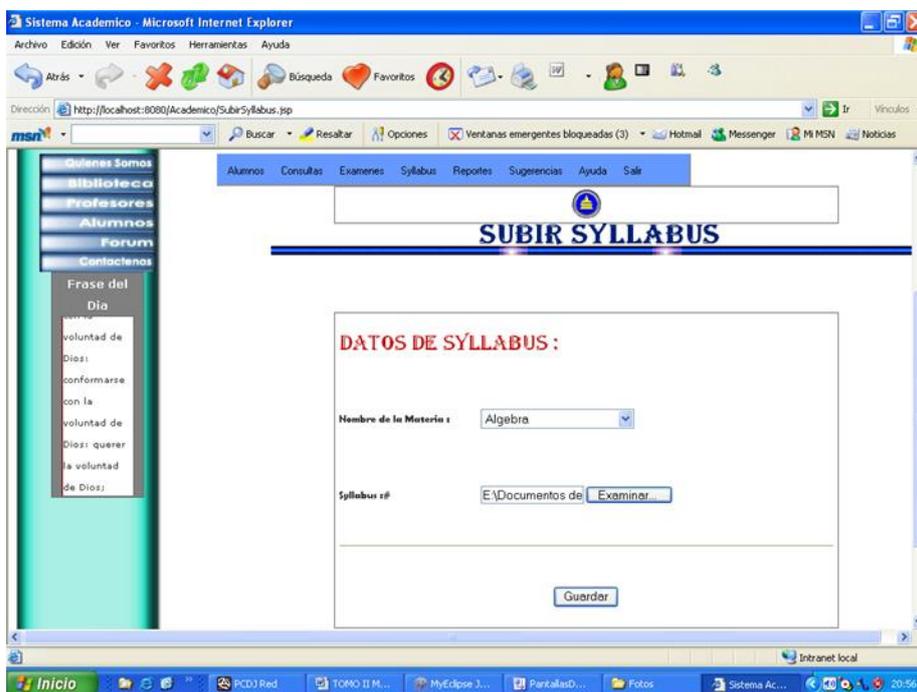


Fig. 2-21. Interfaz para Subir Syllabus.

Al presionar el botón Guardar éste se almacenará en la base de datos y nos mostrará un mensaje de que el documento ha sido guardado correctamente, tal como se muestra en la (fig. 2-22).

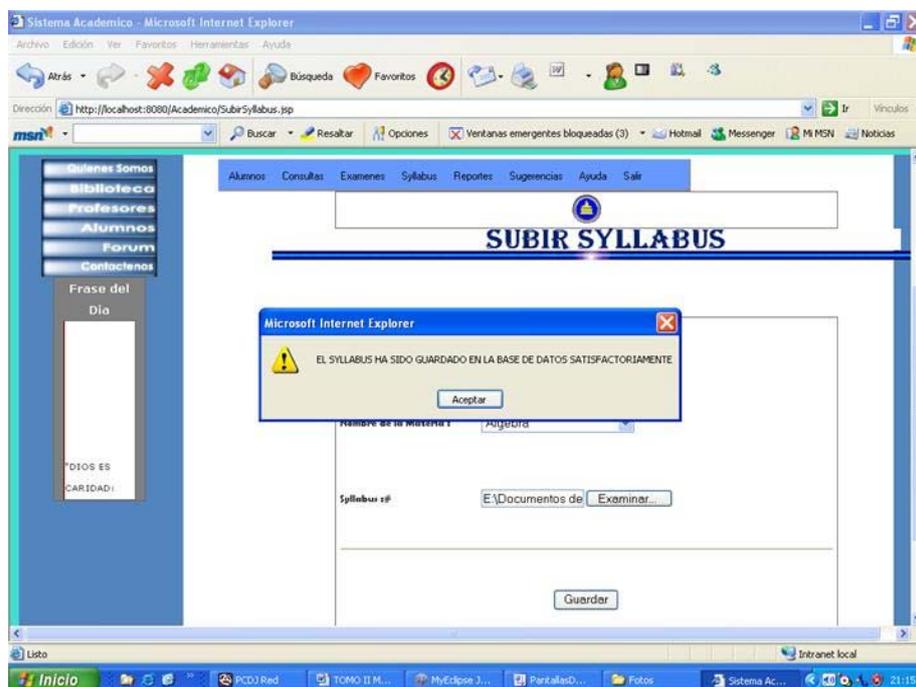


Fig. 2-22. Alerta de confirmación de Syllabus subido a la base de datos .

2.3.1.10 Descargar Syllabus.- Al escoger del menú principal Descargar syllabus, éste nos mostrará una interfaz en donde permitirá seleccionar de un combo alguna materia específica y poder bajar de la base cualquier documento que esté en la base, véase en la (fig. 2-23).

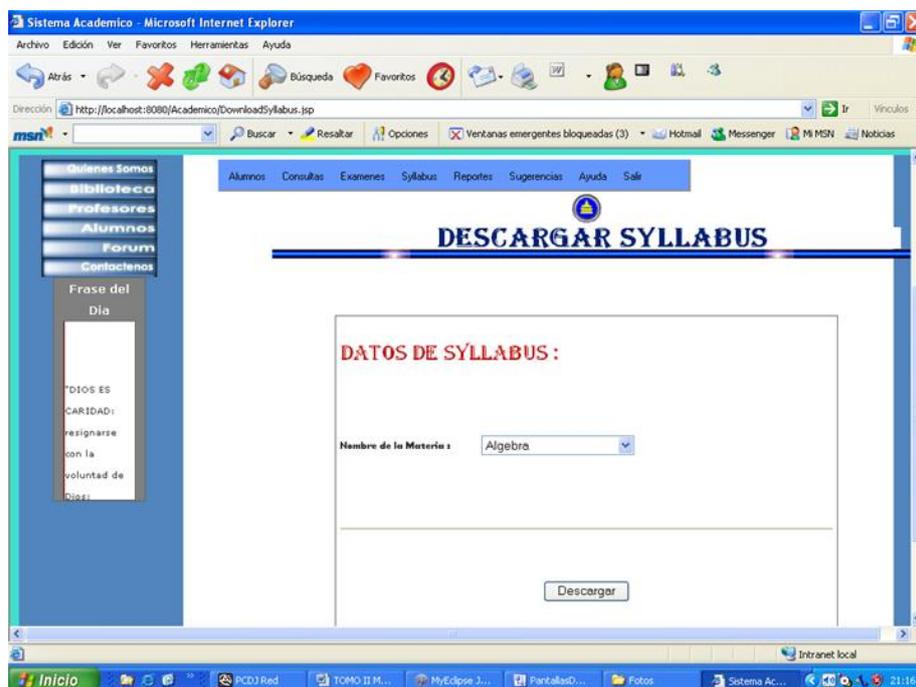


Fig. 2-23. Interfaz para Descargar Syllabus.

Al presionar el botón Descargar éste será bajado en una ruta específica, y así de ésta manera se podrá visualizar el documento, mostrándonos un mensaje de que el documento ha sido descargado con éxito, véase en la (fig. 2-24).

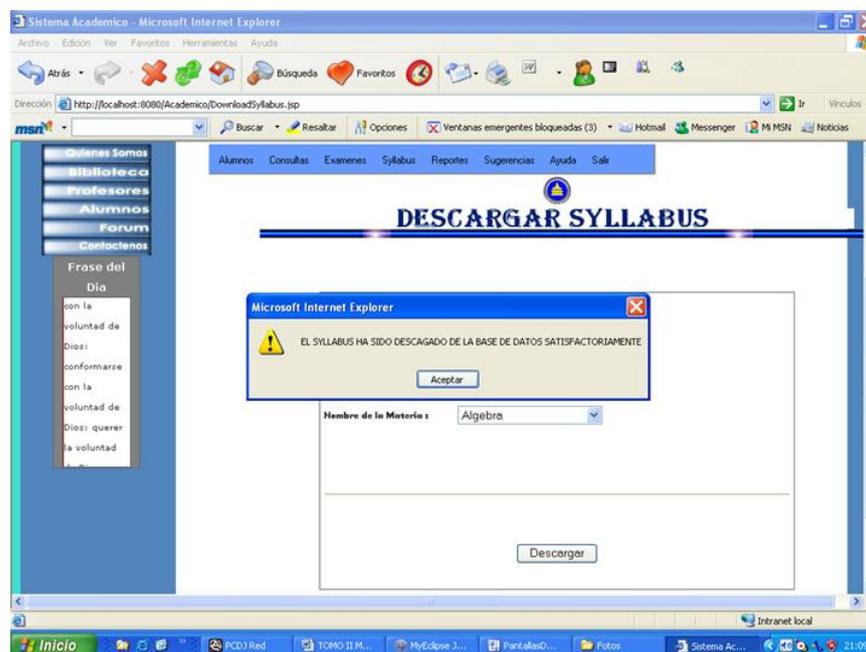


Fig. 2-24. Alerta de confirmación de Syllabus Descargado de la base de datos .

