



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE INSCRIPCIÓN AUTOMATIZADO PARA
LAS SALAS DE LA UNIDAD DE SERVICIOS DE
CÓMPUTO ACADÉMICO DE LA FACULTAD DE
INGENIERÍA
(S I A S U)**

TESIS
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
PRESENTA:

Guillermo Rosales García

DIRECTORA DE TESIS: Ing. Ma. del Rosario Barragán Paz





**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA**



***SISTEMA DE INSCRIPCIÓN AUTOMATIZADO PARA
LAS SALAS DE LA UNIDAD DE SERVICIOS DE
CÓMPUTO ACADÉMICO DE
LA FACULTAD DE INGENIERÍA
(S I A S U)***

Directora de Tesis

Ing. Ma. del Rosario Barragán Paz

Presentada por

Guillermo Rosales García



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE INSCRIPCIÓN AUTOMATIZADO PARA
LAS SALAS DE LA UNIDAD DE SERVICIOS DE
CÓMPUTO ACADÉMICO DE LA FACULTAD DE
INGENIERÍA
(S I A S U)**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
GUILLERMO ROSALES GARCÍA



DIRECTORA DE TESIS: ING. Ma. DEL ROSARIO BARRAGÁN PAZ

CD. UNIVERSITARIA MÉXICO D.F.

JUNIO 2004

AGRADECIMIENTOS

A mi Alma Mater, la Universidad Nacional Autónoma de México.

A la Facultad de Ingeniería y a la Unidad de Servicios de Cómputo Académico (UNICA), por brindarme la oportunidad de ser parte de su gente y por prepararme profesionalmente durante toda mi estancia.

A la Ing. María del Rosario Barragán Paz por haberme brindado su amistad, su apoyo y su consejos para la realización de este trabajo.

A la Ing. Beatriz Barrera Hernández, por sus consejos y observaciones en todo momento.

A todos mis amigos que me ayudaron en la realización de este trabajo, el cuál comenzó con mi llegada a la facultad y que hoy concluye en estas hojas. Gracias a todos.

Dedicatoria

Este trabajo se lo dedico a todos mis **amigos** que me han motivado y apoyado en todo momento, y que han estado conmigo. Muchas gracias por sus palabras de aliento, por sus consejos y por su cariño, pero sobre todo **gracias por su amistad**.

A mis hermanos, Mónica, Jonatan y Edgar, por su apoyo, su cariño y su fuerza. Los quiero mucho a los tres y les deseo lo mejor de la vida. Disfrútenla y den lo mejor de ustedes.

A mi papá, Guillermo Rosales Islas, que me enseñó a ser fuerte y pelear por lo que quiero. Que me enseñó que cualquier trabajo se puede realizar si uno lo desea, y que no existe oficio que uno no pueda aprender. Gracias por enseñarme a depender por mí mismo.

Este trabajo te lo dedico principalmente a ti **mamá**, por que me apoyaste en todo momento y por que siempre creíste en mí. Este trabajo no solo refleja lo que aprendí en la que veo como mi segunda casa, la UNAM, sino que es el reflejo de todo tu esfuerzo, de tu paciencia y entrega por formar a un hombre responsable y respetable que sepa valorar la vida y luchar por aquello que quiere; es el reflejo de ti misma. Este trabajo esta dedicado para la mujer que más quiero y que admiro, su nombre es **María del Carmen García Olivares**, mi MAMÁ. No solo te debo la gracia de la vida; te debo todo, y este trabajo es una muestra de lo agradecido que estoy contigo. Te quiero mucho mamá y espero que Dios me permita disfrutar de tu compañía muchos años más.

ÍNDICE

Introducción	1
Objetivos	3
Capítulo 1. Marco General	5
1.1 Antecedentes de la Unidad de Servicios de Cómputo Académico.....	7
1.2 Objetivos y metas de UNICA.....	9
1.3 Servicios que ofrece.....	9
1.4 Organización y funciones por Departamentos.....	12
1.5 Necesidades de un sistema de inscripción automatizado.....	14
1.6 Alcance y justificación del sistema.....	15
Capítulo 2. Análisis del Sistema	17
2.1 Especificaciones del sistema.....	19
2.2 Requerimientos.....	24
2.3 Metodología empleada.....	27
2.3.1 Ciclo de vida clásico	27
2.3.1.1 Planeación del Sistema.....	27
2.3.1.2 Análisis.....	28
2.3.1.3 Diseño.....	28
2.3.1.4 Codificación	29
2.3.1.5 Implantación	30
2.3.1.6 Pruebas.....	30
2.3.1.7 Mantenimiento.....	30
2.3.2 Ciclo de vida clásico modificado.....	30
2.5 Selección del software	31
2.5.1 Análisis de requerimientos del software.....	31
2.5.2 Sistema operativo.....	31
2.5.3 Sistema manejador de base de datos	35
2.5.4 Manejador de la interfaz gráfica.....	40
2.6 Elección del Software.....	43
Capítulo 3. Diseño del Sistema	45
3.1 Formulación del modelo.....	48
3.2 Diseño de datos.....	54
3.3 Diccionario de datos	56
3.4 Creación de la base de datos de SIASU.....	59

Capítulo 4. Desarrollo del Sistema	61
4.1 Manejo de código.....	63
4.2 Implementación.....	66
4.3 Pruebas.....	75
4.3.1 Pruebas de caja blanca	76
4.3.1.1 Pruebas de unidad y de integración.....	76
4.3.2 Pruebas de caja negra	77
4.3.2.1 Pruebas de volumen	77
4.3.2.2 Pruebas funcionales o de validación	78
Capítulo 5. Administración y Mantenimiento	79
5.1 Mantenimiento del Sistema	81
5.2 Administración de la información.....	83
5.3 Perfiles de protección	83
Capítulo 6. Resultados y Costos	87
6.1 Resultados	89
6.1.1 Resultados de las pruebas de caja blanca.....	89
6.1.1.1 Pruebas de unidad y de integración.....	89
6.1.2 Resultados de las pruebas de caja negra.....	89
6.1.2.1 Pruebas de volumen	89
6.1.2.2 Pruebas funcionales	90
6.1.2.3 Pruebas estructurales	90
6.1.2.4 Pruebas de aceptación del sistema.....	90
6.2 Costos	93
CONCLUSIONES	95
Apéndice 1 (MANUAL DE USUARIO)	97
Apéndice 2 (GLOSARIO)	119
Apéndice 3 (CREACIÓN DE UNA BASE DE DATOS)	121
BIBLIOGRAFÍA	137

INTRODUCCIÓN

La Unidad de Servicios de Cómputo Académico (UNICA) ofrece, a los alumnos y académicos de la Facultad de Ingeniería servicios institucionales como son: servicio Web, servicio de correo electrónico, manejador de bases de datos, publicación de páginas Web, y software para compilación; estos servicios pueden ser utilizados desde cualquier sitio donde haya una computadora y una conexión a Internet. Además de proporcionar a los alumnos el servicio de Salas de cómputo, en el cual pueden hacer uso de los equipos de cómputo y del software de tipo académico de mayor demanda y siempre actualizado.

El siguiente trabajo muestra cómo se construyó el Sistema de Inscripción Automatizado para las Salas de la Unidad de Servicios de Cómputo Académico (SIASU) de la Facultad de Ingeniería, así como las bases teóricas que determinaron el desarrollo y su puesta en marcha.

En el primer capítulo se da un panorama general de la historia y situación actual de UNICA y se explica la necesidad y alcances de SIASU.

Se lleva a cabo un análisis de acuerdo a las especificaciones dadas por el cliente (UNICA) y en las cuales se basa este trabajo. Se analizaron los requerimientos necesarios para la elaboración de este trabajo y conforme a este análisis se hizo la elección del software y de la metodología de acuerdo a la problemática. Estos puntos se tratan en el capítulo 2 (Análisis del sistema).

En el capítulo 3 se describe el diseño del sistema. Se formula el modelo con base en los Diagramas de Flujo de Datos, se expone un diseño de datos y se analiza un Modelo Relacional para darnos una perspectiva del flujo que tomarán los datos. Terminando con la creación de la Base de Datos.

En el cuarto capítulo se muestra el desarrollo del sistema y se explica cómo se llevó a cabo su implementación con una introducción al tipo de programación.

Las pruebas que se realizaron al producto terminado para poder ser “liberado” y poder trabajar con el mismo se comentan en el quinto capítulo de esta tesis.

En el sexto capítulo se indican los pasos a seguir para la instalación y puesta en marcha del Sistema así como las pruebas que se realizaron para verificar su funcionamiento y detectar fallas.

El proyecto además cuenta con la documentación correspondiente para dar inicio y mantenerlo en marcha de manera adecuada y eficiente expresado ampliamente en el anexo A y capítulo 6.

Objetivo General

Crear un sistema automatizado, para el alta de usuarios de la Facultad de Ingeniería de forma que puedan tener acceso a las Salas de Cómputo y a los servicios que presta la Unidad de Servicios de Cómputo Académico.

Objetivos Particulares

- Desarrollar un sistema para dar de alta a los alumnos desde las distintas instalaciones de UNICA ubicadas en la División de Ciencias Básicas y en el edificio principal de la Facultad de Ingeniería.
- Generar cuentas de usuarios para los alumnos de la Facultad que requieren hacer uso de los servicios que la Unidad de Servicios de Cómputo Académico ofrece.
- Monitorear la afluencia de usuarios que hacen uso de las instalaciones y obtener estadísticas de acuerdo al semestre que cursan y por la carrera a la cual están inscritos.
- Generara un módulo de consultas.
- Contar con un historial de las personas que hacen uso del módulo de consultas.
- Crear un documento en el que aparezca la información necesaria, para que el alumno pueda hacer uso de los servicios institucionales e información extra sobre los servicios y características propias de UNICA.
- Generar estadísticas de los alumnos.

- Apoyar a otro sistema con la información que se generan en SIASU para que de manera conjunta agilicen el acceso a las distintas Salas de cómputo de UNICA. Esto se pretende llevar a cabo tomando como punto de partida una de las tablas de la base de datos de SIASU que estará creando con los alumnos que se den de alta en UNICA. El sistema es llamado Sistema de Control de Salas de UNICA (SCOSU).
- Generar la documentación correspondiente, tal como manual de usuario.

Marco General

Capítulo

1

1. 1- Antecedentes de la Unidad de Servicios de Cómputo Académico

La Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería (UNICA) es resultado de la división del antiguo Centro de Cálculo (CECAFI), que se realizó en febrero de 1994.

El CECAFI se fundó en 1972 por acuerdo del entonces Director Juan Casillas García de León; entre los principales servicios que brindaba eran: la elaboración de sistemas, programas de apoyo académico-administrativo, asesorías, cursos para alumnos y personal académico, apoyo a las asignaturas ofreciendo servicios en temas de cómputo (préstamo de equipo y asesorías), elaboración de material didáctico de computación, realización de sistemas y bases de datos de los alumnos de la Facultad, así también se investigaba acerca de las nuevas aplicaciones de la computación en las distintas áreas de estudio de la Facultad de Ingeniería.

La estructura de CECAFI hasta 1992 se muestra en la Fig. 1.1

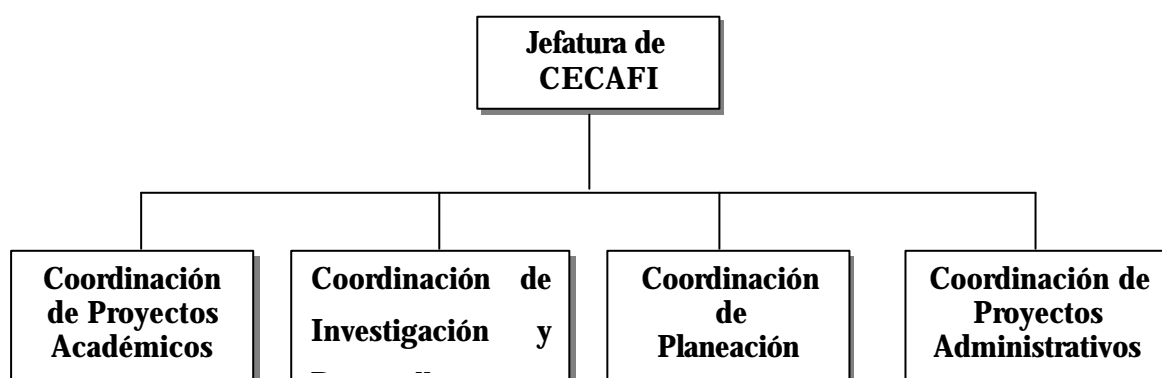


Fig. 1.1 Estructura de CECAFI hasta 1992

En 1993 el CECAFI se reestructuró dividiéndose en dos Centros de Cómputo (Fig. 1.2):

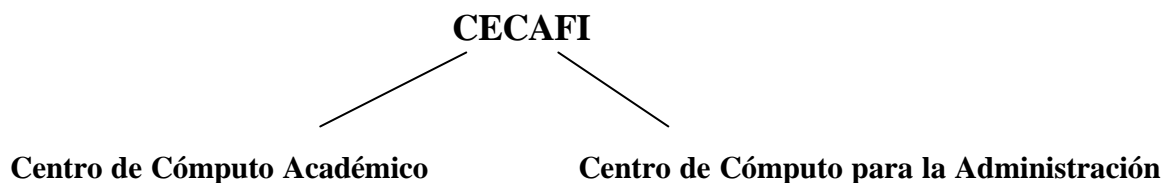


Fig. 1.2 Reestructuración de CECAFI 1993

Hasta entonces el Centro de Cómputo Académico tuvo las funciones de formar recursos humanos y proporcionar servicios en cómputo; mientras tanto, las funciones del Centro de Cómputo para la Administración fueron mantener la información académica, elaborando sistemas de información y administrando equipos de cómputo.

El personal que integra UNICA es: personal académico, administrativos de confianza, administrativos de base, becarios y servicio social. Cabe hacer notar que la mayor parte del personal la constituyen los becarios.

1. 2- Objetivos y Metas de UNICA

- Proporcionar recursos de cómputo de calidad a la comunidad de la Facultad de Ingeniería.
- Apoyar a Divisiones y Secretarías en cualquier actividad relativa a cómputo.
- Apoyar y colaborar con la Secretaría General en las actividades que involucran a toda la Facultad de Ingeniería.
- Formar recursos humanos de calidad no sólo en el área de cómputo sino también en su vida profesional.

Misión:

Ofrecer a la comunidad de la Facultad de Ingeniería un servicio de calidad suministrando recursos de cómputo, comerciales y especializados, que el avance de la educación demande.

Visión:

Mantener el liderazgo y estar a la vanguardia en cómputo tanto dentro de la Facultad de Ingeniería como en el entorno universitario.

1. 3- Servicios que ofrece

La Facultad de Ingeniería (FI) cuenta con ocho Divisiones que ofrecen recursos de cómputo en temas especializados. UNICA no particulariza en ninguna rama de la ingeniería, sino que ofrece recursos desde un punto de vista general a toda la comunidad de FI. Actualmente se ofrecen los siguientes servicios:

Servicios de cómputo. Para brindar este servicio a la comunidad de la FI, UNICA cuenta con tres salas de cómputo. La primera se encuentra en el edificio Principal y las siguientes dos en el anexo de la Facultad.

- Sala 1.- Ubicada en el edificio principal (edificio del CECAFI) de la FI. Cuenta con dos salas: La sala de estaciones de trabajo y la sala de computadoras personales.
- Sala 2.- Ubicada en el anexo de Ingeniería abajo del auditorio Sotero Prieto. Esta sala cuenta con 4 salas A, B, C y D.
- Sala 3.- Ubicada en el anexo de Ingeniería (en la antigua biblioteca). Esta sala cuenta con cuatro salas E, F, G y H.

Cursos de computación.

- Intersemestrales.- Cada periodo intersemestral la Facultad de Ingeniería ofrece diversos cursos de computación, los cuales son impartidos por las divisiones de la Facultad, éstas envían sus propuestas de cursos a UNICA donde se realiza la planeación, inscripción, propaganda, difusión y se generan las constancias de aprovechamiento.
- Semestrales.- Funciona igual que los intersemestrales, solo que son dirigidos principalmente a los alumnos de la facultad
- Sábados y domingos.- UNICA también coordina los cursos que se ofrecen en sábados y domingos durante el periodo semestral, es decir, durante el periodo de clases. Estos cursos están abiertos para el público en general.
- Especiales.- Así también se preparan cursos especiales para las Divisiones y Secretarías de la Facultad cuando éstas así lo soliciten.
- Externos.- Se preparan cursos especiales para empresas o personas externas a la UNAM.

Formación de recursos humanos. Al final de semestre, UNICA convoca a los alumnos de la Facultad de Ingeniería al concurso de ingreso al plan de formación de becarios, en esta etapa se selecciona y prepara a los candidatos que cumplan con el perfil requerido y se les da capacitación en cómputo a fin de contar con los mejores candidatos para formar parte de su personal. Al final de su preparación, los aspirantes a becarios tienen que

presentar un examen final y de acuerdo a su desempeño son seleccionados por el comité del programa “Plan de Formación de Becarios”. Siendo ya becarios, UNICA los asigna de acuerdo con sus intereses y los de la Unidad a un Departamento, donde posteriormente desarrollarán proyectos especiales, sistemas, y docencia entre otros.

Control del equipo de cómputo. UNICA tiene la responsabilidad de concentrar los requerimientos para la compra de equipo de cómputo de las Divisiones y Secretarías de la FI; así mismo realiza investigación de características técnicas y contacta con proveedores.

Coordinación del mantenimiento preventivo del equipo de cómputo. La Facultad de Ingeniería tiene contrato con una empresa de mantenimiento preventivo de equipo de cómputo, UNICA en conjunto con Secretaría Administrativa realiza un plan de mantenimiento para el equipo de la Facultad y verifica que se cumpla.

Censo de equipo de cómputo. Cada año, el Consejo Asesor de Cómputo de la UNAM solicita una relación de equipo de cómputo a cada Facultad, UNICA es responsable de hacer el concentrado del inventario de las Divisiones y Secretarías y entregar el censo correspondiente a la Facultad de Ingeniería.

Coordinación del inventario de la Facultad de Ingeniería. UNICA realiza un plan de acción para concentrar la información del inventario de equipo de cómputo de la Facultad de Ingeniería, coordina a todas las Divisiones y Secretarías de la Facultad para realizar un inventario de sus áreas.

Elaboración de material didáctico. En la Unidad se elaboran notas y material didáctico sobre temas de computación de acuerdo a los cursos impartidos.

Estudio y aplicación de software. Investigación y desarrollo del software comercial y especializado que la actividad de la Facultad requiere y prestación de servicios de asesoría sobre estos tópicos; desarrollo de sistemas necesarios en UNICA y que la comunidad de la Facultad demanda.

1. 4 Organización y Funciones por departamento

Para el cumplimiento de sus funciones UNICA, está integrada por su Jefatura, por los Departamentos de Servicios Académicos, de Investigación y Desarrollo, de Redes y Operación de Servidores, así como por la Coordinación de Salas de Atención a Alumnos.

Esto se puede apreciar en la figura 1.3.

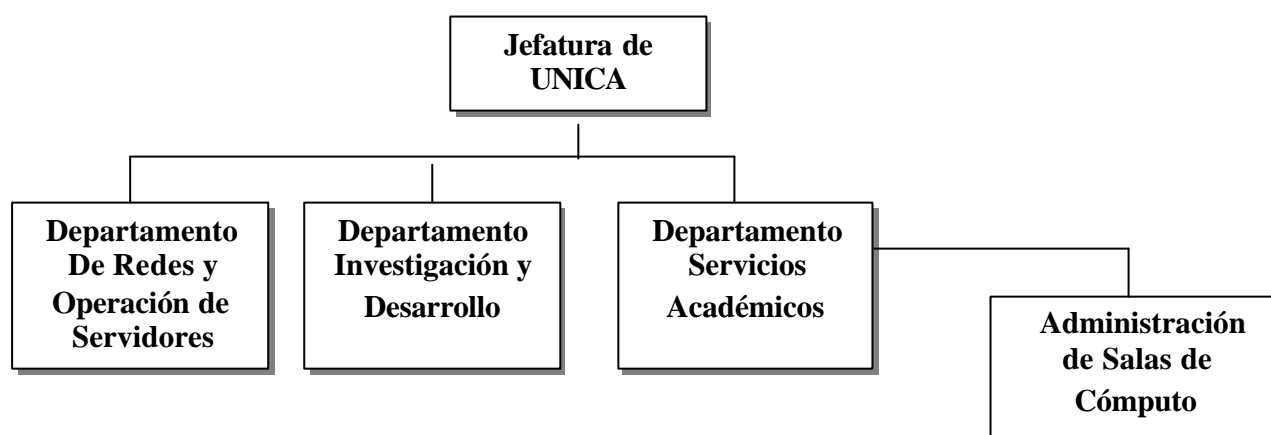


Fig. 1.3 Departamentos UNICA

Departamento de Servicios Académicos

Este departamento se encarga de planear, organizar e impartir los cursos de cómputo que la Unidad ofrece sistemáticamente, semestre a semestre en las modalidades de: semestrales, intersemestrales, de fin de semana, especiales, internos, así como los cursos requeridos para la formación de los alumnos becarios de UNICA.