2.3.1.11 Generación de Reportes para Alumnos.- Al escoger del menú principal Reportes, éste nos mostrará una pantalla en donde permitirá crear reportes de alumnos ya sean por nombres, por código, a partir de una fecha específica o por el estado es decir si se encuentra ingresado o eliminado, tal como se muestra en la (fig. 2-25).

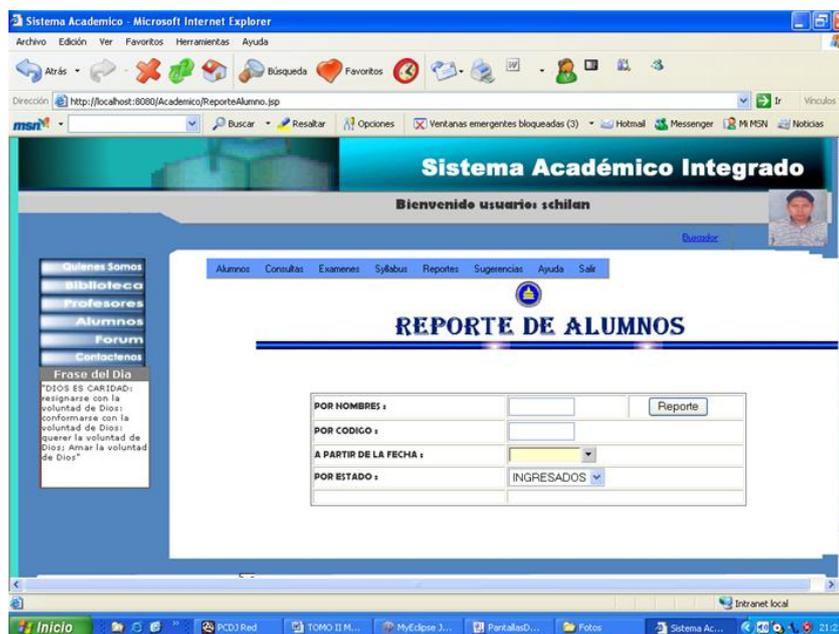


Fig. 2-25. Pantalla para Reportes de Alumnos.

Si se desea generar un reporte por ejemplo por el estado o sea si el alumno está activo o eliminado, lo que tiene que hacer es escoger del combo por estados escoger la opción eliminados, presiona el botón de Reporte y automáticamente se crea un reporte listo para imprimirse.

2.3.2. Mantenimiento de Atributos.- En caso de que el usuario necesite de información adicional en los combos de los datos de los alumnos ya sean estos como las provincias, ciudades, parroquias, colegios y especializaciones deberá de acercarse al administrador del sistema para que éste le ingrese la información que sea necesaria, pues solamente él tendrá los permisos necesarios para hacerlo.

CAPITULO III

3. MANUAL DEL ADMINISTRADOR

3.1. Descripción

Este manual está orientado para el fácil y correcto manejo del Módulo de Alumnos en lo que tiene que ver con las parametrizaciones de algunos de los atributos, con la finalidad de que cualquier administrador del sistema esté en la capacidad de poder manejar el software implementado.

3.2. Objetivo

El objetivo de éste manual es que mediante éste podemos especificar de forma detallada el funcionamiento del sistema, en lo que tiene que ver con el mantenimiento de algunos atributos como es el de ingreso, consulta, modificación y eliminación de provincias, ciudades, parroquias, colegios y especializaciones, además de realizar migraciones o respaldos históricos de la base de datos.

3.3. Detalle de Acciones y Respuestas del Sistema

A continuación se explicará el uso correcto del módulo de alumno del Sistema Académico en la parte que le compete al administrador:

3.3.1 Pantalla de Bienvenida General.- Cuando el administrador desea ingresar información adicional éste deberá hacerlo con un Usuario y su Clave respectiva desde la interfaz principal, al ingresar al sistema integrado y seleccionar del menú desplegado a la izquierda, cuando selecciona alumnos ahora se mostrará en la pantalla otro menú en donde el administrador podrá manipular la información de las provincias, ciudades, parroquias, colegios y de las especializaciones, tal como se muestra en la (fig. 3-1).



Fig. 3-1. Interfaz de bienvenida y menú principal de las Provincias, Ciudades, Parroquias, Colegios y Especializaciones.

3.3.2 Ingreso de una Provincia.- Cuando el administrador desea adicionar una nueva provincia debe del menú provincias, escoger Ingreso y se mostrará una interfase, así como se muestra en la (fig. 3-2).

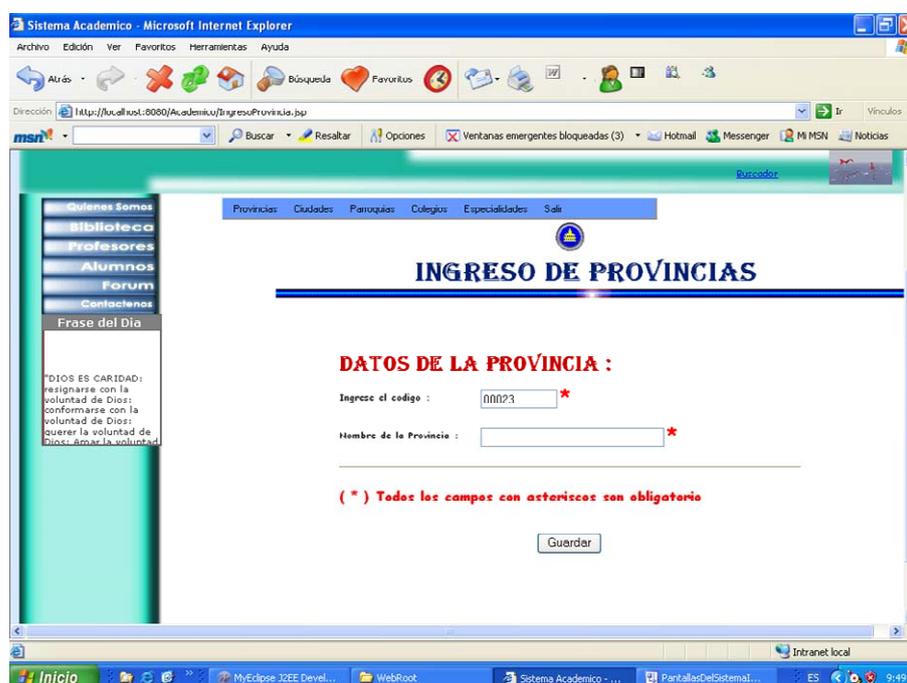


Fig. 3-2. Interfaz de Ingreso de las Provincias.

Si el administrador presiona el botón de Guardar sin haber llenado el campo obligatorio aparecerá en la pantalla el mensaje de error correspondiente, tal como se muestra en la (fig. 3-3).

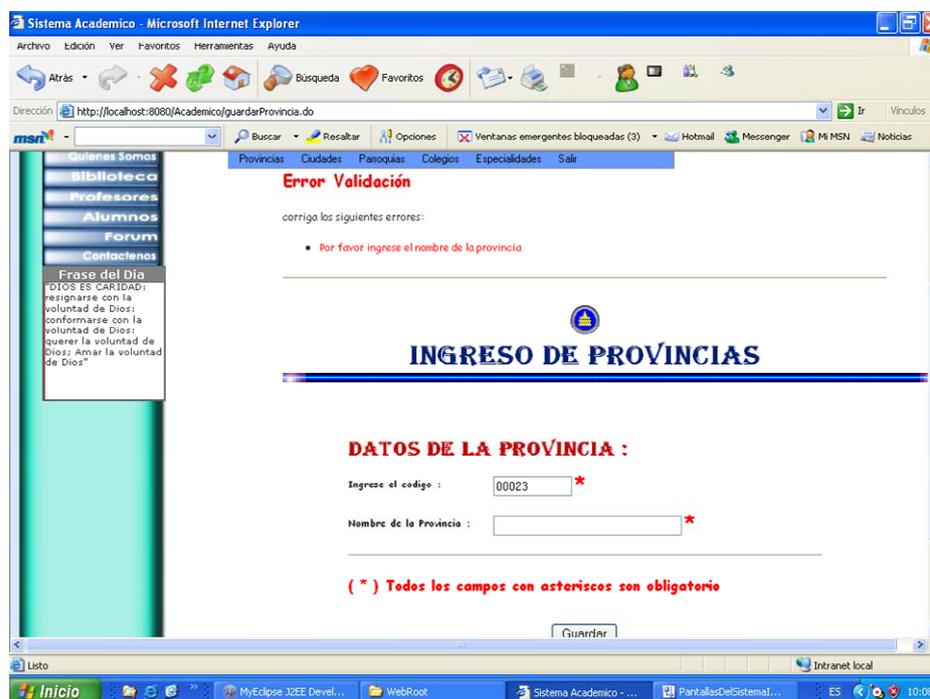


Fig. 3-3. Interfaz de Ingreso cuando no se ha completado los campos obligatorios presenta mensajes de error.

Si le ha ocurrido eso necesariamente tendrá que llenar los campos obligatorios, tal como se muestra en la (fig. 3-4).