Se encarga así mismo de la formación de los recursos humanos que la Unidad necesita para el cumplimiento de sus funciones; esto es mediante la coordinación del Programa de Formación de Becarios de UNICA; en el que alumnos de la Facultad, con alto rendimiento académico, que así lo deseen y manifiesten, previas selección, capacitación, evaluación y

aceptación, se integran como personal de apoyo a la Unidad; siendo de hacerse destacar la importante participación de este recurso humano de alta especialización, para el cumplimiento de las responsabilidades de UNICA. Se imparten unos 100 cursos al año en los que se atienden del orden de 7000 alumnos en promedio y se preparan 15 becarios en promedio al semestre.

Departamento de Investigación y Desarrollo

Se encarga de estar al día en la tecnología y avances en materia de cómputo, a efecto de estar en la posibilidad de incorporarle a la brevedad, a los servicios que la Unidad ofrece a los alumnos, manteniéndoles oportunamente actualizados. Se encarga así mismo del desarrollo de sistemas de información, administración y control, apoyo a las actividades académico-administrativas internas de la Unidad, de la Facultad y para la solución de problemas específicos de clientes externos.

Es responsable del área de Multimedia de la Unidad. Administra y da mantenimiento al servicio de manejo de bases de datos, que se ofrece a los alumnos. Es el responsable de dar mantenimiento y actualizar la página Web de la Unidad.

Departamento de Redes y Operación de Servidores

Responsable de la administración, operación, mantenimiento y seguridad de la red de comunicación de la Facultad y de la intercomunicación con la red central de la UNAM; así como del desarrollo e implantación de proyectos para la expansión del servicio.

Es también responsable técnico en la Web de la Facultad de Ingeniería, actualizar y mantener el área de la Web correspondiente a UNICA. Coordinar el subcomité de administradores y fungir como el área que administra la red de cómputo general ante DGSCA.

Proporcionar apoyo para la automatización de las actividades de la Secretaría General que así lo requieran. Administra y da mantenimiento a las cuentas del servicio de correo electrónico y de Internet de la Facultad. Unas 7,000 cuentas individuales, en promedio semestral.

Coordinación de Salas de Atención a Alumnos

Su función principal es la de proporcionar el servicio de cómputo y de impresión a los alumnos de la Facultad, para que estos puedan realizar sus tareas y sus trabajos de investigación, para lo cual además les proporciona servicio de correo electrónico, acceso a Internet y servicios varios de apoyo en materia de cómputo.

Al efecto se cuenta con tres salas de atención a usuarios, una en el Edificio Principal y dos en el Área Sur; en las que se cuenta con un total de 240 computadoras personales, 12 impresoras y 10 estaciones de trabajo, como equipo al servicio exclusivo de los alumnos, el cual se complementa con 9 servidores y las computadoras necesarias para la administración y control del servicio. Las salas funcionan de 9 de la mañana a 9 de la noche y se cuenta además con el servicio de asesoría especializada, brindada por becarios instructores, prácticamente durante todo el día.

Los servicios se ofrecen en plataforma Windows, plataforma Linux y plataforma HP-UX; y se cuenta con la paquetería de uso y aplicación mas frecuentes, la cual se actualiza y complementa constantemente. El año pasado se realizaron 50,560 sesiones-alumno, de 2 horas de duración en promedio.

1. 5 Necesidad de un Sistema de Inscripción Automatizado:

Anteriormente la Unidad de Servicios de Cómputo Académico realizaba el proceso de inscripción mediante el manejo de archivos lo que hacía lento el proceso mismo de inscripción, ya que era necesario hacer búsquedas en los distintos archivos para saber qué alumnos se encontraban registrados y qué alumnos no y una vez dado de alta, el usuario debía esperar al menos un día para poder hacer uso de los servicios.

En caso de que el alumno no estuviera registrado en la Facultad (regularmente sucede con alumnos que ya concluyeron sus créditos pero que aún no cumplen con el servicio social, o aquellos que se encuentran realizando su tesis pero que de igual forma ya concluyeron la carrera) y pretendiera utilizar los servicios institucionales de UNICA, se le solicitaba llenar un formato para uso de los servicios.

Debido al crecimiento tecnológico y a la demanda de usuarios que requieren de estos servicios, la Unidad de Servicios vio la necesidad de que el proceso descrito se realizara de una manera más eficiente y cómoda para el alumno.

Con el sistema propuesto se pretende que los usuarios puedan hacer uso de los servicios institucionales de UNICA y se puedan dar de alta en cualquiera de sus tres Salas. Esto es indispensable, ya que al hacerlo se les otorga un documento en el que aparecen: un identificador de usuario y una contraseña, y con los cuales es posible utilizar todos los servicios que ofrece la Unidad.

Optimizar el trámite de alta de alumnos. Con relación al alta de académicos y grupos de la Facultad a este sistema, se están analizando propuestas para una futura implementación.

Cabe resaltar que la elaboración de este trabajo nos brinda la oportunidad de utilizar las bases teóricas que se aprenden en diferentes asignaturas curriculares y aplicar metodologías para plantear y resolver un problema como lo es “la administración de los recursos de un sistema y un servidor” y explorar al máximo el software existente en la Facultad de Ingeniería.

1. 6 Alcance y justificación del Sistema

El servicio que ofrece UNICA es para toda la comunidad de la Facultad de Ingeniería. El sistema optimizará el servicio ofrecido únicamente al alumnado de esta.

La Unidad de Servicios de Cómputo Académico ofrece servicios institucionales como son: servicio Web, servicio de correo electrónico, manejador de bases de datos, publicación de

paginas Web, el servicio de préstamo de equipos donde los alumnos, pueden hacer uso del software (de tipo académico) de mayor demanda, así como de software para compilación; estos servicios pueden ser utilizados desde cualquier sitio donde haya una computadora y una conexión a Internet.

Actualmente se realiza este proceso mediante el manejo de archivos, lo que hace lento el proceso mismo, ya que no solamente se tiene que hacer búsquedas por medio de filtros, sino que es necesario que el alumno, una vez dado de alta, se tenga que esperar al menos un día para poder hacer uso de los servicios ofrecidos.

Para poder llevar a cabo el trámite de alta de alumnos de manera óptima y confiable, es necesario conformar un sistema que realice las mismas operaciones pero apoyada con software dedicado a dichas tareas, lo que no sólo dará mayor rapidez y seguridad sino que permitirá al usuario hacer uso de todos los servicios automáticamente. Además de realizar una adecuada administración de las visitas realizadas y de los movimientos llevados a cabo por las personas encargadas del alta de usuarios.

Análisis del Sistema

Capítulo

2

La Unidad de Servicios de Cómputo Académico, a través de sus Salas de Cómputo, ofrece servicios institucionales a los alumnos de la Facultad de Ingeniería. Para poder ofrecer estos servicios cada periodo semestral UNICA solicita a los alumnos que se den de alta en sus instalaciones. Para este efecto se solicita su credencial y su tira de materias, para comprobar que se trata de un alumno regular de la Facultad. A su vez UNICA le otorga un documento con datos propios del alumno así como un login y una contraseña. Esto se lleva a cabo cada semestre.

Para que el usuario pueda hacer uso de los servicios institucionales de UNICA, es indispensable que realice el trámite especificado anteriormente. Con el sistema propuesto lo podrá hacer en cualquiera de sus Salas.

2.1- Especificaciones del Sistema.

En esta sección se especifican los requisitos del “cliente”, el concepto general que el cliente tiene sobre el sistema, lo que desea que éste realice, los alcances que deberá tener y hacia quién esta enfocado; así como los datos que se desean obtener en cada una de las consultas.

Para obtener las especificaciones del sistema fue necesario platicar con las personas involucradas con los servicios de las Salas, esto es, jefes de Departamentos y personal que está directamente en contacto con el movimiento, manejo y expedición de los comprobantes de altas, ya que ellos serán los usuarios finales del Sistema.

Es importante determinar desde un principio los alcances y límites que debe cumplir este sistema en cuanto al manejo de información, los usuarios finales, y los recursos con los que cuenta la Institución.

Una vez que se tiene contemplado estos puntos se deben considerar los aspectos necesarios por parte de la Institución para la creación del sistema.

Requerimientos del cliente.

1. La Unidad de Servicios de Cómputo Académico, requiere un sistema que pueda ser operado desde cualquiera de sus Salas.
2. Desde cualquier equipo que ellos designen para este fin.
3. Que el trámite de Alta de usuarios sea más ágil para beneficio de los usuarios.
4. El sistema, automáticamente, debe generar un login de usuario.
5. Con base en este login, el Sistema Operativo se encargará de asignarle una “cuenta” con cierta “cuota”. Esta cuenta o espacio en el disco duro del servidor se genera para que el alumno pueda hacer uso de los servicios institucionales que presta esta Unidad.
6. Estadística de alumnos por semestre y de alumnos por carrera.

Es indispensable, por tanto, que el sistema dé de alta a los usuarios, y de esta forma, permitir su acceso a las Salas de cómputo de cualquiera de las instalaciones de UNICA. El comprobante, que valide que el usuario realizó el trámite, le será entregado.. Por tal motivo el sistema será capaz de imprimir dichos documentos.

Se crearán grupos y claves de usuario para llevar un control de las personas que hacen uso del sistema. Se cuenta con tres grupos:

- de consulta
- de altas
- de administradores

El primer grupo tendrá los mínimos privilegios del sistema, las personas a las que se les otorgue claves de consulta, únicamente podrán hacer uso del sistema para otorgar, de manera impresa, el comprobante de inscripción, a las personas que ya hayan sido dadas de alta. Las claves de usuario del grupo de consultas, no podrán dar de alta a algún usuario.

El segundo grupo, el grupo de altas, tendrá otros privilegios en comparación con el grupo de consulta. Las persona, con claves de usuario pertenecientes a este grupo, serán responsable del alta de alumnos a la Unidad de Cómputo, ellos podrán expedir comprobantes de inscripción de los alumnos que la soliciten. También podrán llevar a cabo la actividad del grupo de consulta.

El último grupo, el grupo de Administración, estará encargado de la supervisión, de los reportes del monitoreo de claves y del monitoreo de altas. Este grupo tendrá privilegios totales del sistema, las personas con estas claves podrán realizar, además de los monitoreos, el alta de usuarios y la expedición de comprobantes de inscripción. El número de personas para cada uno de los grupos estará en función de las necesidades de la propia Unidad, con la opción de modificar (eliminar o generar) el número de claves.

El sistema podrá operarse vía Web. Es decir que no será necesaria la instalación del sistema en un equipo de cómputo específico. El sistema completo residirá en uno de los servidores de UNICA, con la finalidad de que sí el equipo asignado para el alta de alumnos falla por cualquier razón, no se detenga el servicio, y sólo se necesite “desplegar” en otro equipo el navegador en la dirección Web asignada para este sistema.

Se deberá llevar una serie de reportes que el sistema generará con los datos de aquellos usuarios dados de alta. Estos reportes deberán tener como mínimo, el nombre completo del usuario, su número de cuenta, su número telefónico y su dirección.

Se generarán estadísticas de los alumnos dados de alta, esto con la finalidad de llevar un seguimiento que nos indique cuál de las carreras que se imparten en la Facultad de Ingeniería hace mayor uso de las instalaciones de la Unidad. Las estadísticas serán:

- Número de alumnos por semestre o año que hicieron uso de las instalaciones de UNICA.
- Número de alumnos dependiendo del semestre que cursan, y de esta forma saber alumnos de qué semestre (1ro, 2do, 3ro, 4to, etc.) hacen mayor uso de las instalaciones.

- Se solicita estadísticas por carrera, es decir, que cantidad de alumno por carrera, del total de la comunidad de la Facultad, hacen uso de los servicios de la Unidad de Cómputo.

Otra parte que es importante mencionar el hecho de poder dar de alta a alumnos que, por cierto motivo, no se encontraran en el registro de alumnos proporcionado por la Unidad de Servicios de Cómputo Administrativos de la Facultad de Ingeniería (USECAD) y del cuál nos valemos para llevar el avance del alumnado de la Facultad, como son tesistas, recursadores y más.

Por último, el comprobante que se le proporcionará al alumno, una vez que concluya el trámite de alta, debe reunir las siguientes características:

- Encabezado
- Nombre completo del alumno
- Número de cuenta del alumno
- Login asignado al alumno
- Contraseña correspondiente
- Contraseña para base de datos (en caso de que el alumno lo requiera)
- Cuenta de correo
- El logotipo correspondientes a UNICA
- La Fecha de expedición

El encabezado debe tener la leyenda “COMPROBANTE PARA EL ACCESO A LAS SALAS DE CÓMPUTO DE UNICA” en caso de ser la primera vez que el usuario obtiene su comprobante.

El comprobante debe contener la fecha de la siguiente forma: UNICAddmmaa. Donde dd corresponde al día, mm corresponde al mes, y aa los dos últimos dígitos del año.

Vg. UNICA230703

que corresponde al 23 de julio del 2003.

En cuanto a los logotipos, deberá aparecer, cuando menos, el de la Unidad de Servicios de Cómputo Académico.

La figura 2.1 muestra el formato básico que tendrá el comprobante de acceso a las Salas de UNICA.



Fig. 2.1 Formato básico del comprobante

2.2- Requerimientos:

Dentro de la ingeniería de software, las fases de análisis de requerimientos y de especificación son señaladas como de particular importancia; primero, porque en ellas se deben definir de manera precisa, completa y sin ambigüedades, las características y requerimientos que deberá satisfacer la futura aplicación; segundo, porque son las etapas en las que las correcciones se pueden realizar a un costo moderado, comparado con el costo que implica corregir la aplicación en fases más avanzadas de su desarrollo.

El análisis de requerimientos facilita al ingeniero especificar la función y comportamiento de los programas, indicar la interfaz con otros elementos del sistema y establecer las ligaduras de diseño que deben cumplir los programas. Este análisis de requerimientos del sistema también permite establecer las características que deberá tener el hardware, dichos requerimientos de hardware dependerán de la ejecución del sistema y del sistema operativo que utilice.

Los Sistemas de Información por computadora normalmente están integrados por muchos componentes. En la mayor parte de los casos, es difícil para los analistas entender todos estos componentes a un mismo tiempo; por lo tanto los investigadores tienen que comenzar con preguntas de tipo general con relación al propósito del sistema, sus entradas y salidas de los procesos incluidos.

En los grandes proyectos de sistemas, varios analistas llevan a cabo una investigación en forma seccionada que la distribuye entre ellos mismos, de manera que cada uno pueda trabajar en forma independiente.

El análisis de requerimientos es la primera etapa de un proyecto de software, en ella se tratan de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

Hay diferentes tipos de requerimientos: de entorno (sistema operativo, sistema gestor de base de datos, sistema de archivos, y más), ergonómicos (interfaz gráfica), funcionales (que debe hacer el sistema), de rendimiento, de tiempo, formato de entrega, y más.

Para la creación global del sistema se necesitan comprender todos los objetivos y necesidades del usuario. En primer lugar, hemos de especificar el comportamiento externo del sistema desde el punto de vista del usuario. Una vez acabado, podemos pensar en la arquitectura general del sistema, en términos de componentes físicos: hardware, software, usuarios y la comunicación entre ellos.

La determinación de los requerimientos es con base en la experiencia, de hablar con los usuarios finales sobre sus necesidades y analizar un sistema software existente.

Comenzaremos a plasmar todos estos conceptos pero abocándonos a las necesidades de la Unidad de Servicios de Cómputo Académico.

Una vez finalizadas las pláticas con los encargados de ofrecer los servicios en la Unidad, con las personas que trabajarán directamente con el sistema, y con los administradores que operan los servidores y están a cargo del alta de usuarios en el sistema; y habiendo reunido los datos correspondientes a las especificaciones dadas por todos los involucrados, se procede al análisis de los requerimientos del sistema.

- Se desea que el sistema sea un sistema Web y que pueda ser operado desde cualquier equipo con conexión a Internet. El sistema deberá estar en un equipo que cuente con una dirección IP, ya que de otra forma no sería posible que estuviera siempre en servicio y con una buena rapidez de respuesta.
- Los datos, que anteriormente se manejaban en archivos, pretenden ser emigrados a una base de datos. Es necesario por tal motivo, el empleo de una Bases de Datos y del software manejador de ésta, el cual además de ser económico, debe ser eficiente y seguro para resguardar los datos.

- Deberá contener una interfaz de usuario, que aparte de ser agradable y sencilla de utilizar, deberá de ser segura con respecto a los datos que se visualicen y que capturen. Además tendrá que ser un software que pueda interactuar con el manejador de la Base de Datos y con el Sistema Operativo que se asigne. Debe ser fácil de “cargar” al seleccionar la dirección Web donde se encuentra.
- Puesto que el sistema deberá de dar de alta las cuentas de los alumnos de manera automática, para que ellos puedan hacer uso de los servicios institucionales de la Unidad, será necesario que el sistema radique directamente en el servidor que proporciona dichos servicios para que de esta forma el acceso sea más rápido y seguro. El sistema operativo se elegirá con base en el sistema instalado en el servidor con el que cuenta la Unidad.
- Será necesaria la recopilación de los datos de todos los alumnos registrados en la Facultad de Ingeniería. No importando la manera en que se nos otorgue, aunque de preferencia, podría ser en formato de texto o directamente en tablas. Pues de esta forma es posible convertirlos, manipularlos y editarlos para que tengan el formato que se va a utilizar para el nuevo sistema.
- Los datos podrán ser visualizados exclusivamente por el personal que la Unidad considere para este fin, de modo que personas ajenas a UNICA no puedan visualizar o alterar la información que se maneja. Estos usuarios estarán registrados por un nombre de usuario y una contraseña que será distinta para cada uno de ellos.
- Por seguridad los equipos que hagan uso de SIASU serán registrados en el servidor que contendrá la información para que solo desde dicho equipos se pueda acceder al sistema.

2.3 Metodología empleada:

Con base en las especificaciones dadas en la Unidad de Servicios de Cómputo Académico por los jefes de las respectivas áreas y por aquellas personas (trabajadores) que están involucradas en la atención y trato con los usuarios y tomando en cuenta el análisis llevado a cabo sobre los requerimientos del sistema solicitado, se optó como metodología a seguir el Ciclo de Vida Clásico Modificado. Esta metodología lleva las mismas fases que el ciclo de vida clásico pero tiene la ventaja de permitirnos el traslape entre fases, lo que se traduce en rapidez, y una retroalimentación, lo que de cierta manera es un apoyo para poder entregar un producto, que aunque no es un producto terminado, nos ayuda a tener un esquema de lo que se nos solicita para que sea analizado por el usuario. De esta forma el usuario no tiene que esperar hasta que el Sistema esté en las últimas etapas del ciclo para poder ver avances en el proyecto.

En las líneas siguientes se describe el ciclo de vida clásico y, de manera detallada, las fases que lo componen. Posteriormente se indican las ventajas que el ciclo de vida clásico modificado tienen sobre éste.

2.3.1 Ciclo de vida clásico

Esta metodología aplica una serie de pasos en forma secuencial, donde cada fase requiere información de entrada, procesos y resultados que serán utilizados en la siguiente fase.

El ciclo de vida clásico esta compuesta por las siguientes fases:

1. Planificación del Sistema.
2. Análisis.
3. Diseño.
4. Codificación.

5. Pruebas.
6. Implantación.
7. Mantenimiento.

2.3.1.1- Planeación del Sistema

En esta etapa se considera si el Sistema solicitado es factible de realizarse; ya que uno de los principales propósitos es aclarar los objetivos, problemas o necesidades y restricciones.

Para saber si un Sistema es factible de desarrollarse es necesario realizar un estudio de factibilidad y para ello se deben considerar los siguientes tres aspectos:

Factibilidad Técnica: Estudia si el proyecto puede desarrollarse con el software existente o es necesario utilizar una nueva tecnología.

Factibilidad Económica: Investiga si los costos se justifican con los beneficios que se obtienen.

Factibilidad Operacional: Investiga si será utilizado el sistema, como para obtener beneficio.

2.3.1.2-Análisis

En esta parte del sistema se hace la recopilación de los requisitos necesarios para el desarrollo del sistema, la cual es indispensable para comprender perfectamente los requisitos del software y de esta manera también determinar los requerimientos de hardware.

2.3.1.3-Diseño

El diseño de un sistema de información se refiere a la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis.

Para hacer más sencilla esta tarea se puede hacer un esquema del formato o pantallas que se espera aparezcan cuando el sistema este terminado, dicho esquema se puede elaborar en papel o en la pantalla de alguna terminal.

En esta fase también se indican los datos de entrada, los que serán calculados y los que deben ser almacenados. Los procedimientos que indiquen como procesar los datos y como producir las salidas.

También es la etapa en la que se elaboran los documentos que contienen las especificaciones de diseño el cual se representa mediante diagramas, tablas y símbolos especiales.

Es importante que la información del sistema este bien detallada y elaborada para comenzar con la fase de desarrollo de software.

En resumen podemos decir que el diseño del software es realmente un proceso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

2.3.1.4-Codificación

El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede hacerse mecánicamente. Es poner en marcha los conceptos de programación para obtener el sistema solicitado.

2.3.1.5-Implantación

Es el proceso que da cabida a la instalación y verificación, instruir a los usuarios, instalar la aplicación y construir todos los archivos de datos necesarios para utilizar el Sistema, procurando que el uso inicial del Sistema se encuentre libre de problemas.

2.3.1.6- Pruebas

Durante esta fase el sistema puede ser empleado de manera experimental para asegurarse que el software no tenga fallas, es decir que funcione de acuerdo a las especificaciones y en la forma en que los usuarios lo necesitan. La prueba se centra en la lógica interna del software, asegurándose que todas las sentencias sean probadas, alimentando sus entradas con conjuntos de datos de prueba para su procedimiento y después analizando los resultados.

2.3.1.7-Mantenimiento

En el mantenimiento de software se aplica cada uno de los pasos derivados del ciclo de vida a nuestro Sistema.

2.3.2 Ciclo de vida clásico modificado

- Fue creado para permitir retroalimentación y traslape entre fases
- Es un modelo interactivo y no lineal.
- Para facilitar la terminación de las metas y tareas, es normal congelar partes del desarrollo después de cierto número de iteraciones.
- Se agregaron pasos de verificación (verificar que el sistema es correcto) y validación (verificar que el sistema cumple con los requerimientos del cliente)

2. 5- Selección del Software

En todo diseño de sistemas es necesario analizar bien el problema y los objetivos a alcanzar, ya que de esta forma podemos seleccionar las herramientas adecuadas, y cubrir las necesidades del Sistema; es por ello que en este capítulo se recopila la información necesaria para justificar el diferente software que se utilizará en el desarrollo del Sistema.

2.5.1 Análisis de requerimientos de software.

En este punto se darán los requerimientos del software utilizado para el desarrollo del sistema, así como sus especificaciones y definición. Se analizará las principales propiedades del software propuesto y los beneficios que cada uno de ellos nos ofrece tanto de forma individual como su desempeño en conjunto.

2.5.2 Sistema Operativo

Podemos encontrar diversas definiciones de lo que es un Sistema Operativo por lo que en las líneas siguientes se detallarán algunas de estas:

- Sistema operativo, se le llama al software del sistema que administra y controla las actividades y recursos de la computadora. El sistema operativo de una computadora comprende todo aquel software que le permite funcionar como tal. Por tanto, este software está integrado como un conjunto de herramientas (programas de utilería, paquetes de administración o control), intérpretes de comandos, y su núcleo (administrador de memoria, administrador y controladores de entrada y salida a dispositivos periféricos, y el administrador de procesos), con lo que el operador, usuario y la computadora misma pueden administrar los recursos a su alcance y controlar todo programa que se ejecute en éste.

- Un Sistema Operativo es un conjunto de programas que controla la ejecución de programas de aplicación y actúa como una interfaz entre el usuario y el hardware de una computadora, esto es, un Sistema Operativo explota y administra los recursos de hardware de la computadora con el objeto de proporcionar un conjunto de servicios a los usuarios del sistema. Entonces, el objetivo principal de un Sistema Operativo es que el hardware de la computadora se emplee de manera eficiente, y el objetivo secundario es lograr que el sistema sea lo más sencillo de manejar por el usuario.

Tomando como base las definiciones anteriores podemos resumir que el Sistema Operativo es un conjunto de programas encargados de brindar al usuario una forma sencilla y agradable para operar, codificar y presentar las ordenes al procesador central para que éste las interprete y realice las tareas necesarias y específicas para completar una orden.

En la actualidad existen varios Sistema Operativos con excelentes características en cuanto a rapidez, estabilidad, escalabilidad, potencia y sobre todo confiables. Tales son los casos de los Sistema Operativos Windows 2000, Windows NT, UNIX (en todas sus distribuciones), Mac OS X (sistema operativo propio de los equipos Apple), y Linux.

Se analizarán los Sistema operativos Windows (en su versión 2000) y Linux, por ser las plataformas más comunes en instituciones educativas, y de mayor demanda en la Facultad.

El análisis entre los Sistemas Operativos Linux y Windows esta enfocado en su desempeño y utilización como servidores Web, ya que es en este ámbito en el cuál estará enfocado este sistema.

La plataforma que se elija para hospedar el sitio Web dependerá de varios aspectos; aquí se tratarán los que influyen directamente en el desempeño del servicio de Web.

Facilidad de Uso:

Tradicionalmente hay quienes piensan que Linux es una plataforma de difícil manejo y solo para usuarios muy avanzados, pero la realidad es otra; y si lo que se desea es hospedar un

sitio Web, tanto Linux como Windows son una buena opción por esto. Hoy en día no hay diferencias en la facilidad de utilización entre estas plataformas.

Estabilidad:

El consenso en la industria es que Linux, a pesar de estar en una etapa de creciendo, aunque de forma muy acelerada, es más estable que Windows, era cierto pero eso hasta antes de la llegada de Windows 2000, ya que este ha demostrado ser de lo más estable comparado con sus antecesores. Se ha trabajado con ambos sistemas y ambos cuentan con buena estabilidad, por lo que no se ha encontrado diferencias significativas.

Velocidad:

No se encontró diferencias reales de velocidad entre estas dos plataformas. Lo que en cierta forma incrementa la velocidad es el hardware que conforma el equipo así como los recursos que consuman otro software instalado.

Funcionalidad:

En este punto es donde se pueden hallar diferencias en las dos plataformas. Si decide utilizar el sistema operativo Linux será necesario utilizar PHP y MySQL o PostgreSQL y en el caso de Windows 2000 se requeriría de ASP y SQL.

Costo:

En este rubro el sistema operativo Linux tiene puntos a favor, ya que la mayoría de los software son de libre utilización, por lo que los costes son casi nulos, a diferencia del sistema operativo Windows donde el precio de las licencias tienen un costo considerable al igual que la mayoría de las aplicaciones que sobre este sistema trabajan.

La siguiente figura (Fig. 2.1) muestra una tabla comparativa de las características soportadas por ambos Sistema Operativos.

Características Soportadas		
Característica	Linux	Windows
MS FrontPage 2000,2002	•	•
Flash	•	•
Shockwave	•	•
RealAudio/vides	•	•
Cgi-Script	•	•
Perl	•	•
PHP	•	•
SSH (Secure Telnet)	•	
MySQL	•	
PostgreSQL	•	
Miva	•	
ASP (active server pages)		•
MS Access		•
MS SQL		•
Visual Basic Scripts		•
Windows Media		•
Las siguientes características son soportadas en ambas plataformas Unix y Windows®2000		
-Mailing List -Estadísticas de visitas Webstats -Web mail		
Información Técnica para usuarios avanzados.		
	Linux	Windows
Web Server	Apache	IIS (Internet Information Server)

Fig. 2.1 Tabla comparativa

2.5.3 Sistema Manejador de Bases de Datos

El funcionamiento del Data Base Manager System (DBMS) está muy relacionado con el funcionamiento del Sistema Operativo, especialmente con la parte del sistema de comunicaciones, ya que utiliza las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (el cuál puede estar utilizando una conexión directa en consola o de manera remota en una terminal).

Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica.

El objetivo primordial de un sistema manejador de base de datos es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los usuarios y la base de datos.

El DBMS realiza la traducción entre los diferentes esquemas de la Base de Datos. Si un usuario quiere acceder a los datos, el manejador comprobará su esquema externo para averiguar a qué datos tiene privilegios de acceso; posteriormente estudia el esquema conceptual completo, a continuación al esquema físico para saber cómo trabajar con ellos y finalmente proporcionará al usuario los datos solicitados.

Funciones del Data Base Manager System (DBMS)

El DBMS es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos, esta compuesto por tres funciones:

DDL: Lenguaje de Definición de Datos

DML: Lenguaje de Manipulación de Datos

SQL: Lenguaje de Consulta Simple.

Función de definición: Permite describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del DBMS denominada lenguaje de definición de datos (*DDL - Data Definition Language*).

Función de manipulación: Permite buscar, añadir, suprimir y modificar los datos de la Base de Datos. Esta función es realizada por una parte del DBMS denominada lenguaje de manipulación de datos (*DML - Data Manipulation Language*).

Función de utilización: Incluye otras funcionalidades: modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, arranque, protección frente a accesos no autorizados, estadísticas de utilización, y más.

El DBMS se compone de una serie de módulos que son los encargados de realizar las funciones antes mencionadas:

1. El Compilador de DDL

Se utiliza para crear y mantener la base de datos y los elementos que contiene a nivel externo, lógico e interno. Es propio de cada DBMS. Permite definir entidades, identificadores (claves), atributos, interrelaciones, autorizaciones de acceso, restricciones de integridad, etc. A nivel interno facilita la definición del espacio físico, longitud de los campos, representación de los datos y caminos de acceso.

2. El Precompilador DML (Data Manipulation Language)

Lenguaje de manipulación de datos, se utiliza para la actualización y consulta de los datos almacenados en la base de datos. Permite añadir, seleccionar, suprimir o modificar los datos respetando las reglas establecidas por el DDL.

3. Procesador de Consultas

Permite al usuario consultar los datos sin necesidad de construir un programa de aplicación. Cuenta con un Optimizador de DML para mejorar esas consultas.

Una vez definidas las características generales de los Manejadores de Bases de Datos, daremos paso a la selección de aquel que se hará cargo de la administración de la Base Datos que se desarrollara en esta tesis.

Se cuenta con varias opciones para la elección del manejador. Se compararán y se hará, un análisis de ventajas y desventaja aquellos de mayor uso dentro y fuera de la Facultad de Ingeniería.

MySQL:

Este manejador fue diseñado para obtener una respuesta rápida, es decir su principal objetivo de diseño fue la VELOCIDAD, aunque se sacrificaron características esenciales en sistemas más robustos, como lo es la seguridad.

Otra característica importante es que consume MUY POCOS RECURSOS, tanto de CPU como de memoria. Posee licencia GPL (desplegar iniciales) a partir de la versión 3.23.19.

Entre sus ventajas se encuentran:

- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir “queries” y más.
- Mejores utilidades de administración (backup, recuperación de errores).
- Aunque se cuelgue, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.

- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL responde mejor que Postgres a la hora de modificar o añadir campos a una tabla cuando esto se desea hacer "al momento".

Desventajas de MySQL:

- No soporta transacciones ni **subselects**.
- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

En cuanto a su estabilidad:

- MySQL pierde puntos en cuanto la estabilidad en lapsos semi-largos (de 30 a 60 días).
- Lo que pierde en su robustez lo gana en la integridad de sus datos. Es difícil que MySQL llegue a corromper los datos, por lo que la información está segura.

PostgreSQL:

Postgres intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Está sustentado bajo licencia BSD.

Como ventajas encontramos que:

- PostgreSQL soporta hasta tres veces la carga de trabajo que MySQL, antes de lanzar mensajes de error, tomando en cuenta la misma combinación de hardware/OS.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).

- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.
- La seguridad es algo primordial en este manejador. Los datos no solo se mantienen íntegros, si no que la consistencia en general (de toda la estructura de base de datos) se mantiene segura.

Entre los inconvenientes que presenta están:

- Consume bastantes recursos y carga más el sistema.
- Límite del tamaño de cada fila de las tablas a 8k!!! (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- A la hora de consultas es de 2 a 3 veces más lenta que MySQL.
- Menos funciones en PHP.

Se puede concluir a la comparación entre MySQL y PostgreSQL que:

MySQL junto con Apache y PHP forman un buen equipo para servir páginas Web con contenido dinámico, discusiones, noticias, etc.. En general, sistemas en los que la velocidad y el número de accesos concurrentes sea algo primordial, y la seguridad no sea muy importante. En cambio, para sistemas más serios en las que la consistencia de la BD sea **fundamental** (BD con información realmente importante, bancos, etc.) PostgreSQL es una mejor opción pese a su "lentitud". Al igual que MySQL, puede hacer equipo con Apache y PHP, para crear páginas Web dinámicas y seguras.

Aunque **ORACLE** es la mejor opción para ser utilizado en este y otros sistema, pues de las características descritas en este capítulo éste supera a cualquier manejador de base de datos en todos los aspectos, se optó por descartarlo por su alto costo. Ya que, aunque la Facultad puede pagar la licencia para la utilización de este software, como ingeniero, uno trata de resolver los problemas con el mínimo de presupuesto y limitado en recursos.

2.5.4 Manejador de la Interfaz Grafica.

En este punto se tiene varios tipos de software, entre estos encontramos ASP (si la plataforma sobre la que vamos a trabajar es Windows) o PHP (que, aun que ya se tiene una versión para Windows, es propio de plataformas Unix-Linux y es donde mejor se desenvuelve). Otra opción estupenda sería la utilización del software Java, ya que no importa bajo qué plataforma se trabaje, el software java es una muy buena opción.

Se dará una breve explicación de cada uno de los programas para interfaces graficas, resaltando sus ventajas así como sus desventajas y los requerimientos mínimos de software para su buen desempeño.

Descripción y Características de ASP

Páginas Active Server (ASP, Active Server Pages) es un entorno para crear y ejecutar aplicaciones dinámicas e interactivas en la Web.

Se puede combinar páginas HTML, secuencias de comandos y componentes ActiveX para crear páginas y aplicaciones Web interactivas.

Las páginas ASP comienzan a ejecutarse cuando un usuario solicita un archivo **.asp** al servidor Web a través de su navegador. El servidor Web llama al software ASP, que lee el archivo solicitado, ejecuta las secuencias de comandos que encuentre y envía los resultados al navegador del cliente.

Puesto que las secuencias de comandos se ejecutan en el servidor, y no en el cliente, es el servidor el que hace todo el trabajo necesario para generar las paginas que se envían al navegador. Las secuencias de comandos quedan ocultas a los usuarios, estos solo reciben el resultado de la ejecución en formato HTML.

De esta forma desaparece el problema de si el cliente puede o no ejecutar sentencias de comandos; el servidor Web solo envía el resultado en código HTML estándar interpretable por cualquier navegador.

Los archivos .asp son archivos de texto normales, no es necesario ningún editor especial para crearlos, puede usarse cualquier editor que genere código ascii.

Software necesario para la ejecución de Páginas Active Server.

Para la implantación de un servidor Web que soporte ASP el software necesario es, si lo que estamos configurando es un servidor de alto rendimiento:

- WINDOWS NT 4.0 , 2000 o XP
- IIS 4.0 o 5.0 (INTERNET INFORMATION SERVER 4.0 - 5.0) Ó IIS3.0 + ASP.EXE

Para desarrollo o sistemas Intranet los requerimientos son más sencillos:

WINDOWS 95 + PERSONAL WEB SERVER 1.0 + ASP.EXE

WINDOWS 98 o Millenium + PERSONAL WEB SERVER 4.0

Nota: Personal Web Server 4.0 esta incluido en algunas de las distribuciones de Windows 98 en el directorio ADD-ONS\PWS

Descripción y Características de PHP

El lenguaje PHP (Acrónimo de PHP Hipertext Preprocessor) es un lenguaje para el desarrollo de páginas Web que se instala y trabaja directamente en un servidor Web. Es un lenguaje interpretado de alto nivel enfocado al trabajo Web de páginas seguras.

La programación que se desarrolla en PHP puede convivir con código HTML el cual puede ser escrito en cualquier editor de palabras, no es necesario de algún editor especial para escribir este tipo de código.

Los navegadores, sin importar bajo qué plataforma se encuentren, no tienen ningún problema al trabajar con código PHP, y puesto que éste se interpreta directamente en el servidor Web para generar la página HTML solicitada, el cliente se olvida de ejecutar programa alguno.

PHP ha sido desarrollado principalmente para usarse conjuntamente con el servidor Web “Apache”, en cualquiera de las plataformas que éste soporta. De hecho, está implementado como una aplicación en los servidores UNIX y Linux de forma que aprovecha la máxima integración con el propio servidor.

Una de las características principales de PHP es el API para construir aplicaciones y conectar Bases de Datos. PHP soporta muchos tipos de manejadores de Bases de Datos de forma nativa, de esta manera puede ser configurado para interactuar con la base de datos que se tenga en el servidor Web o con el servidor de Bases de Datos con el que se cuente. PHP es muy utilizado sobre todo con los manejadores PostgreSQL y MySQL, bases de datos que trabajan bajo licencias de software libre.

Todas las aplicaciones de PHP se ejecutan en el servidor, por eso nos permite acceder a los recursos que éste tiene, como por ejemplo sus compiladores y su manejador de Base de Datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador, siendo este último una página HTML con características de seguridad.

Software necesario para la ejecución de PHP.

Para la implantación de un servidor Web que soporte PHP no es necesario contar con algún software especial, sólo se necesita tener el software PHP y configurarse como módulo del servidor Web apache, si lo que estamos configurando es un servidor de alto rendimiento, es necesario modificar ciertas “directivas” y algunos archivos, tanto del servidor Web como del manejador de la interfaz grafica y realizar una buena programación.

La figura 2.2 muestra el esquema de cómo trabaja PHP en la creación de paginas Web.



Fig. 2.2 Manera en que trabaja PHP.

2. 6- Elección del software

El sistema trabajará apoyado en los protocolos TCP/IP, siguiendo una arquitectura Cliente / Servidor para la Web. La arquitectura de Internet se basa en el modelo cliente-servidor, utilizando un sistema de hipertexto, el cual permite enlazar y llevar en un orden, documentos afines, todo en el mismo servidor y/o a lo largo de toda la Internet.

Por lo comentado en las líneas anteriores, el software propuesto para la realización de este proyecto es el siguiente:

- LINUX, como Sistema Operativo sobre el que residirá y trabajará el sistema.
- POSTGRES, como manejador de Base de datos, por su seguridad.
- PHP, como Lenguaje de programación para la interfaz grafica de páginas Web, por ser un lenguaje que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente, lo que se traduce en seguridad.

Diseño del Sistema

Capítulo

3

Una vez evaluadas las características de los diferentes ciclos de vida de los sistemas y elegido el software sobre el cuál se basará esta tesis, procede el diseño del sistema.

Los rubros que se tocaron para el desarrollo de aplicaciones web fueron:

El Sistema Operativo: En el caso de este sistema fue Linux, ya que cubría con los tópicos tomados en cuenta como son, la seguridad, su administración, su soporte para aplicaciones extras, su rapidez y precio.

El Servidor Web: Se opto por manejar el servidor web Apache, en gran parte por la licencia que maneja, y por que puede ser configurado para trabajar en una plataforma Linux de manera segura.

El lenguaje de programación: La elección fue PHP por sus características y la facilidad que nos brinda al trabajar directamente en el servidor, así el usuario final estará tranquilo al no tener que instalar o configurar algo en su computadora. para poder trabajar. Es posible implementar conexiones a distintos manejadores de bases de datos.

El manejador de Base de Datos: Se decidió utilizar PostgreSQL por ser un software de licencia libre y sobre todo muy seguro, asimismo, una buena configuración con el servidor apache y la herramienta de desarrollo PHP hacen de este manejador una buena opción.

3.1- *Formulación del Modelo*

El análisis de la formulación del modelo se analizará a través de la técnica de diagrama de flujo de datos (DFD). Este modelo representará de forma gráfica el flujo de la información y las modificaciones que se aplican a los datos al moverse desde la entrada del sistema hasta la salida del mismo. En el apéndice II se detallan los conceptos y definiciones de esta técnica.

Con base en los requerimientos y limitantes del sistema se procede a diseñar los esquemas o diagramas que muestran la forma en como se estructurará este sistema.

La figura 3.1 señala las claves que se utilizaron en cada uno de los Diagramas de Flujos.

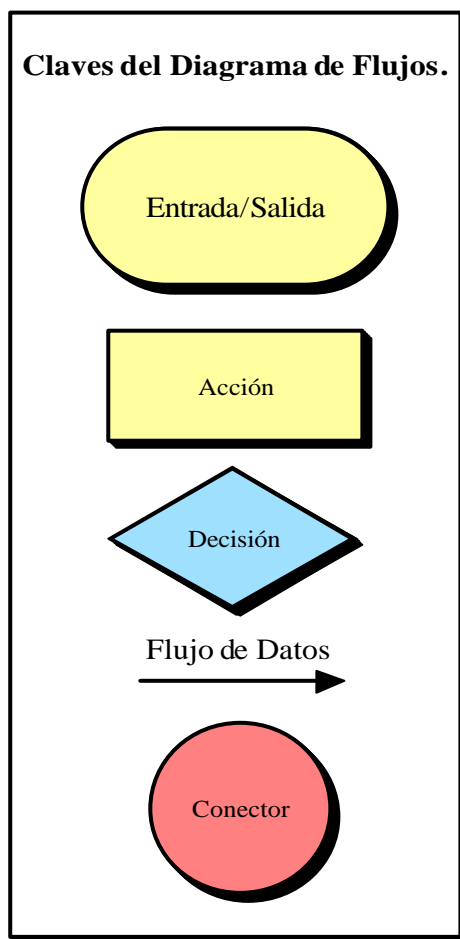


Fig. 3.1

La figura 3.2 muestra el diagrama de flujos para la elección de los grupos de usuarios que puede hacer uso del sistema.