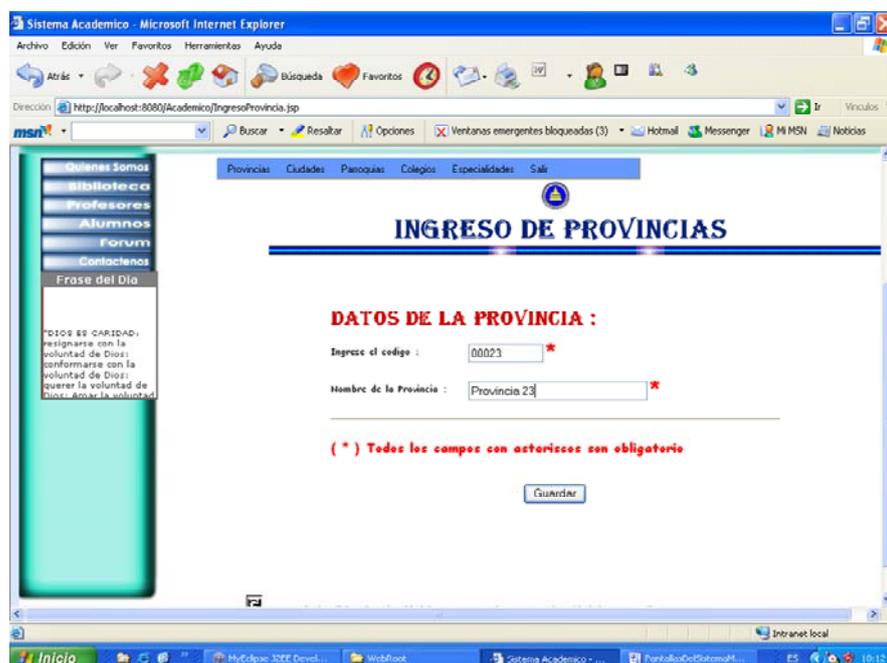


Fig. 3-4. Interfaz de Ingreso de las Provincias con datos.

Si ha cumplido con ello, deberá de presionar el botón Guardar inmediatamente aparecerá una nueva pantalla en donde se mostrará información que se ha almacenado en la base de datos, tal como se muestra en la (fig. 3-5).

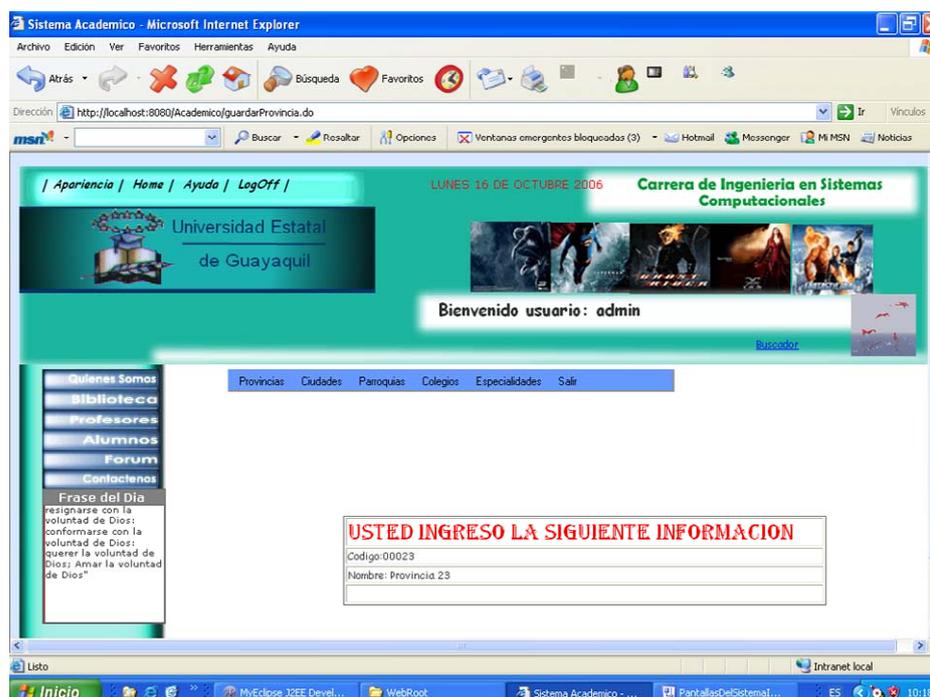


Fig. 3-5. Interfaz de Confirmación de que los datos han sido ingresados con éxito.

3.3.3. Consulta de una Provincia.- Si seleccionamos del menú Consulta aparecerá una pantalla en donde podrá ingresar el código de la provincia, al presionar el botón Buscar aparecerá el nombre que le corresponde a esa provincia, tal como se muestra en la (fig. 3-6).

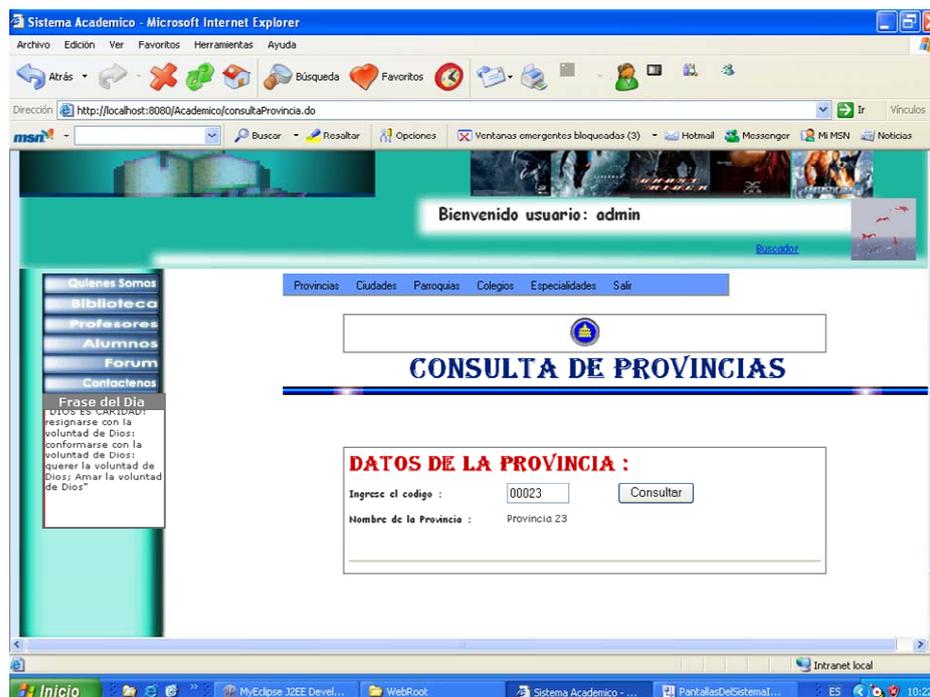


Fig. 3-6. Interfaz de Consulta por código de una Provincia.

3.3.4. Modificación de una Provincia.- Al escoger del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de provincia, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará el nombre correspondiente, véase en la (fig. 3-7).

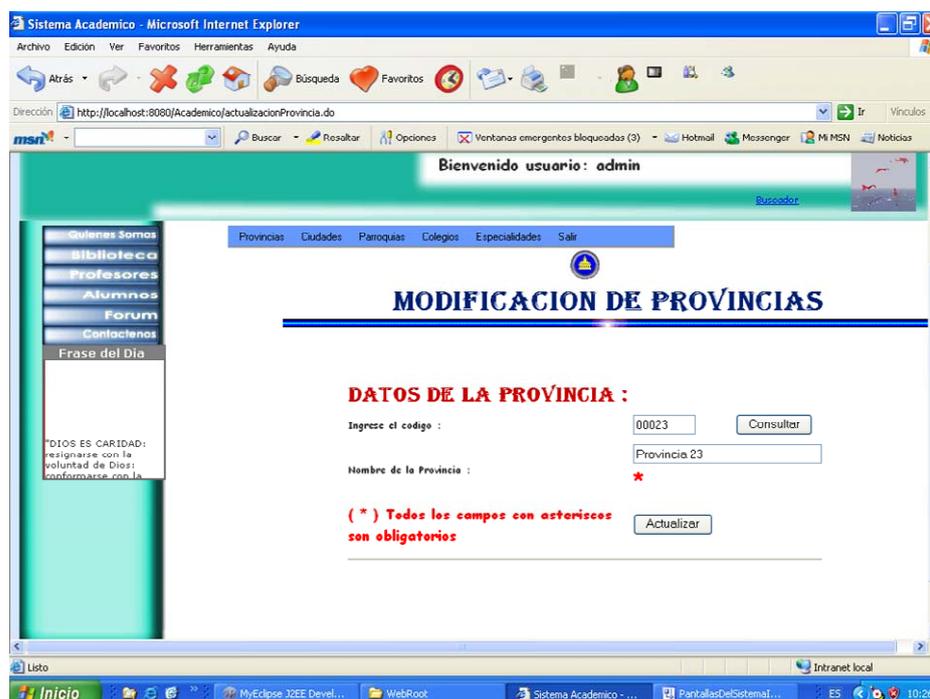


Fig. 3-7. Interfaz de Modificación de los Datos de una Provincia.