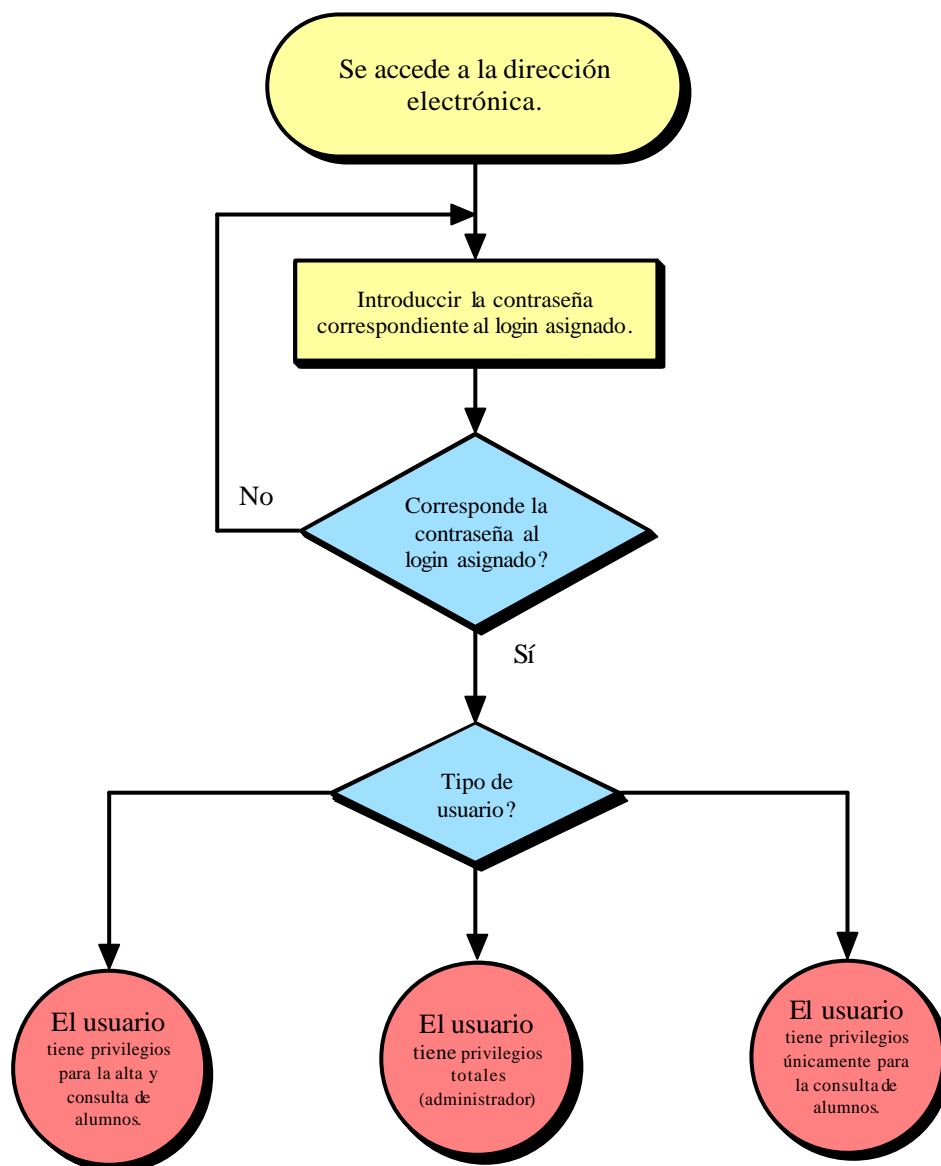


Figura 3.2 Elección del grupo.

Como puede apreciarse, se señalan los tipos de grupo que van a poder, dependiendo de los privilegios asignados, hacer uso del sistema. Este es el esquema base, ya que dependiendo de los privilegios asignados a los grupos, estos seguirán uno de los esquemas siguientes.

Procesos para el alta de alumnos.

La siguiente figura (Fig. 3.3) muestra el diagrama de flujo de datos del proceso a seguir para el alta de alumnos. En este caso tanto el grupo de Altas como de Administración se basarán en este esquema.

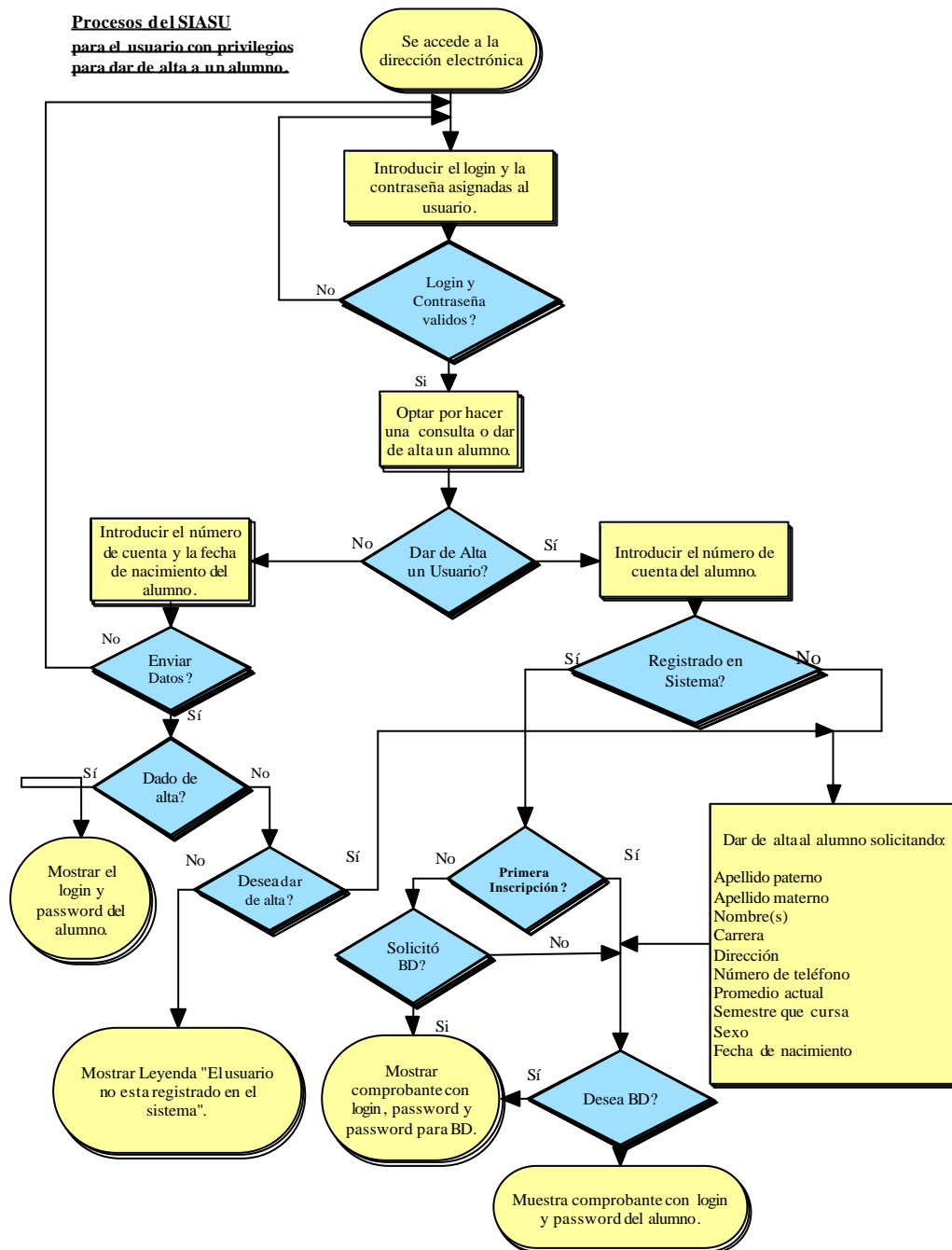


Figura 3.3 Grupo de Altas.

Procesos para la consulta de alumnos.

La figura 3.4 muestra el diagrama de flujo de datos del proceso a seguir para la consulta de datos de alumnos que ya efectuaron su alta. Este esquema se estructura para todos los grupos, de modo que cualquiera de estos pueda consultar datos como nombre de usuario y contraseña de los alumnos que lo soliciten.

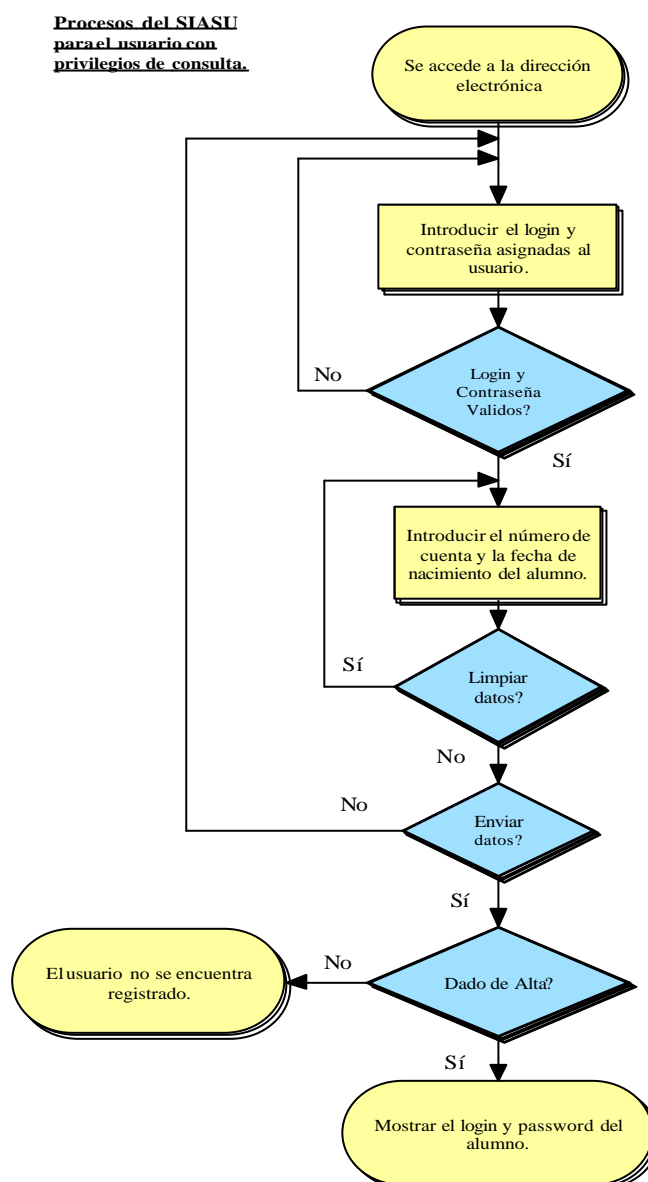


Figura 3.4 Grupo de Consulta.

Procesos para el grupo de administradores.

El grupo de administradores podrá dar de altas a alumnos y además llevará el monitoreo de la fluencia de alumnos y de los usuarios que hacen uso del sistema así como del buen funcionamiento del mismo. Además, este grupo esta facultado a dar de alta y baja a usuario, según la Unidad de Cómputo lo requiera.

El modulo creado para el grupo de administración podrá realizar la consulta de contraseñas y nombre de usuario de los alumnos que así lo requieran y lo soliciten.

Podrá llevar a cabo otras tareas como la revisión de la fluencia de alumnos inscritos, esto con base en las estadísticas que se encuentran en este modulo y que de forma grafica indican el numero de alumnos inscritos hasta este momento.

El registro de las consultas se llevará en este modulo. El grupo de administradores llevará un registro de los usuarios que han realizado consultas y a que alumnos han sido referidas las consultas. De igual forma, se puede apreciar si un alumno a solicitado consultar sus datos y que usuarios lo han llevado a cabo.

La figura 3.5 muestra el diagrama de flujo de datos para este grupo.

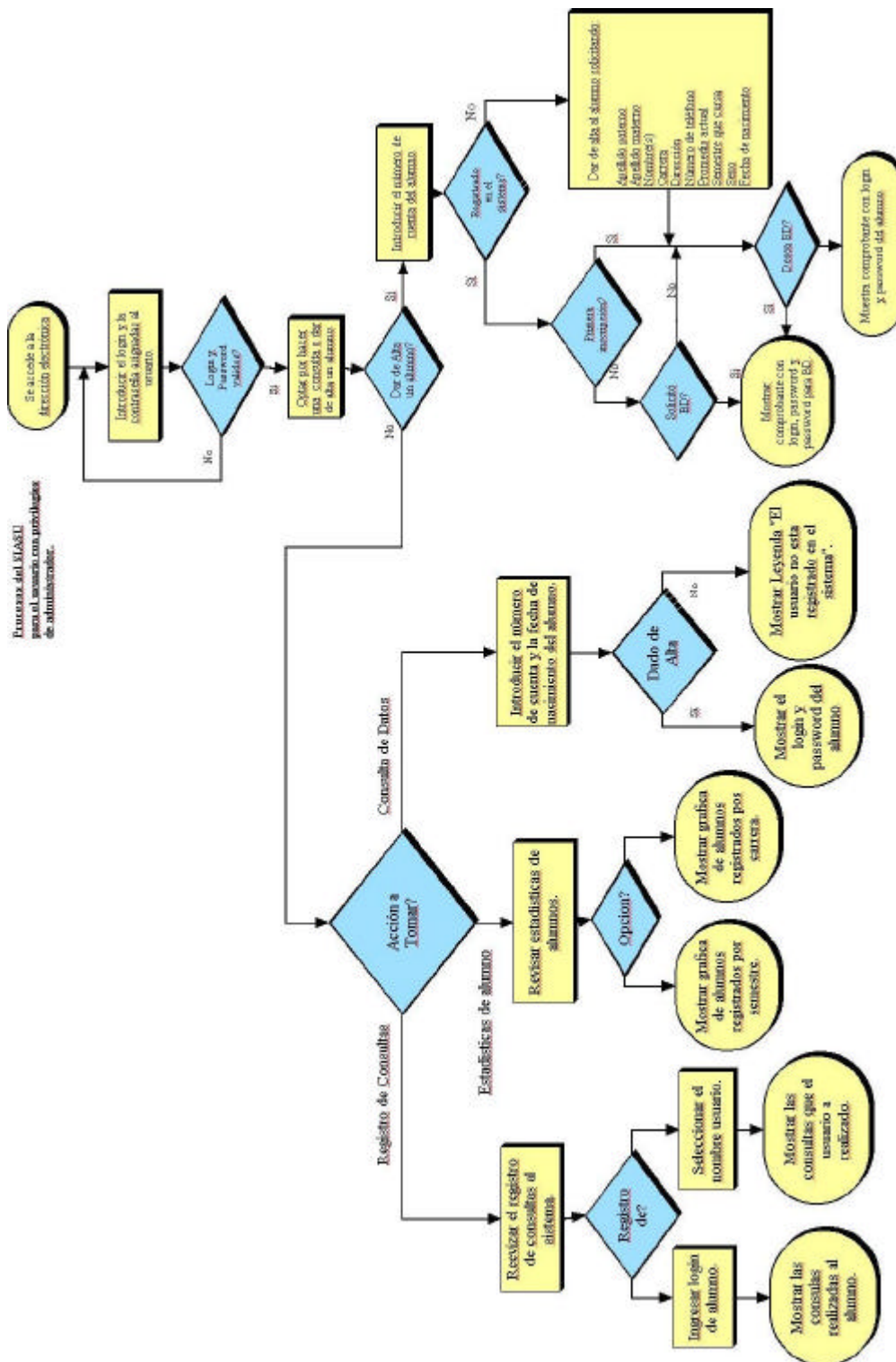


Figura 3.5 Diagrama del Grupo de Administradores.

3.2- *Diseño de Datos*

El diseño de datos consiste en definir las estructuras de datos a construir, dependiendo de la herramienta propuesta. El diseño, por tal, consiste en determinar las tablas que representarán correctamente las entidades a definir.

En principio se cuenta con 4 entidades que definen la base de datos de SIASU, estas se muestran en la tabla 3.1 con una breve descripción de las mismas.

ENTIDAD	DESCRIPCIÓN
Consultas	Información referente a personas que hicieron uso del sistema para consultar datos, así como del usuario que solicitó la consulta.
Verificación	Datos del personal que hace uso del sistema
Base_sistema	Posee los datos de los alumnos inscritos en la Facultad de Ingeniería.
Base_sistema_anterior	Cuenta con los datos de alumnos registrados el año escolar anterior.

Tabla 3.1 Entidades principales

Tomando estas entidades se procede a crear el diagrama que las relacione de modo que el sistema pueda trabajar sin redundancia de información o pérdida de la misma. Las figuras 3.6 y 3.7 muestran los diagramas relacionales de SIASU. La primera muestra el modelo físico del diagrama relacional, donde es posible apreciar que se compone principalmente por cuatro tablas las cuales se muestra en la figura 3.7. A este esquema se le conoce como modelo lógico de la base de datos.

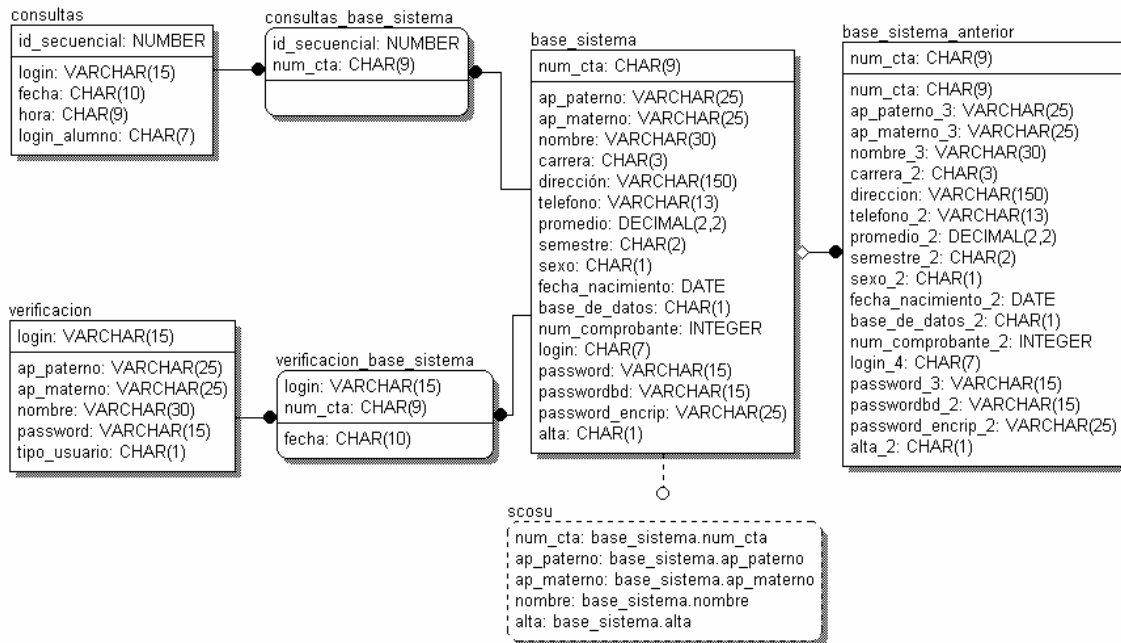


Figura 3.6 Modelo Relacional

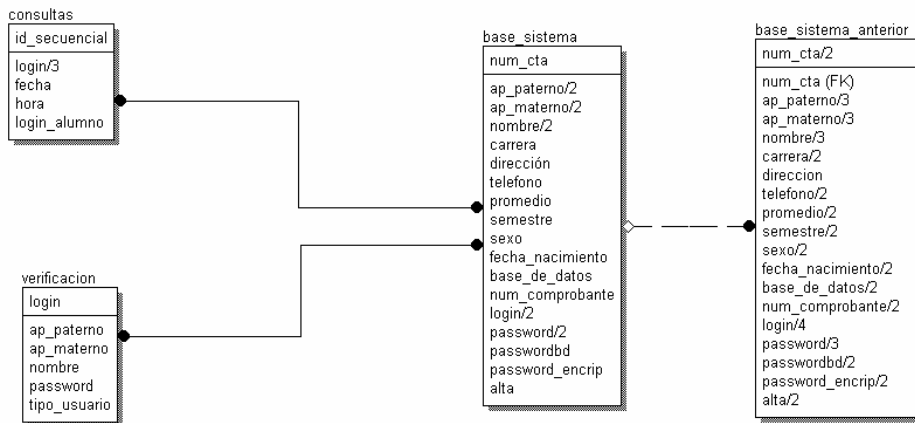


Figura 3.7 Modelo Lógico

3.3 Diccionario de Datos

El Diccionario de Datos del SIASU recopila los nombres de las entidades y sus atributos, así como las características de cada uno de ellos. En el apéndice II (creación de una base de datos) se describe a detalle el concepto de diccionario de datos.