Una vez en mostrado el nombre el usuario podrá modificar cualquier error que hubiese tenido al ingresar la información, luego podrá presionar el botón de Actualizar para que los cambios realizados sean almacenados en la Base de Datos.

3.3.5. Eliminación de una Provincia.- Al escoger del menú eliminación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de la provincia, al presionar el botón Consultar y si se encuentra en la Base de Datos se mostrará todos los datos de esa provincia, tal como se muestra en la (fig. 3-8).

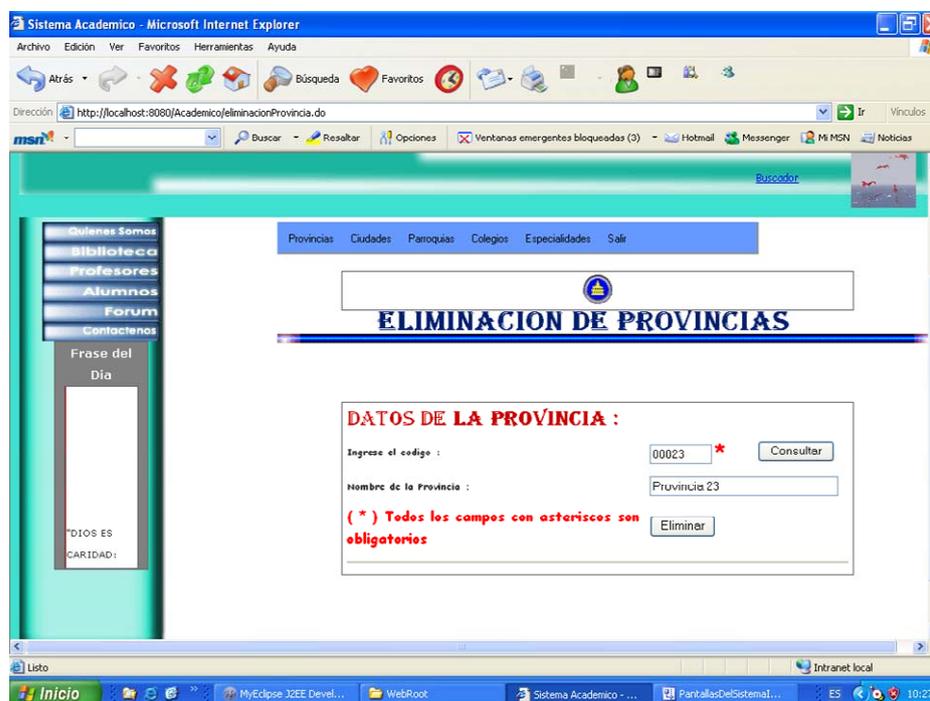


Fig. 3-8. Interfaz de Eliminación de una Provincia.

Cabe recalcar que la información solamente es eliminada de manera lógica de la Base de Datos.

3.3.6. Ingreso de una Ciudad.- Si el administrador desea adicionar una nueva ciudad debe de ejecutar el menú para ingreso de ciudades, así como se muestra en la (fig. 3-9).

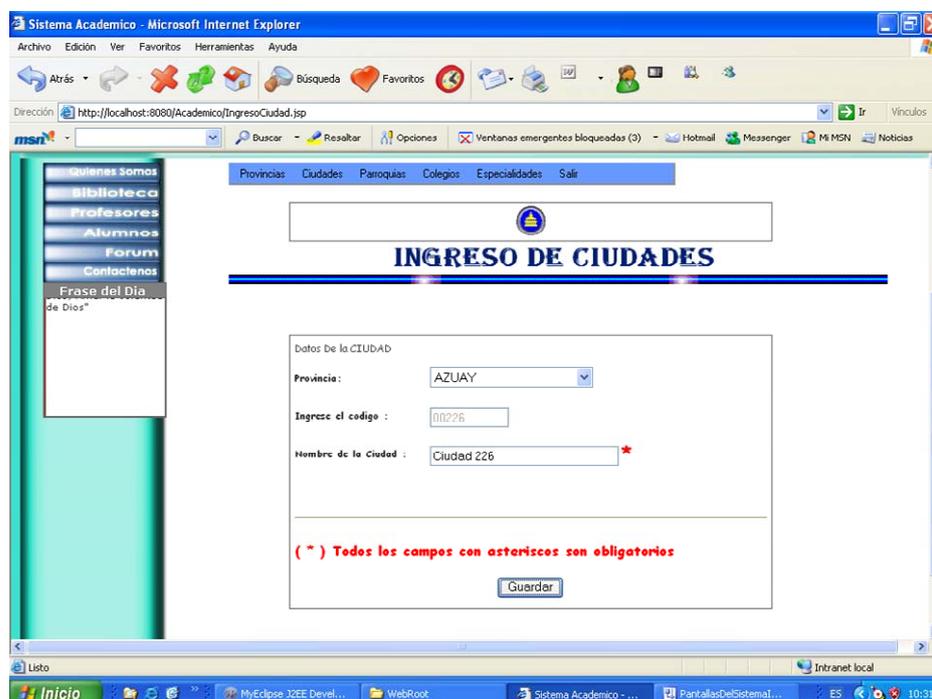


Fig. 3-9. Interfaz de Ingreso de las Ciudades.

En ésta pantalla se podrá ingresar el nombre de una nueva ciudad y su correspondiente provincia, si el administrador presiona el botón Guardar sin haber llenado el campo obligatorio aparecerá en la pantalla el correspondiente mensaje error de que debe de llenar ese campo, de lo contrario si cumple con lo solicitado se mostrará en la pantalla que los datos ingresados han sido almacenados con éxito en la Base de Datos.

3.3.7. Consulta de una Ciudad.- Si seleccionamos del menú Consulta aparecerá una pantalla en donde se podrá ingresar el código de la ciudad, al presionar el botón Buscar aparecerá el nombre que le corresponde a esa ciudad, véase en la (fig. 3-10).

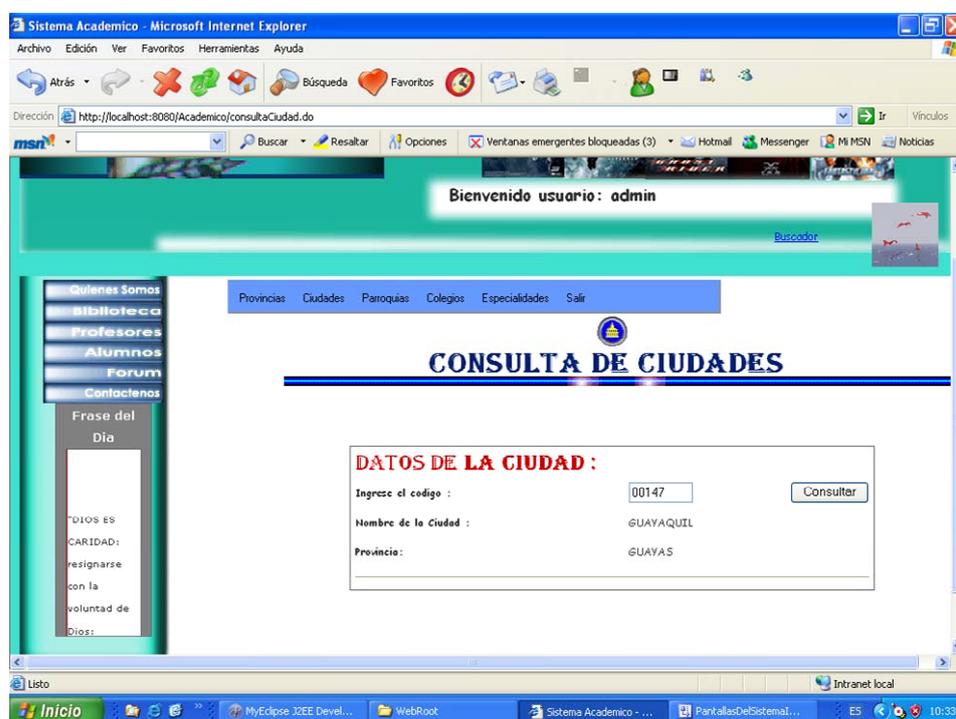


Fig. 3-10. Interfaz de Consulta por código de una Ciudad.

3.3.8. Modificación de una Ciudad.- Si se escoge del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de ciudad, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará el nombre correspondiente y la provincia a la que pertenece, véase en la (fig. 3-11).

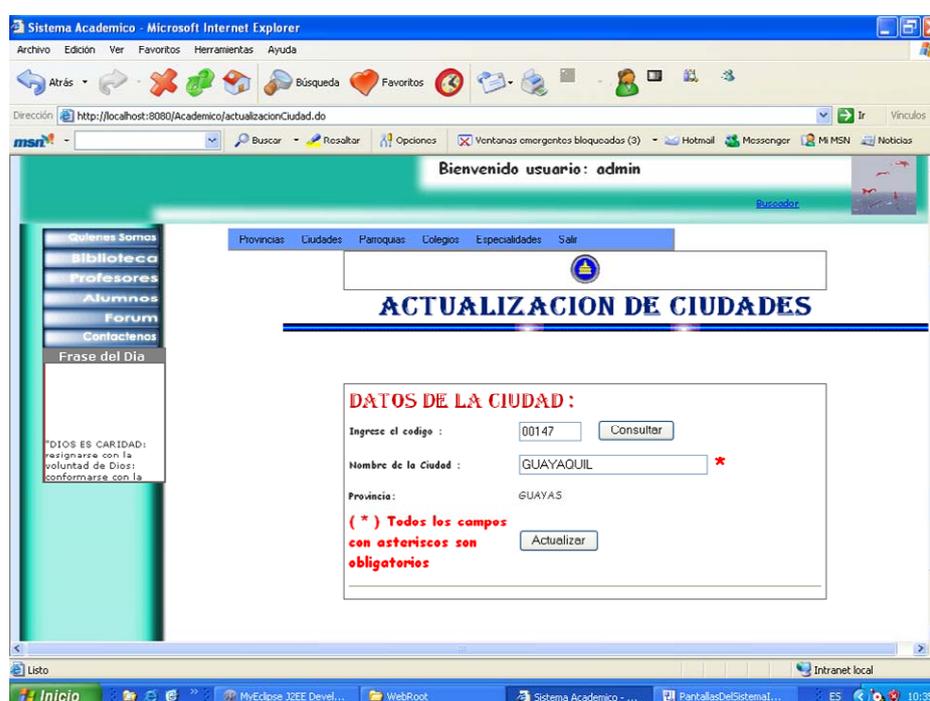


Fig. 3-11. Interfaz de Modificación de una Ciudad.

Una vez en mostrado el nombre de la ciudad el usuario podrá modificar cualquier error que hubiese cometido al ingresar la información, luego de esto podrá presionar el botón de Actualizar para que los cambios realizados sean almacenados en la Base de Datos.

3.3.9. Eliminación de una Ciudad.- Al escoger del menú eliminación nos presentará una pantalla en donde nos permitirá ingresar un código de ciudad, al presionar el botón Consultar y si se encuentra en la Base de Datos se mostrará todos los datos de esa ciudad, tal como se muestra en la (fig. 3-12).

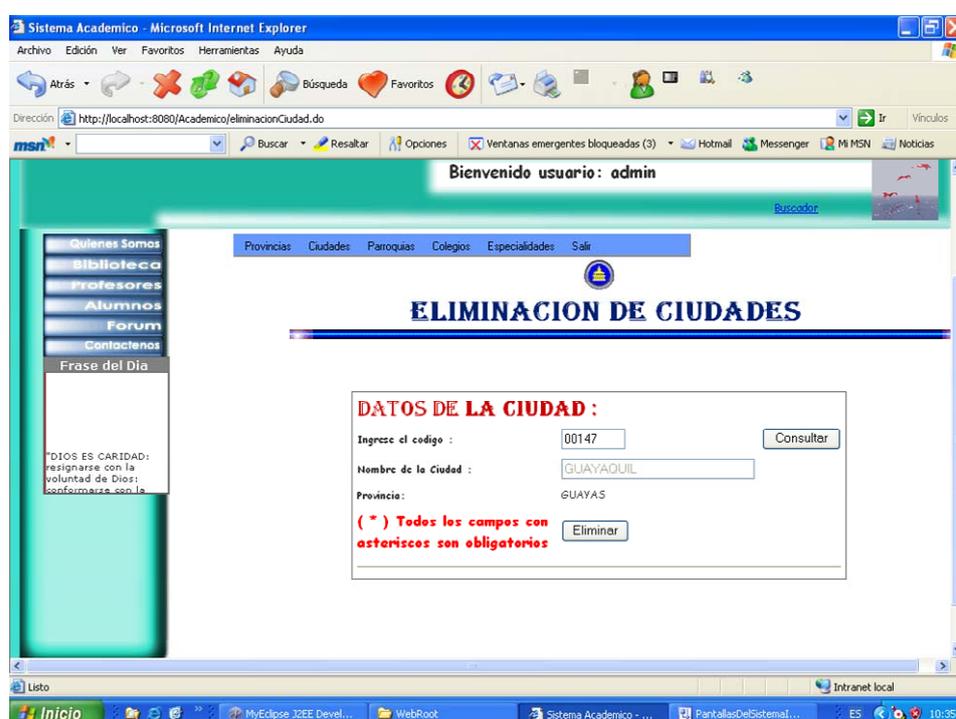


Fig. 3-12. Interfaz de Eliminación de una Ciudad.

Esta información que se elimina de la Base de Datos es solamente de manera lógica.

3.3.10. Ingreso de una Parroquia.- Cuando el administrador desea adicionar una nueva Parroquia debe de ejecutar el menú para ingreso de Parroquias, tal como se muestra en la (fig. 3-13).

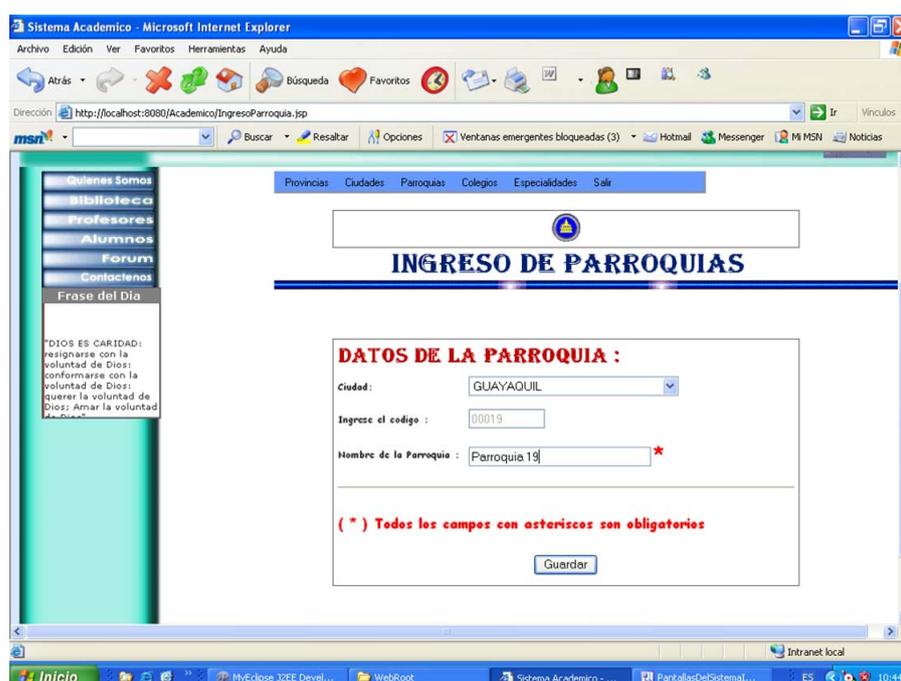


Fig. 3-13. Interfaz de Ingreso de Parroquias.

En ésta pantalla se podrá ingresar el nombre de una nueva parroquia seleccionando su correspondiente ciudad, si el administrador presiona el botón Guardar sin haber llenado el campo obligatorio aparecerá en la pantalla el correspondiente mensaje error de que debe de llenar ese campo, de lo contrario si cumple con lo solicitado se mostrará en la pantalla que los datos ingresados han sido almacenados con éxito en la Base de Datos.

3.3.11. Consulta de una Parroquia.- Si selecciona del menú Consulta aparecerá una pantalla en donde se podrá ingresar el código de la parroquia, al presionar el botón Buscar aparecerá el nombre de la parroquia y su correspondiente ciudad, véase en la (fig. 3-14).

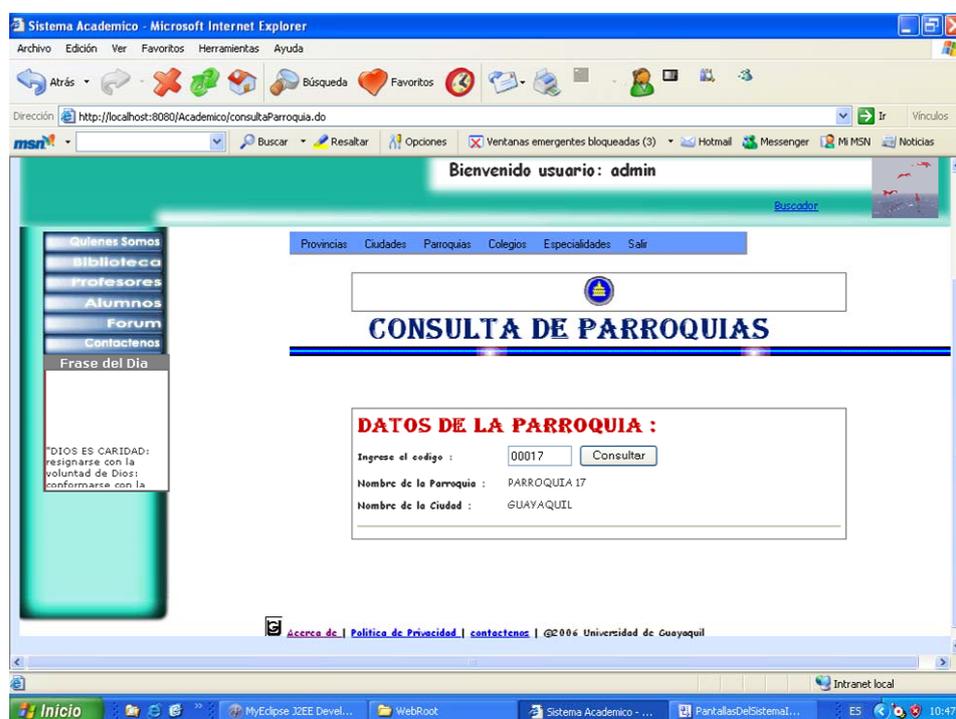


Fig. 3-14. Interfaz de Consulta por código de una Parroquia.