Entidad	Campo	Llave	Nombre Atributo	Tipo	Longitud	Nulo
B A S E - S I S T E M A	1	PK	Num_cta	CHAR	9	NO
	2		Ap_paterno	VARCHAR	25	NO
	3		Ap_materno	VARCHAR	25	SI
	4		Nombre	VARCHAR	30	NO
	5		Carrera	CHAR	3	NO
	6		Dirección	VARCHAR	150	NO
	7		Teléfono	VARCHAR	13	SI
	8		Promedio	REAL	4	SI
	9		Semestre	CHAR	4	NO
	10		Sexo	CHAR	1	NO
	11		Fecha_nacimiento	DATE	aaaammdd	NO
	12		Base_datos	BOOL	FALSE	NO
	13		Num_comprobante	INTEGER	SERIAL	NO
	14	UNIQ	Login	CHAR	7	NO
	15		Password	VARCHAR	15	NO
	16		Passwordbd	VARCHAR	15	NO
	17		Alta	BOOL	FALSE	NO
	18		Password_des	VARCHAR	40	NO

Tabla 3.2 Diccionario de Datos para la entidad base_sistema

Esta entidad lleva la misma estructura que la de base_sistema y se debe a que en ésta se encontrarán los alumnos que un semestre atrás hicieron uso de los laboratorios y de los servicios institucionales de UNICA, de modo que si poseyeran documentación en sus cuentas, nos basaríamos de esta tabla para conservar su nombre de usuario al menos hasta que dejara de utilizar los servicios de UNICA.

Esta es la entidad principal del sistema y es la que posee la información del alumnado.

Entidad	Campo	Llave	Nombre Atributo	Tipo	Longitud	Null	
B A S E - S I S T E M A - A N T E R I O R	1	PK	Num_cta	CHAR	9	NO	
	2		Ap_paterno	VARCHAR	25	NO	
	3		Ap_materno	VARCHAR	25	SI	
	4		Nombre	VARCHAR	30	NO	
	5		Carrera	CHAR	3	NO	
	6		Dirección	VARCHAR	150	NO	
	7		Teléfono	VARCHAR	13	SI	
	8		Promedio	REAL	4	SI	
	9		Semestre	CHAR	4	NO	
	10		Sexo	CHAR	1	NO	
	11		Fecha_nacimiento	DATE	aaaammdd	NO	
	12		Base_datos	BOOL	FALSE	NO	
	13		Num_comprobante	INTEGER	SERIAL	NO	
	14	UNIQ		Login	CHAR	7	NO
	15			Password	VARCHAR	15	NO
	16			Passwordbd	VARCHAR	15	NO
	17			Alta	BOOL	FALSE	NO
	18			Password_des	VARCHAR	40	NO

Tabla 3.3 Diccionario de Datos para la entidad base_sistema_anterior

Entidad	Campo	Llave	Nombre Atributo	Tipo	Longitud	Null
CONSULTAS	1	PK	Id_secuencial	INTEGER	SERIAL	NO
	2		Fecha	CHAR	10	NO
	3		Usuario	VARCHAR	15	NO
	4		Hora	CHAR	9	NO
	5		Login_alumno	CHAR	7	NO

Tabla 3.4 Diccionario de Datos para la entidad consultas

Entidad	Campo	Llave	Nombre Atributo	Tipo	Longitud	Null
VERIFICACION	1	PK	login	CHAR	9	NO
	2		ap_paterno	VARCHAR	25	NO
	3		ap_materno	VARCHAR	25	SI
	4		nombre	VARCHAR	30	NO
	5		password	VARCHAR	15	NO
	6		tipo_usuario	CHAR	1	NO

Tabla 3.5 Diccionario de Datos para la entidad verificacion

Esta última entidad es en la que se apoyará el sistema SCOSU para trabajar.

Entidad	Campo	Llave	Nombre Atributo	Tipo	Longitud	Null
SCOSU	1		num_cta	CHAR	9	NO
	2		ap_paterno	VARCHAR	25	NO
	3		ap_materno	VARCHAR	25	NO
	4		nombre	VARCHAR	30	NO

Tabla 3.6 Diccionario de Datos para la vista SCOSU

3. 4 Creación de la Base de Datos de SIASU

Una vez que se tiene el modelo relacional es necesario pasarlo a un diseño de base de datos relacional. Siguiendo la estructura y sintaxis que maneja PostgreSQL, se crean las tablas de nuestra base de datos. La figura 3.8 indica la forma en que se crearon algunas de las tablas, así como el tipo de atributo que se maneja.

```
CREATE TABLE base_sistema (  
    num_cta          CHAR(9) PRIMARY KEY,  
    ap_paterno      VARCHAR(25) NOT NULL,  
    ap_materno      VARCHAR(25),  
    nombre          VARCHAR(30) NOT NULL,  
    carrera         CHAR(3) NOT NULL,  
    direccion       VARCHAR(150) NOT NULL,  
    telefono        VARCHAR(13),  
    promedio        REAL,  
    semestre        CHAR(2) NOT NULL,  
    sexo            CHAR(1) NOT NULL,  
    fecha_nacimiento DATE NOT NULL,  
    base_de_datos   BOOLEAN NOT NULL DEFAULT false,  
    num_comprobante INTEGER NOT NULL,  
    login           CHAR(7) NOT NULL UNIQUE,  
    password        VARCHAR(15) NOT NULL,  
    passwordbd      VARCHAR(15) NOT NULL,  
    password_encrip VARCHAR(25),  
    alta            BOOLEAN NOT NULL DEFAULT false  
);  
  
CREATE TABLE verificacion (  
    login           CHAR(9) PRIMARY KEY,  
    ap_paterno      VARCHAR(25) NOT NULL,  
    ap_materno      VARCHAR(25),  
    nombre          VARCHAR(30) NOT NULL,  
    password        VARCHAR(15) NOT NULL,  
    tipo_usuario    CHAR(1) NOT NULL  
);  
  
CREATE TABLE consultas (  
    id_secuencial   SERIAL PRIMARY KEY,  
    login           VARCHAR(15) NOT NULL UNIQUE,  
    fecha           CHAR(9) NOT NULL,  
    hora            CHAR(9) NOT NULL,  
    login_alumno    CHAR(7) NOT NULL  
);
```

Fig. 3.8 Script para la creación de algunas tablas de SIASU

Desarrollo del Sistema

Capítulo

4

4. 1- Manejo de código

Por un lado tenemos el HTML que se desarrolla en el ámbito de una página web, cuando la página se está viendo en la pantalla de los usuarios, es decir, en los navegadores. En estos casos, para realizar cualquier tipo de efecto o interactividad en la página tenemos como recurso al navegador, por eso se llama de cliente.

Por otro lado, existen una serie de lenguajes que se basan en el servidor para ejecutar sus Scripts, al igual que la programación del cliente se basa en el navegador. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los Scripts y genera una página resultado, que envía al cliente. La página resultado contiene únicamente código HTML, por lo que puede ser interpretada por cualquier navegador sin lugar a errores, independientemente de su versión.

Esta independencia del navegador ya es una ventaja significativa con respecto a la programación en el cliente, pero lo es aun más que contamos con todos los recursos del servidor donde están alojadas las páginas. Estos recursos, como podrían ser gestores de bases de datos, servidores de correo o el propio sistema de archivos del servidor, son los que nos van a permitir construir todo tipo de aplicaciones.

Como ventajas adicionales se puede destacar que el código de las páginas con los Scripts nunca llega al cliente, recordamos que al navegador sólo le llega HTML, y esto implica que nuestros visitantes nunca van a poder acceder al corazón de las aplicaciones que hayamos desarrollado, es decir, a los Scripts del lado del servidor.

Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

La forma en que se programó el sistema fue con una visión orientada a objetos, por lo que se tuvo que recurrir a un esquema en la programación de PHP para que esto fuera posible.

Para poder manipular objetos y clases se tiene que partir de dos estructuras, las funciones y las clases. La sintaxis para poder crear una función es la siguiente:

```
function nombre-función( parametro1, parametro2, ..., parametroN) {  
    declaraciones de variables;  
  
    sentencia1;  
    senetencia2;  
  
    ...  
    return valor;  
  
}
```

La sintaxis que se sigue para la creación de una clase es la que se muestra a continuación:

```
class nombre-clase {  
    var $atributo1;  
    var $atributo2;  
    ...  
    var $atributoN;  
  
    function nombre-clase(parametr1, parametro2, ..., parametroN) {  
        estructura de la función;  
    }  
    ...  
    function funciónN (... ) {  
        ...  
    }  
  
}
```

La siguiente figura muestra la forma en como se programa con un modelo orientado a objetos y la manera de implementarla en una página Web.

```
<?php
class cuadro_texto{
    var tamaño;
    var nombre;
    var tipo;
    var etiqueta;

    function cuadro_texto($tamano, $nombre, $tipo, $etiqueta) {
        $this->tamano=$tamano;
        $this->nombre=$nombre;
        $this->tipo=$tipo;
        $this->etiqueta=$etiqueta;
    }

    function mostrar_cuadro(){
        echo $this->etiqueta;
        echo "<INPUT TYPE=\ "$this->tipo\ " NAME=\ "$this->nombre\ " SIZE=\ "$this->tamano\ " >";
    }
}
?>
```

Este cuadro muestra la creación de una clase llamada `cuadro_texto`.

```
<?php
echo "<FORM ACTION=\ "siguiente.php\ " METED=\ "post\ ">";
$caja1=new cuadro_texto(12,"nombre", "text", "Nombre: ");
$caja1->mostrar_cuadro();
$caja2=new cuadro_texto(4,"edad", "text", "Edad: ");
$caja2->mostrar_cuadro();
echo "</FORM>";
?>
```

Una llamada a la clase `cuadro_texto` en una página Web estaría representada como se muestra en el cuadro anterior.

Esta es la forma como se implementaron las clases para PHP al momento de trabajar en este sistema.

4. 2- Implementación

La implementación del sistema consta de varias pantallas, dependiendo de la acción que se desea realizar, así como de los atributos que posee el usuario, serán las pantallas que se muestran. A continuación se muestran algunas de ellas y parte del código de programación en el que se apoya.

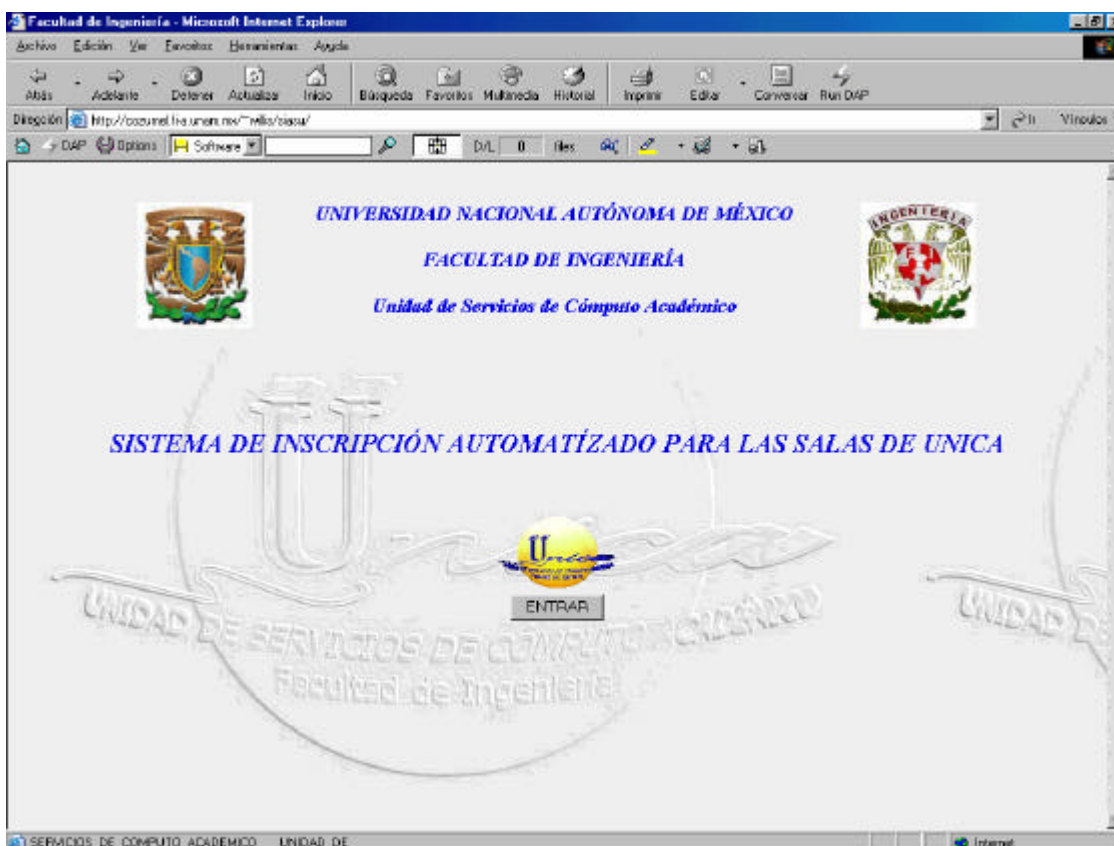


Fig. 4.1 Salida de la página de bienvenida o de presentación de SIASU en php.

La imagen de la figura 4.1, muestra la salida en el Navegador de la página “index” (nombre del archivo inicial de una cuenta de servicio Web), la cuál se da la bienvenida al sistema.

En la tabla 4.1 se muestra el código PHP que compone el documento “index” del sistema.

```

<?php
//Este archivo es la pagina inicial de esta aplicaci3n. Continua con el archivo inicio.php
include ('./m3dulos/M3dulo_funciones.php'); //En estas l3neas estoy incluyendo el c3digo de trabajo de archivos
include './clases/controles_html.php'; //alternos pero necesarios para el buen funcionamiento de esta pagina.
include './clases/tabla_html.php'; //De manera que este archivo no sea muy grande, y pueda ser comprensible.

html_inicio1("Facultad de Ingenier&iacute;a",4, "/imagenes/bg_lock.gif");

echo "<BR>\n<CENTER>";
$u= new formulario_html("", "http://www.unam.mx", "post");
$f= new formulario_html("", "http://www.fi-a.unam.mx", "post");

$t1= new tabla_html("TABLA1");
$t1->abrir_tabla_html(0,0,0,"80%","100","");
$t1->elemento_tabla("TR","CENTER","MIDDLE","","","","");
$t1->elemento_tabla("TD","","","","","","");
$u->abrir_formulario();
$u->mostrar_control_html("image","imagen","/imagenes/UNAM.jpg",105,115,"",0,"");
$u->cerrar_formulario();
$t1->elemento_tabla("TD","CENTER","","","","","2");
echo "<I>\n<B>\n<FONT face=\"Times New Roman, Times, serif\" size=\"4\" color=\"#0000DD\">";
echo "UNIVERSIDAD NACIONAL AUT&Oacute;NOMA DE M&Eacute;XICO";
echo "<BR>\n<BR>FACULTAD DE INGENIER&Iacute;A";
echo "<BR>\n<BR>Unidad de Servicios de C&oacute;mputo Acad&eacute;mico";
echo "</FONT>\n</B>\n</I>";
$t1->elemento_tabla("TD","","","","","","");
$f->abrir_formulario();
$f->mostrar_control_html("image","imagen","/imagenes/logo-fi.jpg",105,115,"",0,"");
$f->cerrar_formulario();
$t1->cerrar_tabla_html();

echo "<BR>\n<BR>";
echo "<I><B>\n<FONT face=\"Times New Roman, Times, serif\" size=\"5\" color=\"#0000DD\">";
echo "<BR>SISTEMA DE INSCRIPCI&Oacute;N AUTOMAT&Iacute;ZADO PARA LAS SALAS DE UNICA";
echo "</FONT>\n</B></I>";
echo "</CENTER>\n<BR>\n<BR>";
echo "<P align=\"center\">";

$i= new formulario_html("", "/inicio.php", "post");
$i->abrir_formulario();
echo "<center>";
$i->mostrar_control_html("image","unica","/imagenes/Unica3Dclr_sinfondo.gif",120,80,"",0,"");
echo "<BR>";
$i->mostrar_control_html("submit","unica","ENTRAR","","","","0","");
$i->cerrar_formulario();

//echo "<BR><B>\n<FONT color=\"#000099\" size=\"5\">SIASU</FONT>\n</B>";
echo "</center>\n<BR>\n<BR>";

html_fin();
?>

```

Tabla 4.1

El inicio de sesión está dado por lo que se podría llamar la pantalla inicial al sistema. Aquí es donde se valida si el usuario es, o no, usuario del sistema y los privilegios que este posee. La figura 4.2 muestra la página de inicio de sesión a SIASU.

En esta pantalla se define también el tipo de operación que realizará el usuario, en este caso, un alta de alumno o simplemente una consulta.

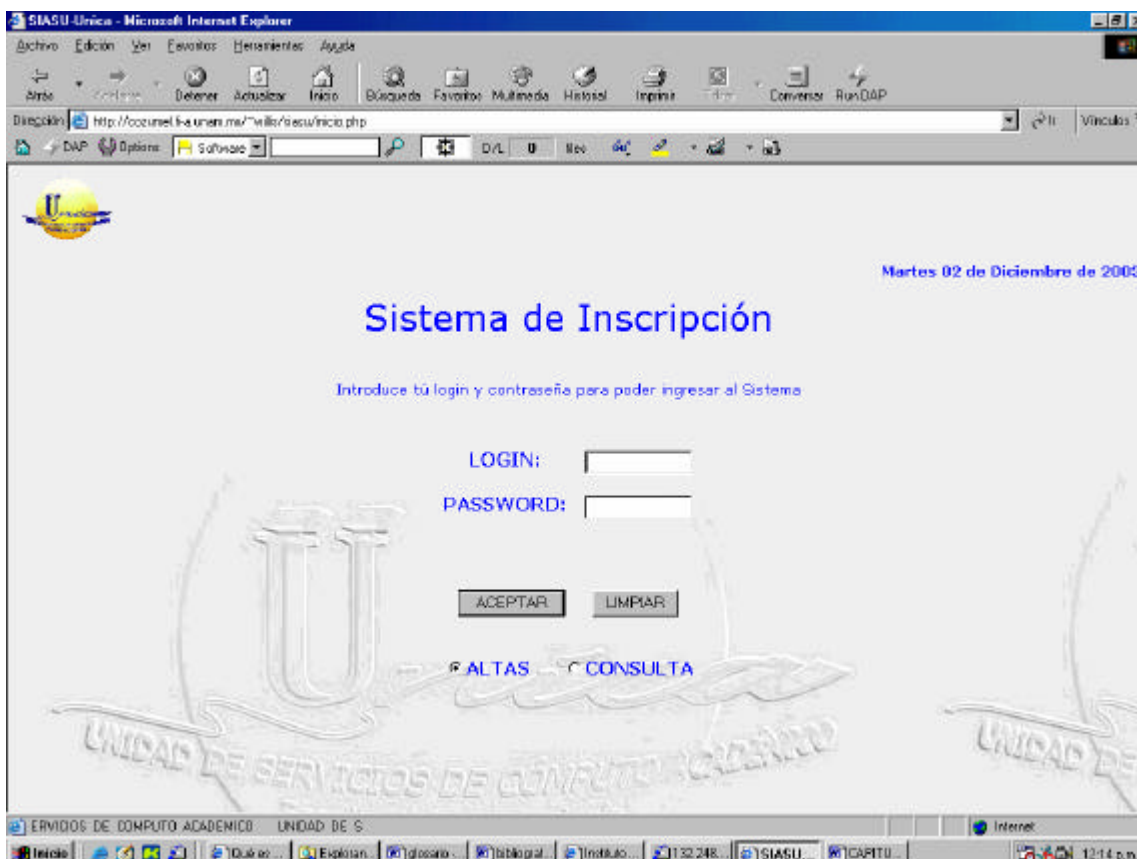


Fig. 4.2 Página de inicio de SIASU.

La tabla 4.2 muestra el código PHP utilizado para la presentación de la pantalla de inicio.

```

<?
/*Viene del archivo inicio.php y toma como entrada el login y la contraseña de las personas
con autorizacion para el manejo de este sistema. Carga el archivo validaEntrada.php
*/

session_start();
require "sesion_registrada.php";
include './módulos/Módulo_funciones.php';
include './clases/controles_html.php';
include './clases/tabla_html.php';

html_inicio1("Bienvenido",6,"./imagenes/Unica3D_gris3.jpg");

$form= new formulario_html("", "validaEntrada.php", "post");
$form->abrir_formulario();
print("<CENTER> Sistema de Inscripti&oacute;n <BR>\n<BR>");
print("<FONT SIZE=1>
Introduce t&uacute; login y contrase&ntilde;a para poder ingresar al Sistema</FONT>");
print("<BR>\n<BR>");

$t1= new tabla_html("TABLA1");
$t1->abrir_tabla_html(0,8,0,"", "", "");
$t1->elemento_tabla("TR", "CENTER", "MIDDLE", "", "", "");
$t1->elemento_tabla("TD", "", "", "", "", "");
print("<B>LOGIN: </B>");
$t1->elemento_tabla("TD", "", "", "", "", "");
$form->mostrar_control_html("text", "login", "", 12,10, "", 0, "");
$t1->elemento_tabla("TR", "CENTER", "MIDDLE", "", "", "");
$t1->elemento_tabla("TD", "", "", "", "", "");
print("<B>PASSWORD: </B>");
$t1->elemento_tabla("TD", "", "", "", "", "");
$form->mostrar_control_html("password", "contrasena", "", 12,16, "", 0, "");
$t1->cerrar_tabla_html();
print("<BR>");

$t2= new tabla_html("TABLA2");
$t2->abrir_tabla_html(0,8,8,"", "", "");
$t2->elemento_tabla("TR", "", "MIDDLE", "", "", "");
$t2->elemento_tabla("TD", "", "", "", "", "");
$form->mostrar_control_html("submit", "enviar", "ACEPTAR", "", "", "", "");
$t2->elemento_tabla("TD", "", "", "", "", "");
$form->mostrar_control_html("reset", "limpiar", "LIMPIAR", "", "", "", "");
$t2->cerrar_tabla_html();

$t3= new tabla_html("TABLA3");
$t3->abrir_tabla_html(0,10,10,"", "", "");
$t3->elemento_tabla("TR", "", "MIDDLE", "", "", "");
$t3->elemento_tabla("TD", "", "", "", "", "");
$lista[0][0]=A;
$lista[0][1]="<B>ALTAS</B></TD><TD>";
$lista[1][0]=C;
$lista[1][1]="<B>CONSULTA</B>";
$form->mostrar_control_html("radio", "respuesta", $lista, 25, "", "si", "0", "");
$t3->cerrar_tabla_html();
$form->cerrar_formulario();
print("</CENTER>");

html_fin();
?>

```

Tabla 4.2 entrada.php

La tabla 4.3 muestra el formulario que hay que llenar en caso de que el usuario no se encuentre registrado en el archivo proporcionado por USECAD. Este archivo no se mostrara si no es a través de otro archivo llamado altaAlumno.php, el cuál se muestra en la figura 4.4.

```

<?php
//Este archivo se incluye en el archivo altaAlumno.inc.php
session_start();
require "sesion_registrada.php";

echo "<CENTER>";
echo "<FONT size=4>";
echo "<B>FAVOR DE INTRODUCIR LOS DATOS</B><BR>";
echo "No. Cta: ".$usuario[clave];
echo "</FONT>";
echo "<br><font face='arial, helvetica' size='2'>Las cajas marcadas con</font>";
echo "<font color='red'> * </font>";
echo "<font face='arial, helvetica' size='2'> es forzoso que se llenen.</font>";
    $f= new formulario_html("", "altaAlumno.php", "post");
    $f->abrir_formulario();
    $f->mostrar_control_html("hidden", "usuario[clave]", "$usuario[clave]", "", "", "0, """);
$t= new tabla_html("TABLA1");
$t->abrir_tabla_html(0,3,3,"65%", "25%", "");
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Apellido paterno </B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[paterno]", "$usuario[paterno]", 20,20, "", 1, "");
    if ((solo_letras($usuario[paterno]) != $usuario[paterno]) AND isset($envio)){
        echo "<font face='arial, helvetica' size='2'>.solo_letras($usuario[paterno]).</font>";
        $avanzar=0;
    }
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Apellido materno </B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[materno]", "$usuario[materno]", 20,20, "", 1, "");
    if ((solo_letras($usuario[materno]) != $usuario[materno]) AND isset($envio) AND $usuario[materno] != ""){
        echo "<font face='arial, helvetica' size='2'>.solo_letras($usuario[materno]).</font>";
        $avanzar=0;
    }
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Nombre </B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[nombre]", "$usuario[nombre]", 20,25, "", 1, "");
    if ((solo_letras($usuario[nombre]) != $usuario[nombre]) AND isset($envio)){
        echo "<font face='arial, helvetica' size='2'>.solo_letras($usuario[nombre]).</font>";
        $avanzar=0;
    }
}
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Fecha de Nacimiento:</B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[dia]", "$usuario[dia]", 2,2, "", "", "");
    echo "/";
    $f->mostrar_control_html("text", "usuario[mes]", "$usuario[mes]", 2,2, "", "", "");
    echo "/";
    $f->mostrar_control_html("text", "usuario[anno]", "$usuario[anno]", 4,4, "", 1, "");
    if ((valida_fecha($usuario[dia], $usuario[mes], $usuario[anno]) == "Esa fecha no existe") AND isset($envio)){
        echo "<font face='arial, helvetica' size='2'>Verificar fecha</font>";
        $avanzar=0;
    }
}

```

Tabla 4.3 formulario.inc.php

```

$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
$t->elemento_tabla("TD", "left", "bottom", "", "", "", "");
    echo "&nbsp;<font face='arial, helvetica' size='2'\>D&iacute;a&nbsp;&nbsp;&nbsp;Mes&nbsp;&nbsp;&nbsp;A&ntilde;o</font>";
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Sexo:</B> ";
$t->elemento_tabla("TD", "left", "bottom", "", "", "", "");
    echo "&nbsp;&nbsp;&nbsp;";
        $lista[0][0]=M;
        $lista[0][1]="M";
        $lista[1][0]=F;
        $lista[1][1]="F";
        $f->mostrar_control_html("select", "usuario[sexo]", $lista, 2, "", "si", "1", "");
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Direcci&oacute;n</B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[direccion]", "Usuario[direccion]", 20, 140, "", 1, "");
    if ((noes_vacia($usuario[direccion]) != $usuario[direccion]) AND isset($envio)){
        echo "<font face='arial, helvetica' size='2'\>.noes_vacia(Usuario[direccion]).</font>";
        $avanzar=0;
    }
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Num. telefonico</B>: ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[telefono]", "Usuario[telefono]", 14, 8, "", 0, "");
    if (( solo_numeros($usuario[telefono], 8) != $usuario[telefono] AND isset($envio)){
        echo "<font face='arial, helvetica' size='2'\>.solo_numeros(Usuario[telefono], 8).</font>";
    }
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    for($i=-1; $i<10; $i++) {
        $lista_s[$i][0]= $i + 1;
        $lista_s[$i][1]= $i + 1;
    }
    echo "<B>Semestre Actual:</B>";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("select", "usuario[semestre]", $lista_s, 2, "", "si", "1,");
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
    echo "<B>Promedio</B>: ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("text", "usuario[promedio]", "Usuario[promedio]", 4, 4, "", 1, "");
    if (( solo_reales($usuario[promedio]) != $usuario[promedio] AND isset($envio) AND $usuario[promedio] != ""){
        echo "<font face='arial, helvetica' size='2'\>.solo_reales(Usuario[promedio]).</font>";
    }
}
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
$t->elemento_tabla("TR", "", "TOP", "", "", "", "");
$t->elemento_tabla("TD", "right", "bottom", "", "", "", "");
        $listado[1][0]=ci107;
        $listado[1][1]="107 Civil";
        $listado[2][0]=mm108;
        $listado[2][1]="108 Minas";
    echo "<B>Carrera:</B> ";
$t->elemento_tabla("TD", "", "bottom", "", "", "", "");
    $f->mostrar_control_html("select", "usuario[carrera]", $listado, 1, "", "", "1", "");

```

Tabla 4.3 formulario.inc.php (continuación)

```

$t->elemento_tabla("TR","CENTER","TOP","","","");
$t->elemento_tabla("TD","","bottom","","",2);
    $lista_db[0][0]= 0;
    $lista_db[0][1]= "No requiero Base de Datos<TD>";
    $lista_db[1][0]= 1;
    $lista_db[1][1]= "Requiero Base de Datos";
    echo "<B>Desea Base de Datos: </B>";
$t->elemento_tabla("TR","CENTER","TOP","","","");
$t->elemento_tabla("TD","","bottom","","",2);
    $f->mostrar_control_html("radio","usuario[deseabd]",$lista_db,3,"","si",0,"");
$t->cerrar_tabla_html();
echo "<BR>";

$t1= new tabla_html("TABLA1");
$t1->abrir_tabla_html(0,3,3,"40%","","");
$t1->elemento_tabla("TR","","TOP","","","");
$t1->elemento_tabla("TD","center","","","");
    $f->mostrar_control_html("hidden","avanzar","$avanzar","","",0,"");
    $f->mostrar_control_html("hidden","enviar","enviar","","",0,"");
    $f->mostrar_control_html("submit","enviar","Aceptar","","",0,"");
    // el boton submit tenia como etiqueta "Enviar" cambiada por "Aceptar"
    $f->cerrar_formulario();
$t1->elemento_tabla("TD","center","","","");
    $f1=new formulario_html("","numcuenta.php","post");
    $f1->abrir_formulario();
    $f1->mostrar_control_html("hidden","clave","","",0,"");
    $f1->mostrar_control_html("hidden","listo","Continuar","","",0,"");
    $f1->mostrar_control_html("submit","regresar","Cancelar","","",0,"");
    $f1->cerrar_formulario();
$t1->cerrar_tabla_html();
echo "</center>"; ?>

```

Tabla 4.3 formulario.inc.php (continuación)

La figura 4.3 muestra la salida del archivo altaAlumno.php En caso de ser afirmativa la petición pasará al llenado del formulario. El código para esta página se muestra en la tabla 4.4.

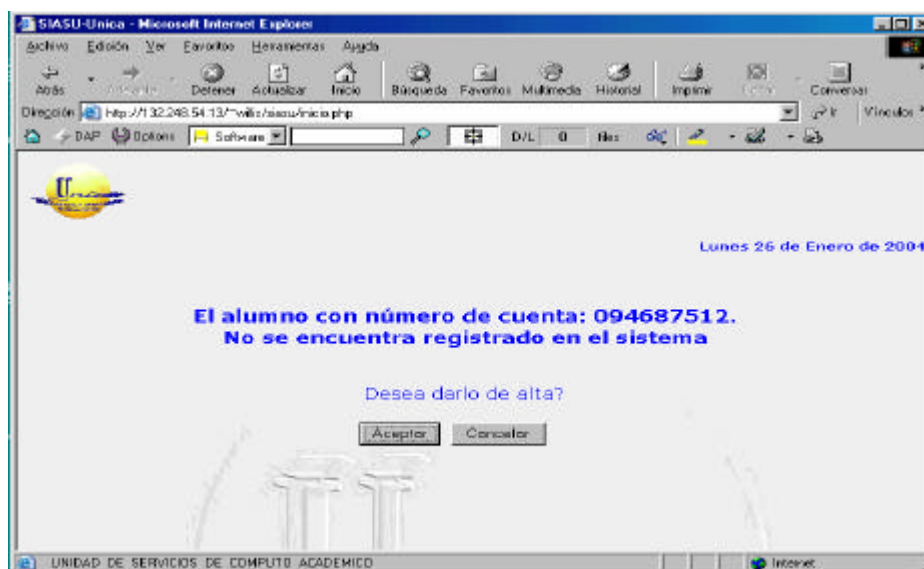


Fig. 4.3 altaAlumno.php

```
<?
/*Este archivo solicita los datos de aquellos usuarios que no se encuentran registrados.
Este archivo se anexaba al archivo verificarbd.php con la funcion require
*/

session_start();
require "sesion_registrada.php";

if($avanzar==0 AND isset($envio))
{
    require "../módulos/Módulo_funciones.php";
    require "../clases/controles_html.php";
    require "../clases/tabla_html.php";
}

html_inicio("Bienvenido",4,"./imagenes/Unica3D_gris3.jpg");

if ( ( solo_letras($usuario[paterno]) == $usuario[paterno]) AND ((solo_letras($usuario[materno]) == $usuario[materno]) OR
$usuario[materno] == NULL) AND (solo_letras($usuario[nombre]) == $usuario[nombre]) AND
(valida_fecha($usuario[dia],$usuario[mes],$usuario[anno]) != "Esa fecha no existe") AND (noes_vacia($usuario[direccion]) ==
$usuario[direccion]) AND (solo_numeros($usuario[telefono],8) == $usuario[telefono]) AND (solo_reales($usuario[promedio]) ==
$usuario[promedio]) )
{
    include('./insertarAlumno.php');
}
else
{
    include('./formulario.inc.php');
}
html_fin();
?>
```

Tabla 4.4 altaAlumno.php

Una vez que el usuario decide dar de alta al alumno en turno, este es el formulario que se muestra para la captura de sus datos, figura. 4.4.

SIASU Unica - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección: http://132.248.54.13/~wills/siasu/inscricao.php

Lunes 26 de Enero de 2004

Las cajas marcadas con * es forzoso que se llenen.

Apellido paterno : * No debe quedar vacío

Apellido materno : *

Nombre : * No debe quedar vacío

Fecha de Nacimiento: / / * Verificar fecha
Día / Mes / Año

Sexo : M * *

Dirección : * No debe quedar vacío

Num. telefonico :

Semestre Actual: 1 *

Promedio : *

Carrera: 122 Tesisla *

Desea Base de Datos:

No requiero Base de Datos Requiero Base de Datos

Aceptar Cancelar

UNIDAD DE SERVICIOS DE COMPUTO ACADÉMICO

Fig. 4.4 archivo altaAlumno.php con el archivo formulario.inc.php insertado.

4.3 Pruebas

Las pruebas constituyen una parte vital del ciclo de vida en el desarrollo de cualquier sistema. Estas se llevan a cabo con el propósito de descubrir defectos, tanto en el código como en la estructura propia del sistema. Además se establecen para mejorar la calidad del sistema.

Las pruebas son un elemento crítico para la garantía de calidad de nuestro sistema. La realización de las pruebas concretará, en cierta forma, los objetivos que fueron planteados para este sistema al inicio de la tesis.

No importando el tipo de sistema de ingeniería desarrollado, estos se pueden probar siguiendo dos aspectos:

El primero es enfocándonos en el código del mismo; es decir, en su lógica y en su estructura interna. Se necesita conocer la función específica para la cual fue creado cada módulo o función, e identificar que cada una de éstas es completamente funcional y operativa.

El segundo se enfoca más en el producto terminado. Son pruebas funcionales basadas en la operación del producto final y que nos aseguran que la operación interna se ajusta a las especificaciones dadas por el cliente.

A las pruebas que llevan a cabo cada uno de estos dos análisis se les conoce como Pruebas de Caja Blanca y Pruebas de Caja Negra, respectivamente.

4.3.1 Pruebas de caja blanca.

4.3.1.1 Pruebas de unidad y de integración.

La prueba de unidad centra el proceso de verificación en la menor unidad del sistema, es decir, en el módulo. Los puntos que comúnmente se tocan para esta prueba son:

- Interfaz. Para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que esta siendo probada.
- Estructuras de datos. Para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo.
- Condiciones límite. Para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento.
- Caminos independientes. Para asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez.

También deben diseñarse pruebas para detectar errores en las estructuras de datos locales de cada módulo, esto es las siguientes categorías:

- Tipificación impropia o inconsistente.
- Iniciación o valores implícitos erróneos.
- Nombre de variables incorrectos.
- Tipos de datos inconsistentes.
- Desbordamiento de direccionamiento.

Además deben establecerse caminos para detectar errores debidos a cálculos o comparaciones incorrectos o flujos de control inadecuados:

- Comparaciones entre tipos de datos distintos.
- Operadores lógicos o de precedencia incorrectos.
- Variables o comparadores incorrectos.
- Terminación inapropiada o inexistente de ciclos.

Por lo que se refiere a las pruebas de integración, estas se establecen para construir la estructura del programa mientras que, simultáneamente se realizan pruebas para detectar errores asociados con la interacción. Para unir los módulos se debe seguir la técnica denominada integración incremental, que consiste en probar pequeños segmentos para que los errores sean más fáciles de aislar y de corregir.

4.3.2 Pruebas de caja negra.

4.3.2.1 Pruebas de Volumen.

La prueba de volumen intenta detectar errores en el software debidos a condiciones anormales, demandando recursos en cantidades, frecuencias o volúmenes fuera del rango que se especifico en el diseño. Algunos ejemplos de pruebas de volumen serían:

- Incrementar las frecuencias de datos de entrada.
- Incrementar el número de equipos que se conectaran y realizarán movimientos en el sistema.
- Casos en los que se requiere el máximo de memoria o de otros recursos.
- Incremento del número de usuarios que hacen uso del sistema.

4.3.2.2 Pruebas Funcionales o de validación.

La validación del software se consigue al demostrar la conformidad con los requisitos, es decir, que el software funciona de acuerdo con las expectativas dadas por el cliente.

Cuando se ha realizado el software para un cliente, es necesario realizar la prueba de validación, para que el cliente establezca conformidad del sistema con todos los requisitos dados por este.

Para realizar esta prueba es necesaria la instalación del producto terminado, posteriormente se lleva a cabo la capacitación de los usuarios. Se entenderá por producto terminado al sistema listo para ser empleado en el equipo para el que fue diseñado, junto con el manual de usuario. El sistema deberá estar listo para comenzar su operación de manera inmediata.

Administración y Mantenimiento

Capítulo

5

5. 1- Mantenimiento del Sistema

El mantenimiento es uno de los puntos determinantes de los ciclos de vida. No hay sistema que este exento de sufrir errores o al paso del tiempo tal vez no cumpla con las necesidades para las que fue diseñado, por continuos cambios, que pudieran surgir en su entorno. El mantenimiento puede ser tan simple como modificar código para que el sistema trabaje adecuadamente, o tan drástico como la modificación de errores de especificación o para amoldarse a nuevos requerimientos.

Como se especifico en los ciclos de vida, es difícil encontrar sistemas que no necesiten mantenimiento. En toda la vida útil de un sistema, sus requerimientos originales serán modificados para reflejar las necesidades cambiantes de la empresa.

El mantenimiento del software recae en tres categorías importantes:

- Mantenimiento perfectivo
- Mantenimiento adaptativo
- Mantenimiento correctivo

El mantenimiento perfectivo son aquellos cambios que mejoran el sistema de alguna manera sin cambiar su funcionalidad.

El mantenimiento adaptativo es el mantenimiento que se requiere debido a los cambios que van surgiendo en el ámbito donde fue desarrollado el software y por tanto este debe seguir cumpliendo con su funcionalidad.

El mantenimiento correctivo modifica de alguna manera el funcionamiento de ciertas partes del software, en lo que podría ser una reestructuración de este.

El proceso de mantenimiento se inicia por un conjunto de solicitudes de cambio por parte de los usuarios del sistema o por parte de los administradores de este. La petición generalmente involucra elementos de mantenimiento adaptativo, correctivo y perfectivo.

Terminando este proceso se libera una nueva versión del sistema. El proceso de esta manera se vuelve iterativo con un nuevo conjunto de cambios propuestos para la nueva liberación.

Los nuevos requerimientos deben ser validos, los componentes del sistema deben ser rediseñados e implementados y parte, o la totalidad, del sistema debe ser probada antes de su liberación para estar seguros de que no se produzcan fallas una vez puesto en marcha.

SIASU, como versión final de un proyecto creado y diseñado por el grupo de administradores de UNICA con base en una necesidad que se venía presentando de tiempo atrás, cumple con las características de ser accesible en su mantenimiento. Puesto que es un sistema modular, es posible modificar solo aquella parte, que por ciertas razones, necesitara de algún tipo de mantenimiento. En caso de que, dada la situación, los requerimientos cambiaran a futuro o se necesitara que el sistema mostrará algunas otras estadísticas, los programadores o administradores podrán darle un mantenimiento adaptativo o correctivo al agregarle módulos extra. Al llevar a cabo esta acción el personal a cargo deberá ubicar las variables globales que son necesarias pasar en sus parámetros y que variables entregaran para que de esta forma no se corrompa el código o se pierda la secuencia de este.

En lo que se refiere a su administración SIASU cuenta con tres grupos con privilegios distintos. Uno de los grupos, el grupo de administración, estará encargado de la dirección del sistema. Este grupo podrá realizar las acciones que realizan los otros dos grupos restantes con la diferencia de que este poseerá los privilegios para saber que personas están haciendo uso de SIASU en lo que se refiere a la consulta de datos de alumnos. Podrá observar las estadísticas que arroja el sistema en lo que se refiere al número de alumnos por semestre y al número de alumnos por carrera que se dan de alta en las salas de UNICA.

Como módulo extra, el grupo de administradores podrá crear y eliminar cuentas de usuarios. Anteriormente, este proceso se llevaba a cabo directamente en el servidor y propiamente en la base de datos, por lo que este modulo será de gran ayuda ya que con el simple hecho de estar dentro del sistema de consulta, el administrador conseguirá dar de alta una cuenta en caso de ser necesario.

5. 2- Administración de la información

La información que genera, una vez que el alumno se da de alta, tiene dos funciones. Primero, ciertos datos son utilizados por el sistema para generar las cuentas personales de los alumnos y de esta forma trabajen en los distintos equipos de computo con los servicios institucionales que ofrece UNICA. La segunda función es llevar estadísticas de los alumnos.

Aun que el grupo de administradores de UNICA son las personas que gestionan el sistema de inscripción así como las cuentas de usuarios, la administración de la información recae en el Departamento de Servicios Académicos de UNICA.

Este Departamento lleva estadísticas de cursos, instructores, alumnos, número de alumnos acreditados y no acreditados, y más.

Las estadísticas que arroje SIASU servirán como apoyo al saber el número de alumnos por carrera que hacen uso de los servicios de UNICA, así como del semestre que cursan actualmente.

5. 3- Perfiles de protección

Cuando se diseño y se puso en marcha la primera versión del sistema de inscripción a las salas de UNICA, el equipo creador de este sistema (el equipo fue el área de administradores del DROS de UNICA) pusieron en práctica sus conocimientos de programación y una gran inquietud para obtener lo que sería el antecesor del SISASU. Al momento de estructurar el sistema de inscripción se tomó en cuenta ya no sólo la funcionalidad sino también se maduró en que debería ser un sistema seguro.

SIASU cuenta con varios perfiles que lo hacen seguro.