3.3.12. Modificación de una Parroquia.- Si se escoge del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código de parroquia, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará el nombre correspondiente, véase en la (fig. 3-15).

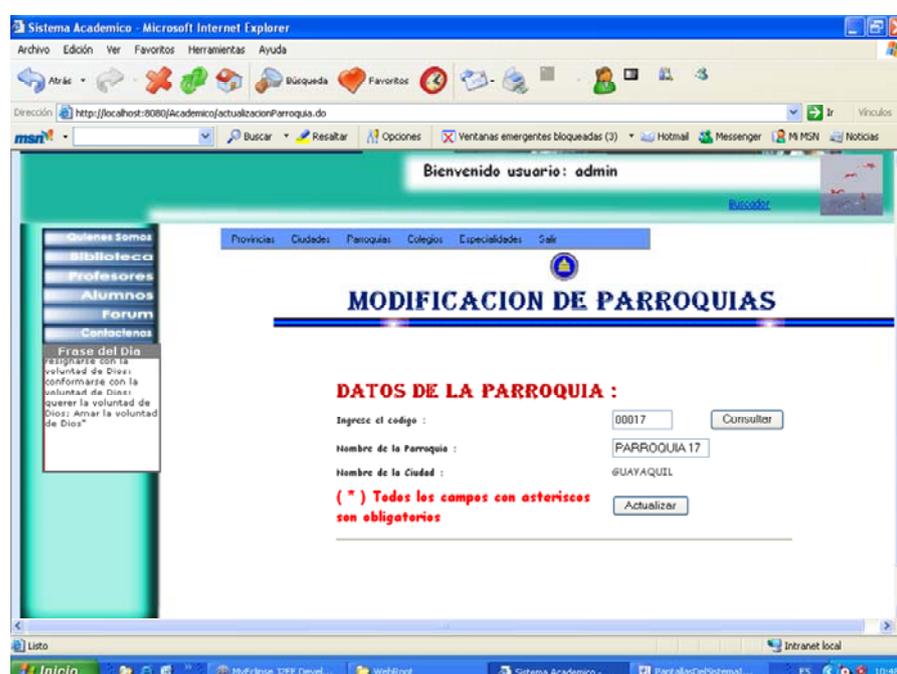


Fig. 3-15. Interfaz de Modificación de una Parroquia.

Una vez en mostrado el nombre de la ciudad el usuario podrá modificar cualquier error que hubiese cometido al ingresar la información, luego de esto podrá presionar el botón de Actualizar para que los cambios realizados sean almacenados en la Base de Datos.

3.3.13. Eliminación de una Parroquia.- Al escoger del menú eliminación nos presentará una pantalla en donde nos permitirá ingresar un código de una parroquia específica, al presionar el botón Consultar y si se encuentra en la Base de Datos se mostrará todos los datos de esa parroquia, tal como se muestra en la (fig. 3-16).

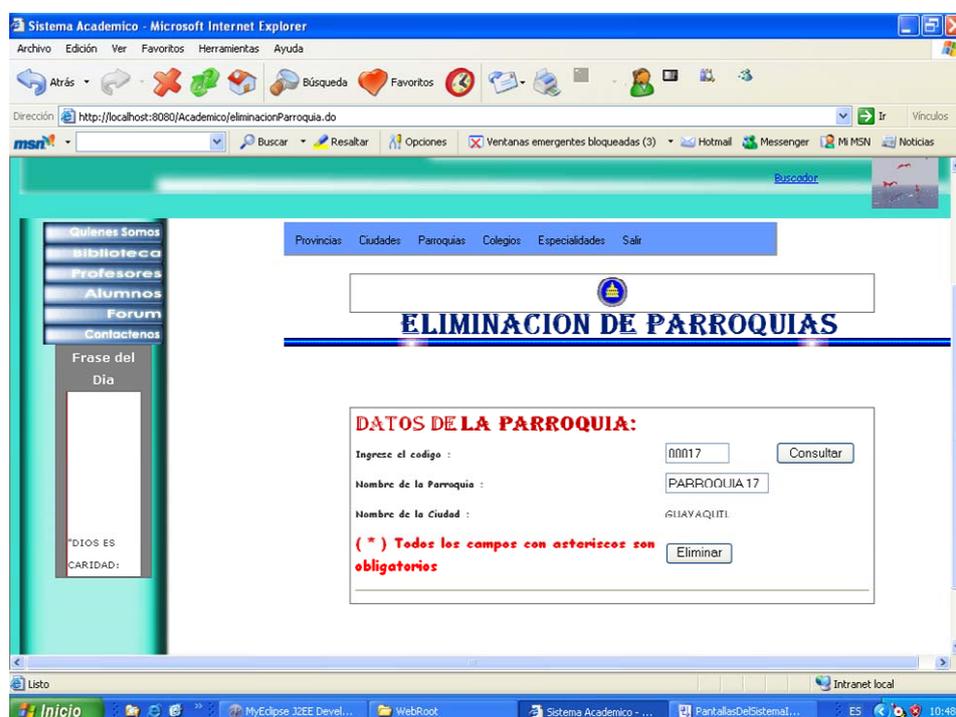


Fig. 3-16. Interfaz de Eliminación de una Parroquia.

Esta información que se elimina de la Base de Datos es solamente de manera lógica.

3.3.14. Ingreso de un Colegio.- Cuando el administrador desea adicionar un nuevo Colegio debe de ejecutar el menú para ingreso de Colegios, tal como se muestra en la (fig. 3-17).

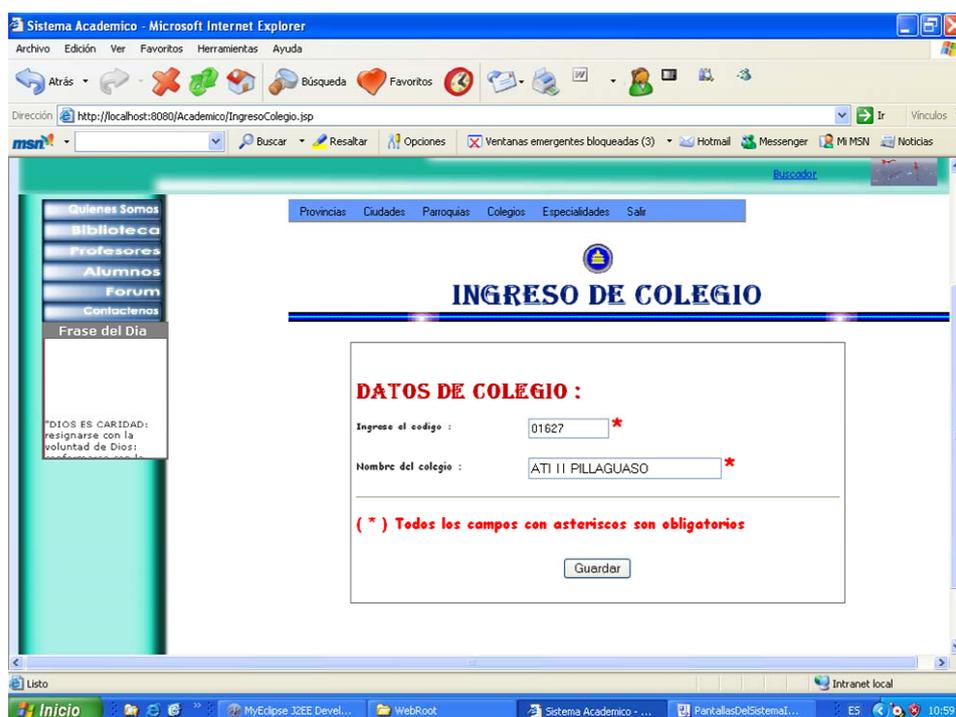


Fig. 3-17. Interfaz de Ingreso de Colegios.

En ésta pantalla se podrá ingresar el nombre del colegio, si el administrador presiona el botón Guardar sin haber llenado el campo obligatorio aparecerá en la pantalla el correspondiente mensaje error de que debe de llenar ese campo, de lo contrario si cumple con lo solicitado se mostrará en la pantalla que los datos ingresados han sido almacenados con éxito en la Base de Datos.

3.3.15. Consulta de un Colegio.- Si seleccionamos del menú Consulta aparecerá una pantalla en donde se podrá ingresar el código del Colegio, al presionar el botón Buscar aparecerá el nombre que le corresponde a ese Colegio, véase en la (fig. 3-18).

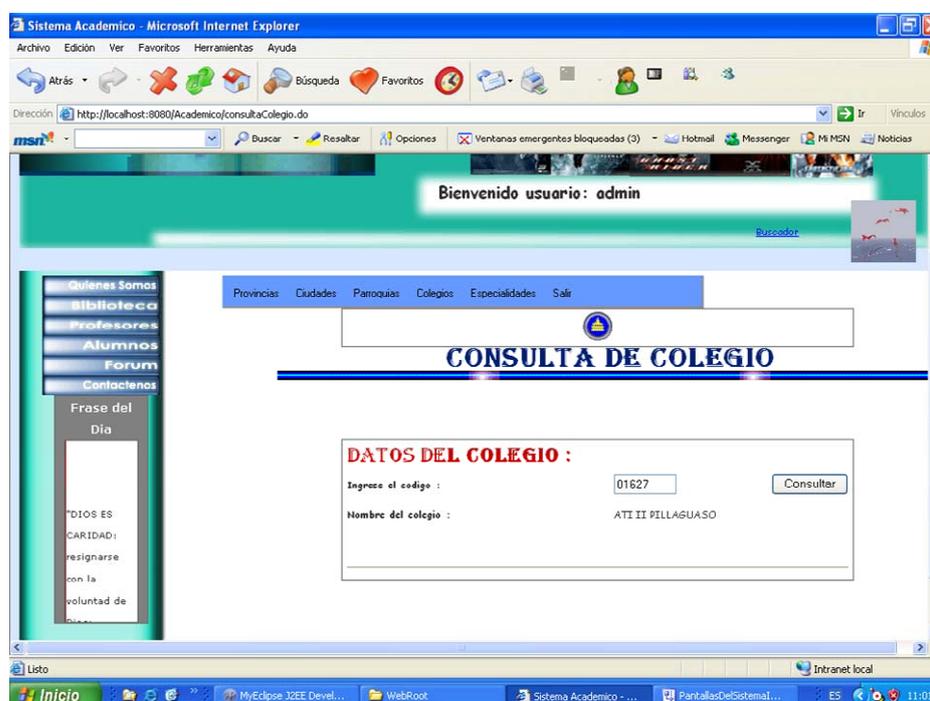


Fig. 3-18. Interfaz de Consulta por código de un Colegio.

3.3.16. Modificación de un Colegio.- Si se escoge del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código del Colegio, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará el nombre correspondiente, véase en la (fig. 3-19).

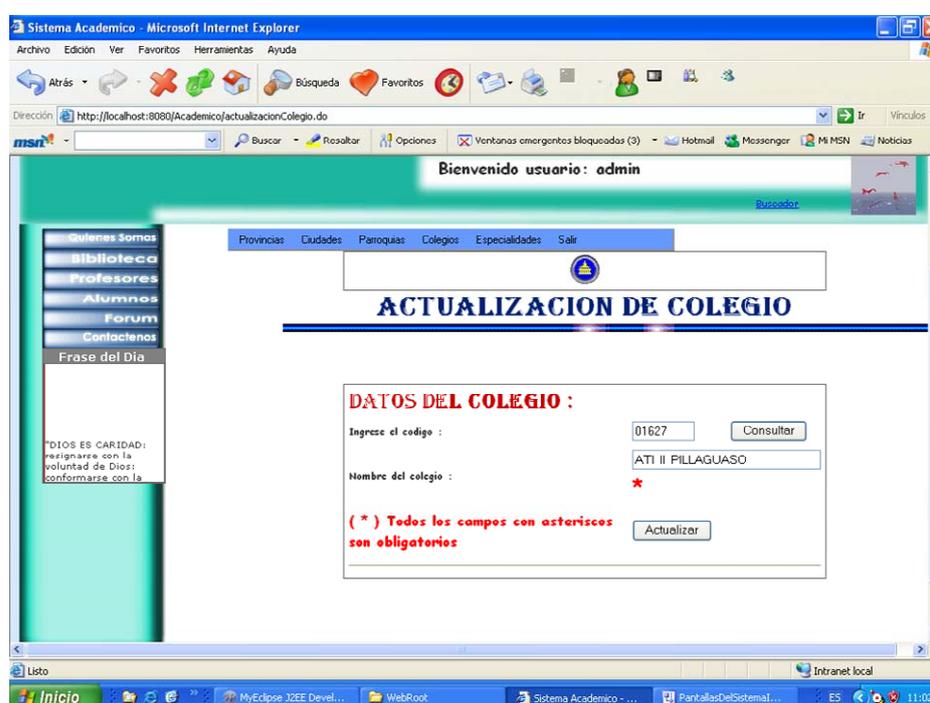


Fig. 3-19. Interfaz de Modificación de un Colegio.

Una vez en mostrado el nombre del colegio el administrador podrá modificar cualquier error que hubiese cometido al ingresar la información, luego de esto podrá presionar el botón de Actualizar para que los cambios realizados sean almacenados en la Base de Datos.

3.3.17. Eliminación de un Colegio.- Al escoger del menú eliminación nos presentará una pantalla en donde nos permitirá ingresar un código de un colegio específico, al presionar el botón Consultar y si se encuentra en la Base de Datos se mostrará todos los datos de ese colegio, tal como se muestra en la (fig. 3-20).

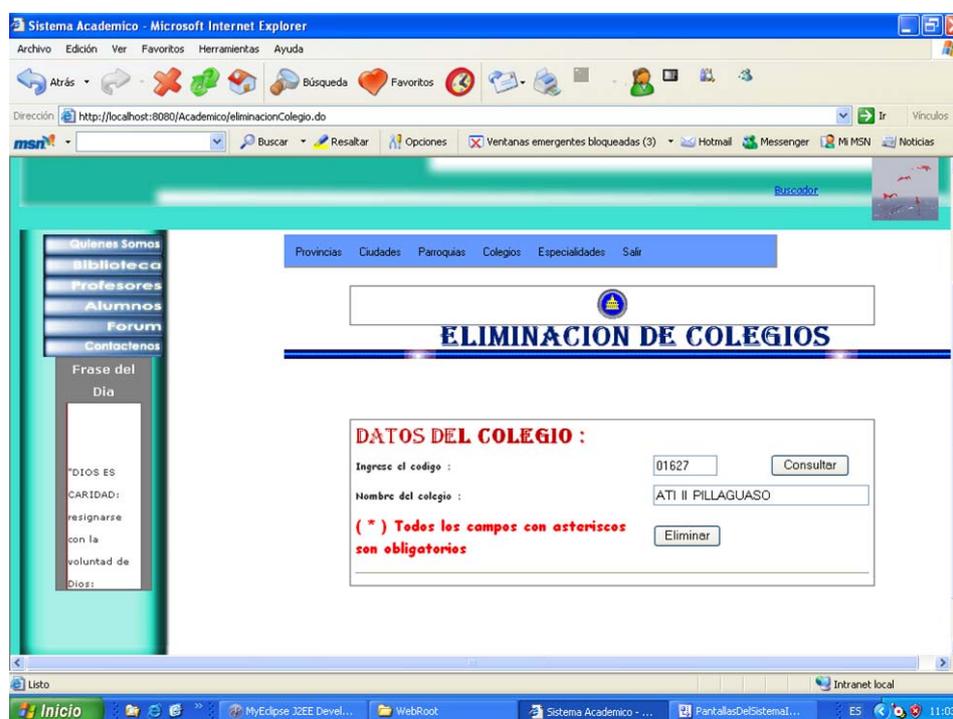


Fig. 3-20. Pantalla de Eliminación de Colegios.

Esta información que se elimina de la Base de Datos es solamente de manera lógica.

3.3.18. Ingreso de una Especialización.- Cuando el administrador desea adicionar una nueva especialización debe de ejecutar el menú para ingreso de especializaciones, tal como se muestra en la (fig. 3-21).

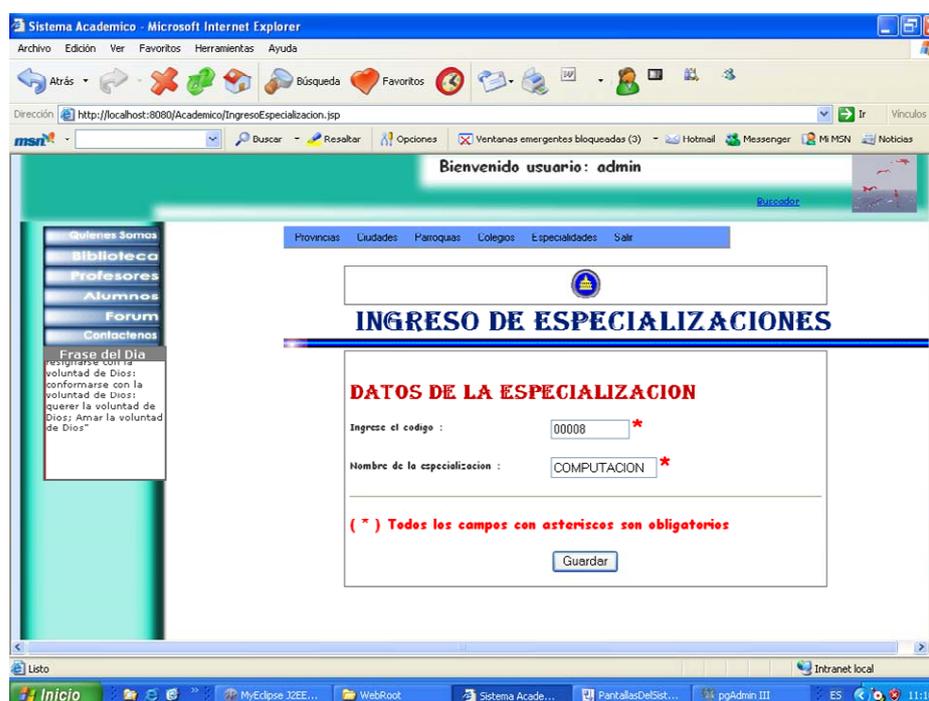


Fig. 3-21. Interfaz de Ingreso de Especializaciones.