- Como primer punto, SIASU cuenta con un sistema de entrada por validación, esta validación está dada por una base de datos. Solo el personal que cuenta con un nombre de usuario y la contraseña correspondiente a ese login podrá acceder al sistema.
- Cuenta con tres grupos de usuarios con diferentes privilegios, de modo que un usuario con privilegios limitados no podrá dar de alta a un alumno o mucho menos ver estadísticas o verificar registros.
- La única forma de entrar es a través de su ventana de inicio o de petición de contraseñas. Si un usuario quisiera entrar desde otra dirección electrónica, que por alguna razón conociera, el sistema le indicará que no es posible entrar salvo que lo haga por la dirección que se le indique.
- Una vez dentro del sistema, el usuario está obligado a moverse exclusivamente con los botones propios del sistema de inscripción. Si intentara regresar o ir a otra página con los controles propios del navegador, la sesión de trabajo terminaría automáticamente.
- En caso de que la sesión se deje de utilizar por un periodo de 20 minutos, el sistema finalizará la sesión de trabajo, de este modo se asegura que personas externas a UNICA no haga uso de éste.
- Cada entrada de texto está validada por código, de modo que si alguna persona que no forme parte de UNICA trata de colocar código malicioso para que el sistema le devuelva datos de usuarios o le conceda el acceso, el sistema quedara fuera de peligro.

Estos puntos hacen que SIASU intente ser un sistema seguro.

La otra parte de la seguridad está dada por el software que se utiliza. Como se habló en el segundo capítulo, se cuenta con el sistema operativo de Linux y con el servidor Web Apache lo que hace que, una vez configurado adecuadamente, sea muy seguro. Con este software es posible restringir la entrada al sistema a ciertos equipos exclusivamente, de modo que gente externa a la Unidad de Cómputo, aun cuando sepa la dirección electrónica de SIASU, no le permita el acceso a éste.

Por otro lado, postgresQL es un manejador de base de datos muy seguro y combinado con el software descrito líneas arriba hacen de la información algo fiable.

Resultados y costos

Capítulo

6

6. 1- Resultados

Los resultados arrojados por las pruebas que se llevaron a cabo en el sistema de inscripción (SIASU) fueron los siguientes:

6.1.1 Resultados de las pruebas de caja blanca.

6.1.1.1 Pruebas de unidad y de integración.

Para poder acreditar estas pruebas, se analizó la lógica de cada uno de los archivos. Se probaron todos los ciclos e interacciones de control del código, siguiendo todas las rutas posibles.

Después, tomando los objetos probados en unidad, se verificó que los datos, resultantes de la salida de algún objeto, fueran pasados adecuadamente, como entrada, al objeto asociado. La prueba de unidad centra el proceso de verificación en los módulos de manera independiente.

El resultado de esta prueba fue alentador, no existe en el código, de cada uno de los archivos, rutas que no sean posibles de recorrer ni sentencias que no se emplearán. Así también, al integrar cada uno de los objetos no se detectó errores asociados con dicha integración, cumpliendo, a su vez, con lo establecido en el diseño.

6.1.2 Resultados de las pruebas de caja negra.

6.1.2.1 Pruebas de Volumen.

Este sistema fue diseñado y desarrollado, según requerimientos, para ser operado, únicamente, en dos o tres ubicaciones distintas. Al realizar la prueba de volumen, este requerimiento fue sobrepasado en más de un 400%, solicitando a un grupo de 15 personas que hicieran uso del sistema conjuntamente a una hora determinada.

El resultado de esta prueba fue exitoso. El sistema trabajo sin algún problema, y la información de la base de datos permaneció integra una vez finalizada la prueba.

6.1.2.2 Pruebas Funcionales.

Se probó la funcionalidad del sistema con base en los requerimientos dados por los miembros de UNICA. Se realizaron varias pruebas insertando usuarios y eliminándolos, así como insertando alumnos prueba en la base de datos, para posteriormente verificar su estado en la misma.

El sistema trabajó sin ningún problema, no se generaron errores de algún tipo. Ni en el archivo de salida ni en la base de datos.

6.1.2.3 Pruebas Estructurales.

La realización de esta prueba, apoyada con la de volumen, nos mostró que el sistema no solo trabaja bien con grandes cargas de trabajo, también mostró su buen desempeño en el tiempo. Es decir que no bajo su rendimiento en tiempo al ingresar datos o al consultar y actualizar la base de datos, pese al gran volumen de información que se trabajo.

6.1.2.4 Pruebas de aceptación del sistema.

Los usuarios mostraron un descontento con la interfaz primera y algunas limitantes que daba el propio software, en la operación del sistema.

Se opto por utilizar conjuntamente otro tipo de software como apoyo al software PHP. También se modifico la interfaz haciéndola más sencilla y explicita para que el usuario no tuviera problemas con el manejo de información o datos que se solicitan.

Nota: La información que se muestran, como resultados de cada una de las pruebas, fue la pesquisa final, dada una vez que el sistema llega a su fin.

Como resultado tangible del buen desempeño de SIASU, se muestra el comprobante de acceso a las Salas y de inscripción a los servicios institucionales que ofrece UNICA. Las imágenes Fig. 6.1 y Fig. 6.2 muestran los datos que el comprobante requiere dependiendo los requerimientos de los alumnos.



Fig. 6.1 Comprobante para utilizar las Salas de UNICA



Fig. 6.2 Comprobante con servicio extra de Base de Datos

El primero muestra un comprobante exclusivo para el uso de los equipos de las Salas de computo y del servicio de correo electrónico; el segundo, en cambio, a demás de ofrecer los mismos servicios, indica que el usuario solicito trabajar con un manejador de base de datos.

Entre los resultados que se obtuvieron se encuentran también las estadísticas de alumnos. Las figuras siguientes Fig. 6.3 y Fig. 6.4 muestran la grafica comparativa del número de alumnos tanto por semestre como por carrera. Este tipo de resultados fue uno de los requerimientos solicitados para este sistema.

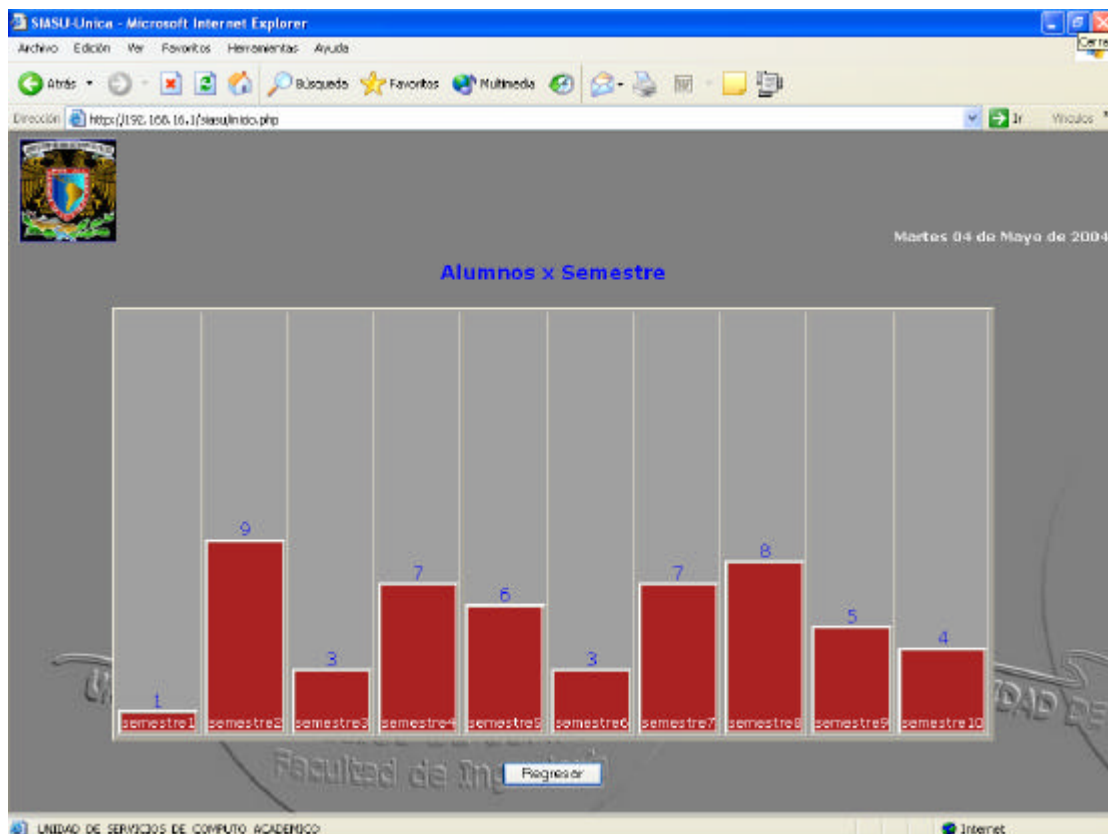


Fig. 6.3 Estadísticas de Alumnos X Semestre

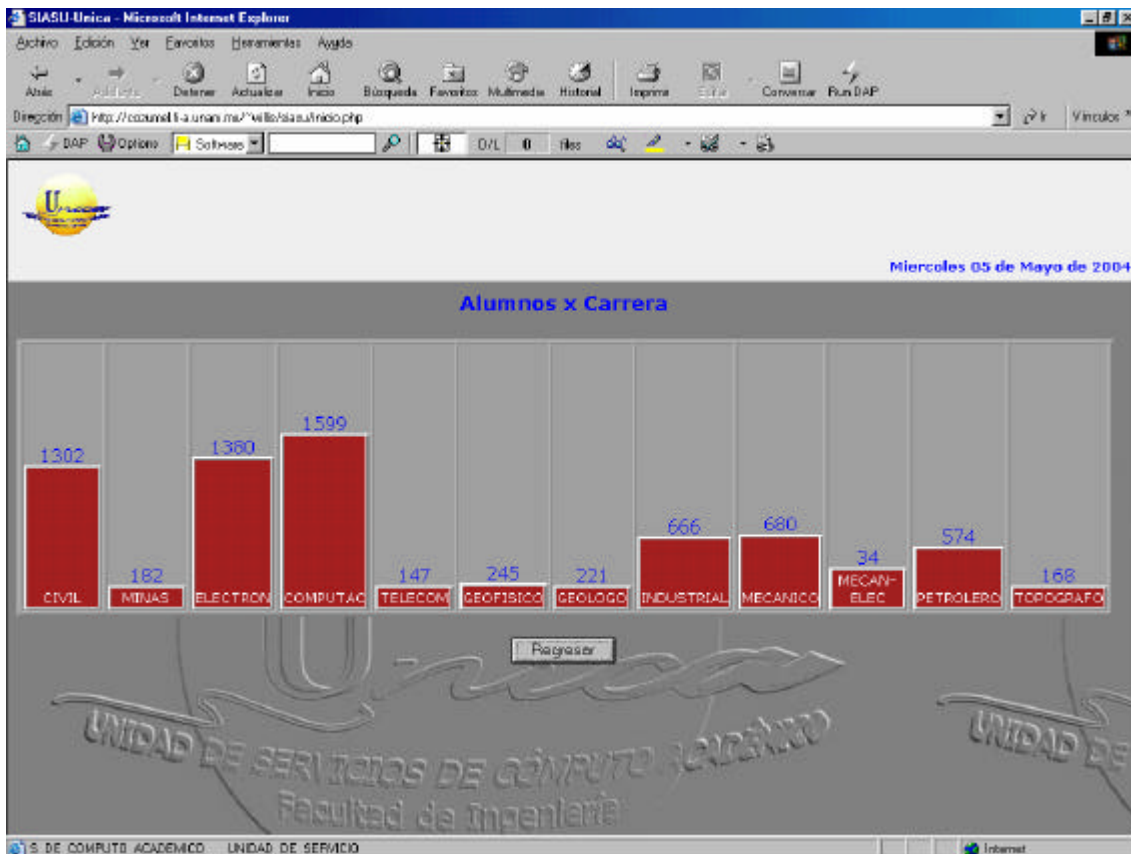


Fig. 6.4 Estadísticas de Alumnos X Carrera

6.2- Costos

Un punto muy importante en la parte de resultados es el referente a los costos del sistema.

Se llevó a cabo el análisis sobre la necesidad de este sistema y si era o no factible su realización. Una vez que se tomó la decisión de que SIASU vendría a traer mejores resultados a la forma de manejar las cuentas de alumnos y una considerable rapidez en la prestación de los servicios, se pasó al punto de diseñar el proyecto. La elaboración de esta tesis constituye este punto en particular.

Para poder cimentar el proyecto, se solicitó un equipo con características de servidor para ser instalado y configurado de modo que se pudiera sacar el mejor provecho a las características propias del software que se trabajó. Fueron solicitados los tres equipos

PC. los cuales van a ser utilizados como prestadores de servicio. Esto debido a que se tomaran los datos respectivos en cuanto a la MAC de la tarjeta de red y la dirección IP con la que trabajaran.

El software utilizado en este sistema se baso en software con licencias publicas. Ya que es software libre no se necesitó del pago de licencias además de poderlo trabajar y amoldar a nuestros requerimientos.

Se capacitará a las personas encargadas de utilizar el sistema. También se capacitará a una persona que llevará a cabo la administración de este.

Se solicito la entrega de un manual de usuario.

Como honorarios se tomaron en cuenta el tiempo invertido para la realización del proyecto, los cursos de capacitación y la compra de libros.

Estos punto fueron tomados en la siguiente es una tabla para representar el precio que tendría SIASU.

PRESUPUESTO	Tiempo Invertido (hrs.)	Precio / hrs.	TOTAL
Análisis	20	125	2500
Proyecto	20	125	2500
Instalación	10	1000	10000
Capacitación	15	1200 x persona	5000
Documentación	-	-	1000
Equipos Cómputo	-	-	60000
Honorarios	400	-	30000
		TOTAL	111000

Tabla 6.1 Costo del proyecto.

Conclusiones:

Se desarrolló SIASU y se puso en marcha durante este semestre, 2004-2, en forma paralela con la versión anterior, logrando concluir que es de gran utilidad para el alta de usuarios a la Unidad de Servicios de Cómputo Académico, ya que ofrece una respuesta más rápida y un mejor servicio. Esto se debe a que se redujo el tiempo de espera para el préstamo de los servicios institucionales que ofrece UNICA, ya que al momento en que un alumno termina su trámite de alta en la Unidad el sistema automáticamente genera un nombre de usuario único en los servidores, y de esta manera, el alumno pueda utilizar dichos servicios. Así también, el alumno puede hacer uso de las instalaciones de UNICA, ya que otro sistema desarrollado en la Unidad de Servicios de Cómputo Académico se apoya de una de las vistas de SIASU para poder operar, de esta forma, al momento que un alumno tramita su alta en UNICA en ese instante ya puede hacer uso de las instalaciones.

El sistema genera un comprobante para cada uno de los alumnos. El comprobante que se genera, una vez que el alumno se da de alta, posee la información necesaria para que éste pueda hacer uso de los servicios institucionales que ofrece UNICA.

Se cuenta con un historial que indica que persona hace uso del sistema y cuál es la fluencia del mismo. Indicando fecha y hora en la que dicha persona hace uso de éste. El historial solo puede ser consultado por el grupo con privilegios de administrador.

Se desarrolló un módulo que puede generar nuevas cuentas de usuario o eliminar las ya existentes, de esta forma el administrador no necesita conectarse a la base de datos y crear la cuenta para el nuevo usuario. Solo necesita teclear su nombre de usuario, su nombre completo y asignarle una contraseña de entrada. También solicita se indique los privilegios que va a poseer dicho usuario en el sistema.

Tratándose de un sistema para el alta de usuarios, es posible monitorear el número de personas que hacen uso de los servicios institucionales. Por lo que se cuenta con un módulo encargado de representar gráficamente la cantidad de alumnos por semestre y por carreras.

Se puede concluir que los objetivos y necesidades planteadas al inicio de esta tesis se cumplieron satisfactoriamente. Durante el desarrollo de este proyecto se agregaron servicios y aplicaciones al sistema que apoyan la forma de administrar SIASU.

En el transcurso de este proyecto fueron apareciendo varias situaciones de las cuales se aprende y enriquecen a uno. Me di cuenta que un trabajo que atañe intereses y puntos de vista distintos, por las diferentes personas a las que va dirigido, no es nada fácil y hay que lograr que sea lo más explícito posible y que cumpla casi al 100% con estas exigencias. Por tal motivo, algo que aprendí, fuera de lo aprendido en clases, fue que sí se quiere un trabajo que cumpla a detalle con las expectativas del cliente, se debe tener una apropiada comunicación con los involucrados, concibiendo hasta el más mínimo detalle.

Manual de
Usuario

SIASU

ÍNDICE

Presentación	101
Inicio	102
Acceso denegado.....	103
Falta de privilegios	104
Ingresar el número de cuenta.....	105
Elegir si desea Base de Datos.....	106
Usuario dado de alta.....	107
Alumno no encontrado.....	108
Formulario.....	109
Error de entrada.....	110
Fin de sesión	111
Comprobante.....	112
Consulta.....	114
Extras	116

La finalidad de este manual es dar a conocer el manejo del Sistema de Inscripción automatizado.

Este manual se diseñó para usuarios con privilegios para realizar consultas y dar de alta a los alumnos de la Facultad de Ingeniería. No contempla al grupo que puede administrar este sistema.

Manual de usuario

En las líneas siguientes, se realizará un recorrido por el sistema SIASU, con la finalidad de explicar la manera de interactuar con éste.

Presentación.

Cuando se accede al sistema a través del navegador a la dirección que la unidad de cómputo señale (en este caso: <http://ingenieria.unam.mx/~willis/siasu>), se inicia el sistema con una pantalla de presentación. La figura inferior (Fig. 1) muestra la pantalla de presentación, en esta encontramos el logo de **UNICA** así como un botón de entrada. Al presionar cualquiera de estos pasamos a la siguiente ventana.

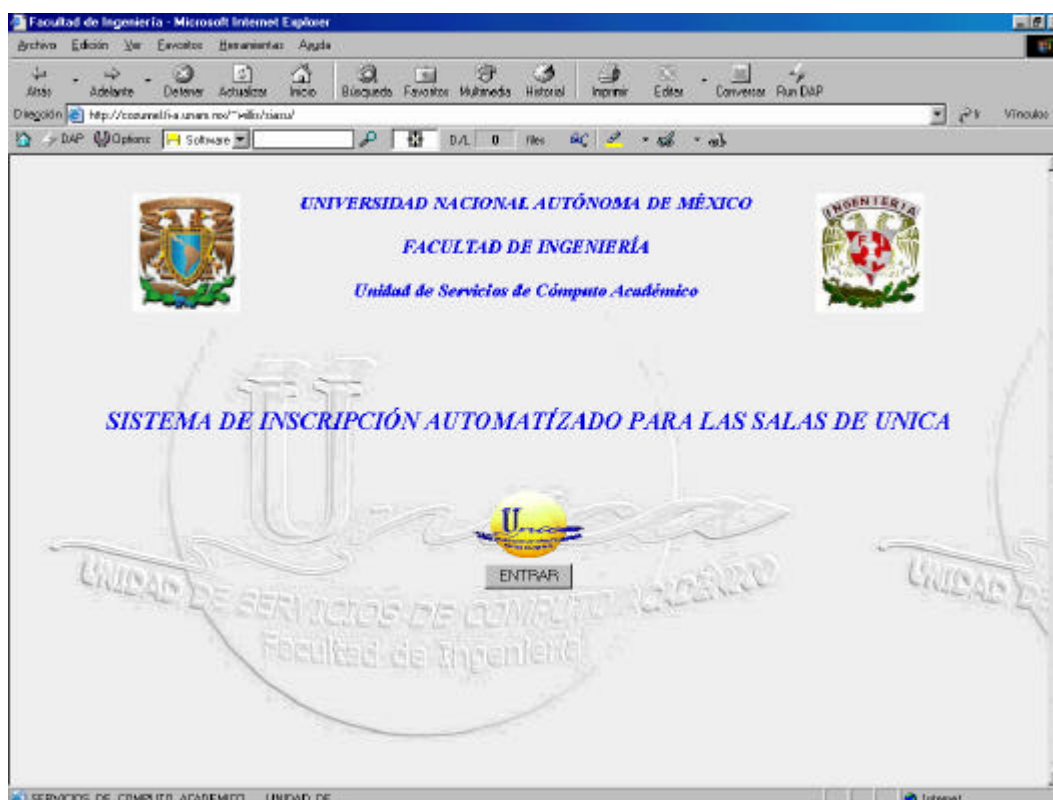


Fig. 1

Inicio.

Esta ventana sirve como entrada al sistema, aquí se nos solicita el nombre de usuario (login) así como la contraseña (password) correspondiente. Se debe presionar el botón **ACEPTAR** para continuar. Esta ventana se muestra en la imagen inferior (Fig. 2).



Fig.2

Acceso Denegado.

Si por alguna razón el nombre de usuario no corresponde con la contraseña tecleada en la ventana de inicio, el sistema publicará la ventana que se muestra en la siguiente imagen (Fig. 3), indicándolo. Para volver a la ventana anterior es necesario presionar la imagen de **Regresar** mostrada.