En ésta pantalla se podrá ingresar el nombre de la especialización, si el administrador presiona el botón Guardar sin haber llenado el campo obligatorio aparecerá en la pantalla el correspondiente mensaje error de que debe de llenar ese campo, de lo contrario si cumple con lo solicitado se mostrará en la pantalla que los datos ingresados han sido almacenados con éxito en la Base de Datos.

3.3.19. Consulta de una Especialización.- Si seleccionamos del menú Consulta aparecerá una pantalla en donde se podrá ingresar el código del Colegio, al presionar el botón Buscar aparecerá el nombre que le corresponde a ese Colegio, véase en la (fig. 3-22).

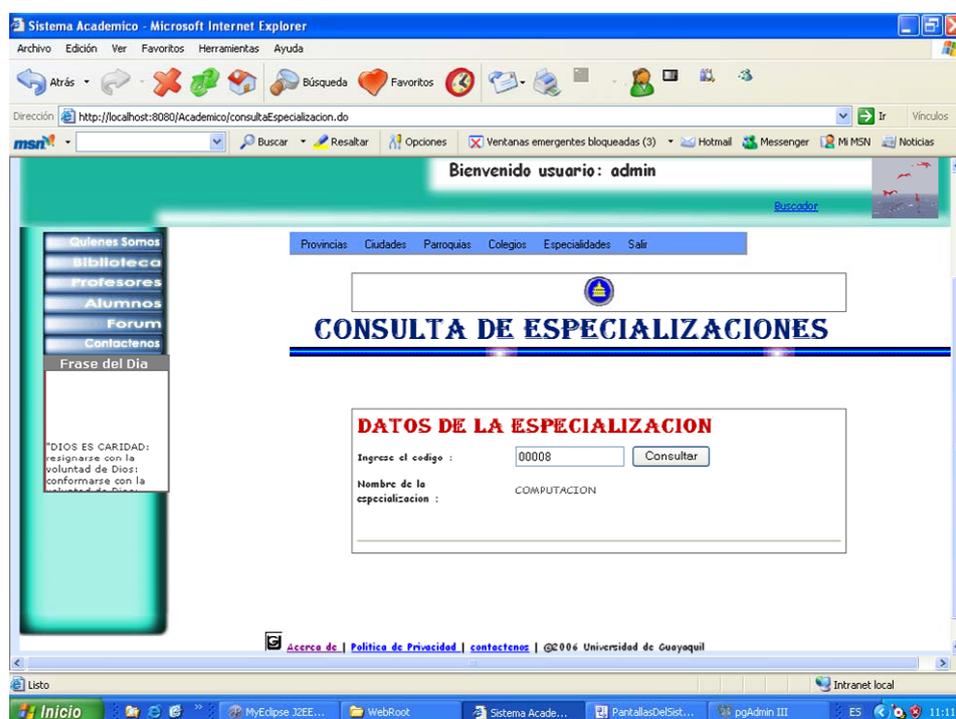


Fig. 3-22. Interfaz de Consulta por código de una Especialización.

3.3.20. Modificación de una Especialización.- Si se escoge del menú principal modificación nos mostrará una pantalla en donde nos dará la opción de ingresar un código del Colegio, al presionar el botón consultar y si se encuentra en la Base de Datos se mostrará el nombre correspondiente, véase en la (fig. 3-23).

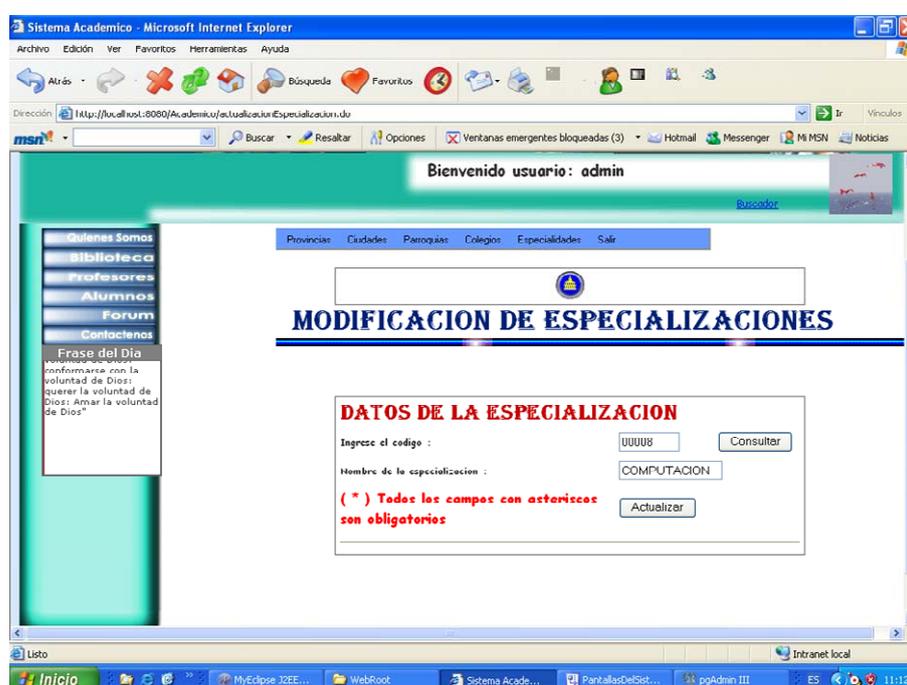


Fig. 3-23. Interfaz de Modificación de Especializaciones.

Una vez en mostrado el nombre del colegio el administrador podrá modificar cualquier error que hubiese cometido al ingresar la información, luego de esto podrá presionar el botón de Actualizar para que los cambios realizados sean almacenados en la Base de Datos.

3.3.21. Eliminación de una Especialización.- Al escoger del menú eliminación nos presentará una pantalla en donde nos permitirá ingresar un código de un colegio específico, al presionar el botón Consultar y si se encuentra en la Base de Datos se mostrará todos los datos de ese colegio, tal como se muestra en la (fig. 3-24).

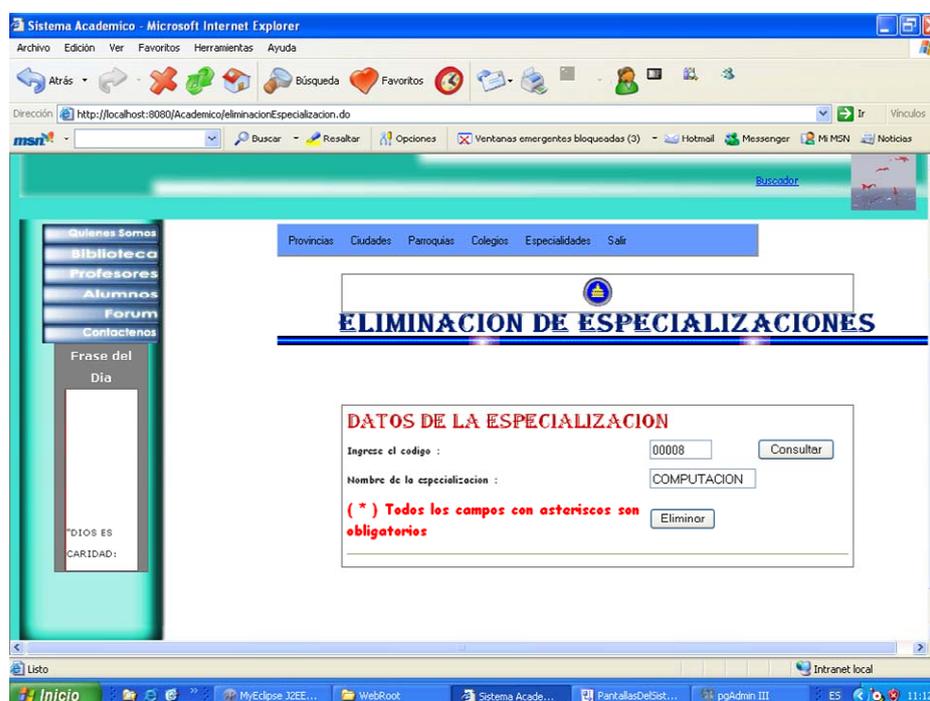


Fig. 3-24. Interfaz de Eliminación de Especializaciones.

Esta información que se elimina de la Base de Datos es solamente de manera lógica.