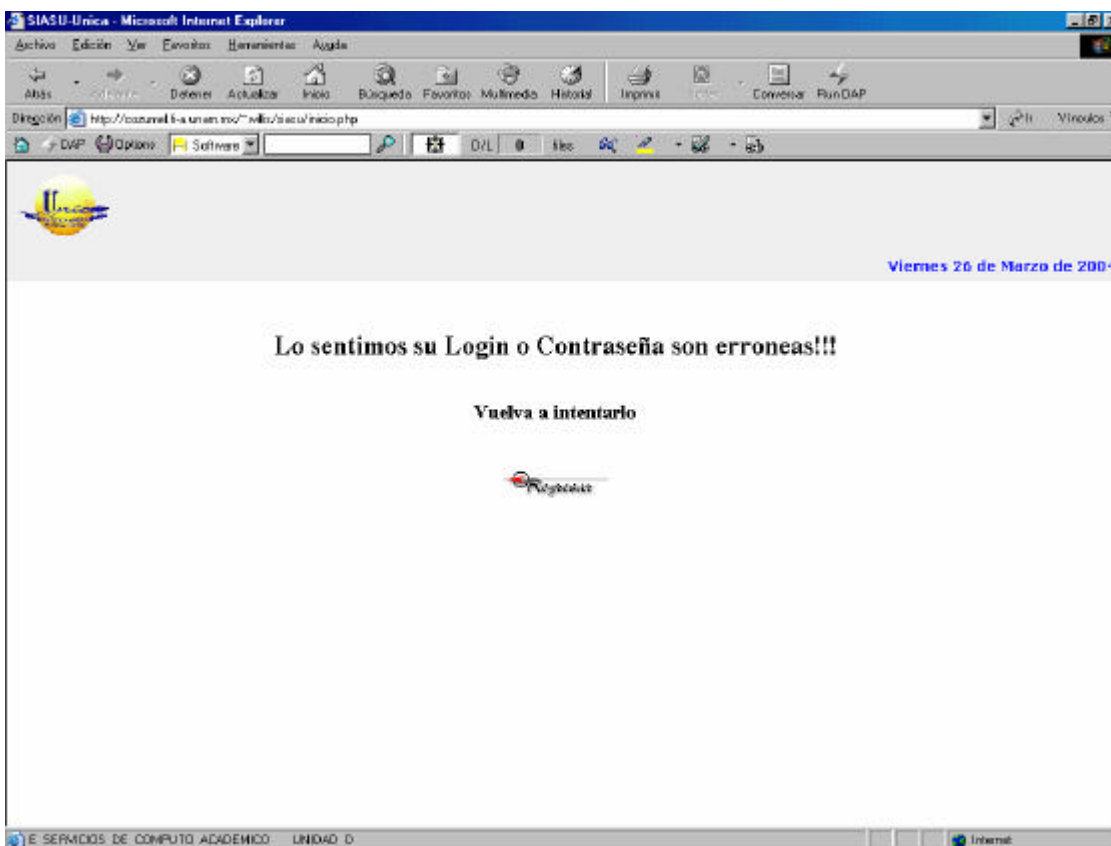


Fig. 4

Falta de Privilegios.

Si el usuario se encuentra en el grupo de consulta, este no podrá dar de alta en el sistema a algún alumno. Por lo que el sistema lo indicará tal y como se muestra en la imagen (Fig. 5). Para volver a la ventana anterior es necesario presionar la imagen de **Regresar** mostrada.

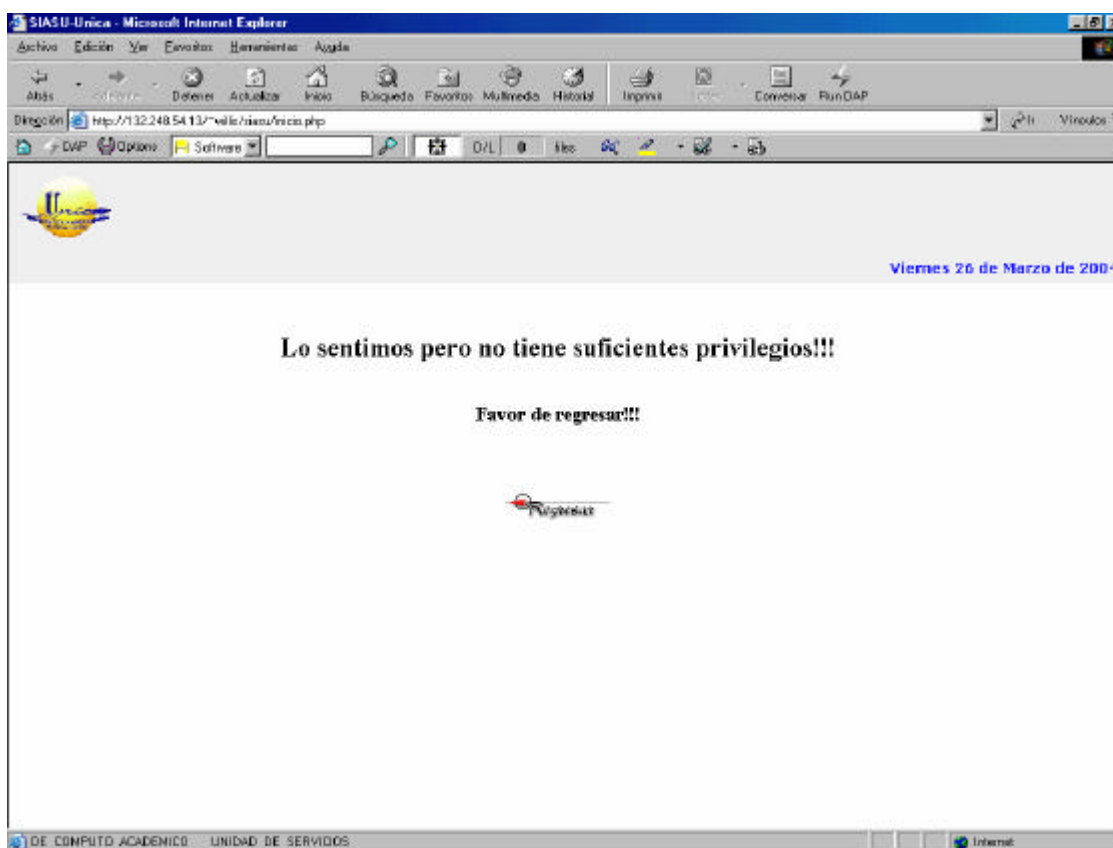


Fig. 5

Ingresar el número de cuenta.

En caso de que los datos dados por el usuario sean correctos y con los privilegios suficientes, y además se eligió la opción de ALTAS, el sistema mostrará la ventana de alta de alumnos. En este caso el usuario del sistema le pide al alumno en turno le proporcione su número de cuenta y una identificación. Este tecleara el número de cuenta en la caja de texto que se muestra en esta pantalla y presionará el botón **Continuar**. La siguiente imagen (Fig. 6) muestra dicha ventana.

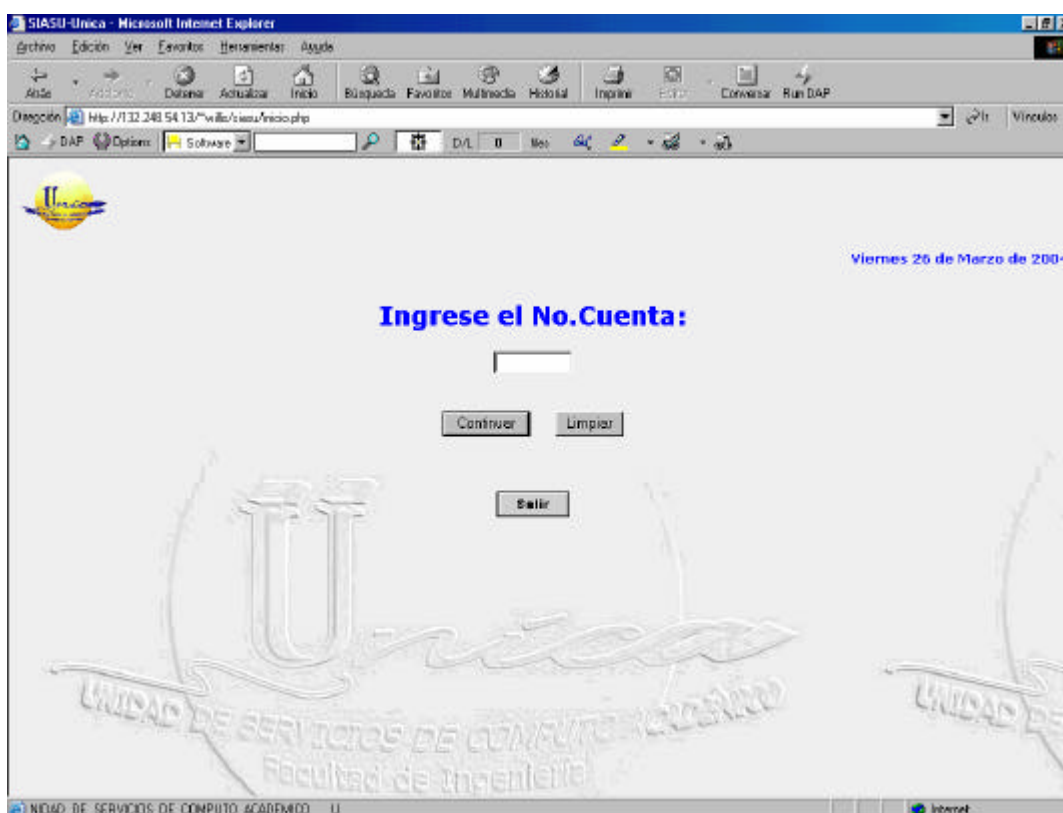


Fig.6

Desea Base de Datos.

Si el alumno estuviera registrado en la base de datos de UNICA, el sistema mostrará la siguiente ventana (Fig. 7), solicitando al usuario le indique al alumno si necesitará trabajar con bases de datos. Dependiendo de la respuesta que dé el alumno se seleccionará y presionará el botón **ACEPTAR**.

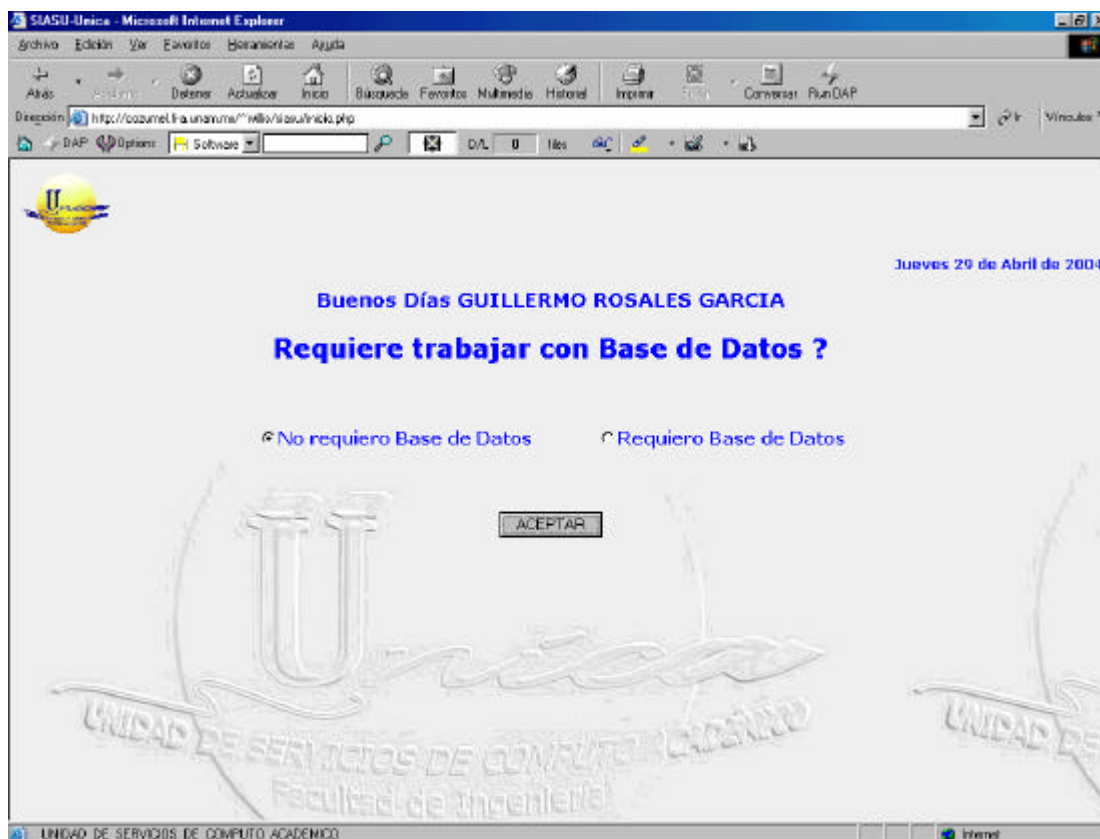


Fig. 7

Usuario dado de Alta.

Si el alumno ya se registro y solicito también trabajar con base de datos, el sistema mostrara la ventana indicada en la figura 8. Queda a consideración del usuario cancelar la operación o imprimir el comprobante del alumno.

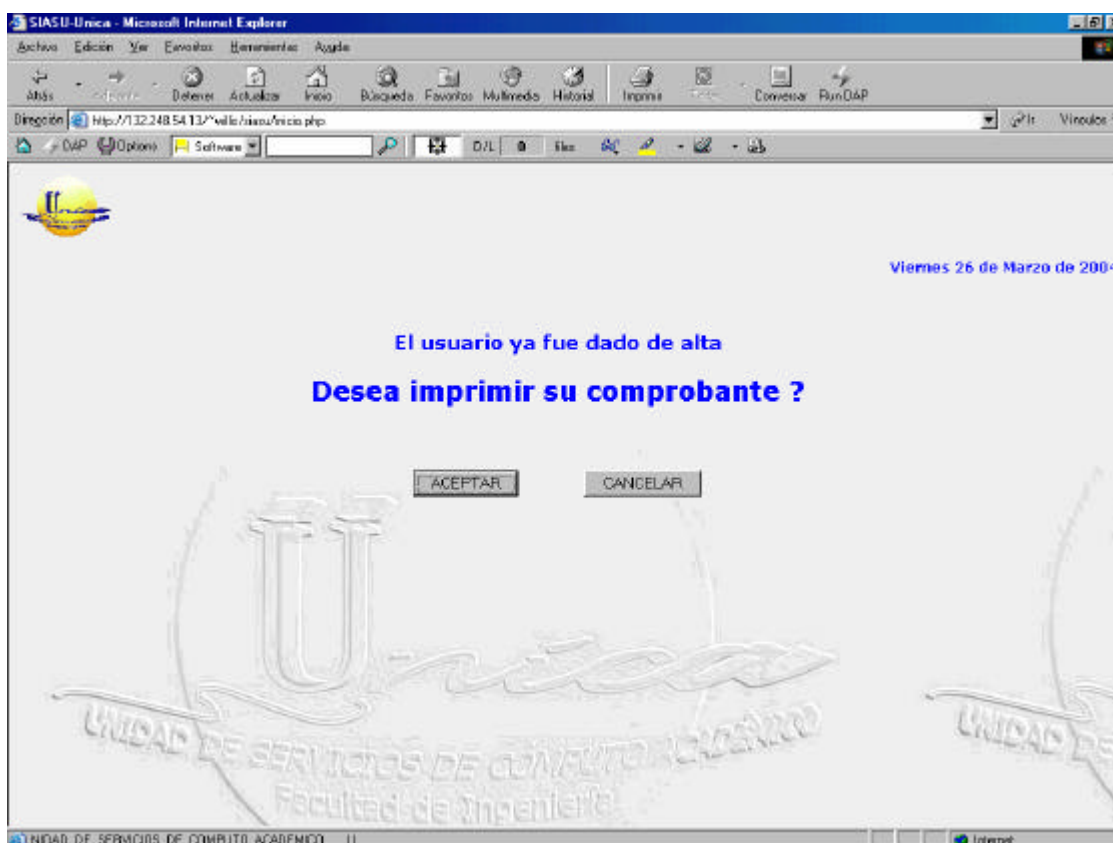


Fig. 8

Alumno no encontrado.

En caso de que el alumno no estuviera registrado en la base de datos de USECAD por algún motivo, dicho alumno tampoco se encontrará en la base de datos de UNICA. La figura (Fig. 9) muestra la ventana que aparecería indicándonos este evento. Presionando el botón **Aceptar** se puede dar de alta como alumno activo.

Nota: Es necesario que el alumno demuestre que se trata de un alumno registrado en la Facultad de Ingeniería, por lo que se le solicitará mostrar alguno de los siguientes documentos: tira de materias del semestre en curso, registro de que esta cumpliendo con el servicio social, o alguno que indique que se encuentra realizando su tesis.

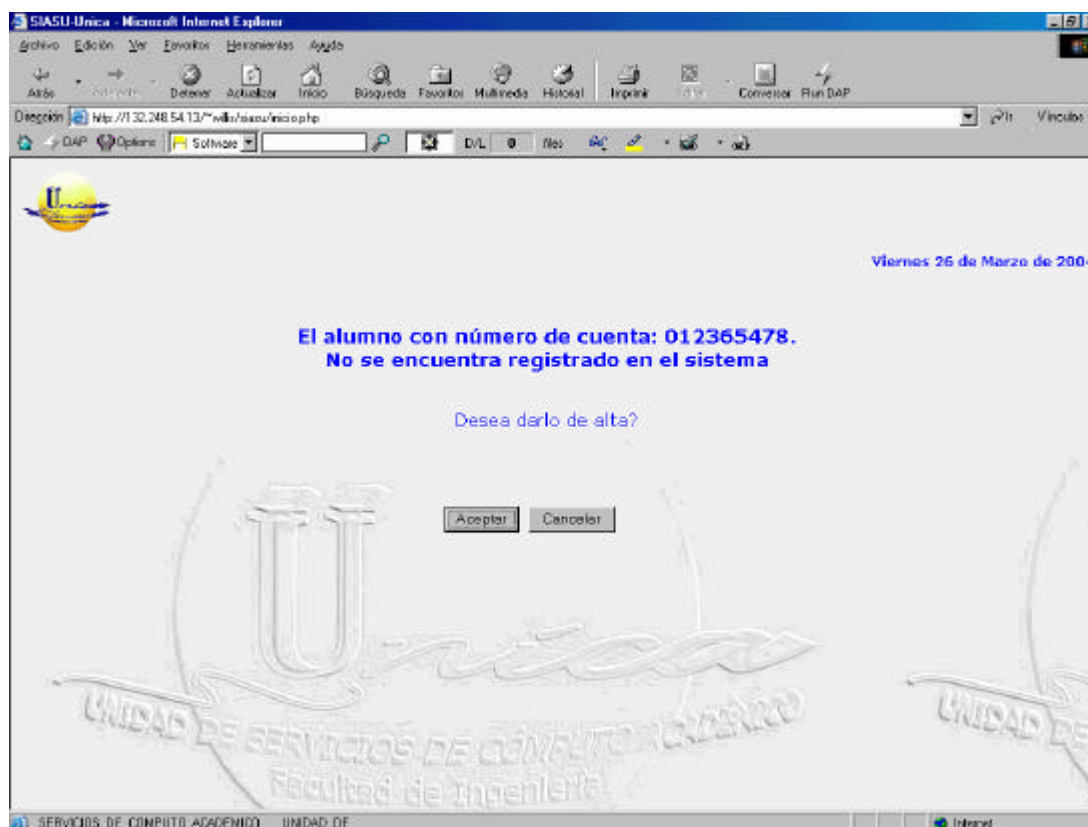


Fig. 9

Formulario.

En esta ventana el usuario colocará los datos del alumno, según se indiquen. Es necesario llenar todas las casillas, o mínimo las que se indican con asterisco color rojo. En caso de que el alumno no proporcione todos los datos o ya no desee continuar, es posible cancelar la operación presionando el botón **Cancelar**. En caso contrario, el botón **Aceptar** validará el correcto llenado del formulario, de manera que si el formulario no es llenado adecuadamente no será posible pasar a la siguiente ventana. Esta pantalla se muestra en la figura inferior (Fig. 10).

SIASU-UNICA - Microsoft Internet Explorer

Viernes 26 de Marzo de 2004

FAVOR DE INTRODUCIR LOS DATOS
No. Cta: 012365478

Las cajas marcadas con * es forzosa que se llenen.

Apellido paterno : *No debe quedar vacío

Apellido materno : *

Nombre : *No debe quedar vacío

Fecha de Nacimiento: *Verificar fecha
Día/ Mes/ Año

Sexo : M *

Dirección : *No debe quedar vacío

Num. telefonico :

Semestre Actual: 1 *

Promedio : *

Carrera: 102 Teórico *

Desea Base de Datos:
 No requiero Base de Datos Requero Base de Datos

UNIDAD DE SERVICIOS ACADÉMICOS UNIDAD DE SERVICIOS ACADÉMICOS

Fig. 10

Error de Entrada.

En caso de que el usuario pretenda entrar al sistema por alguna de las ventanas anteriores, el sistema le pedirá que ingrese de manera correcta por la dirección que ahí se indica. La siguiente figura (Fig. 11) muestra dicha pantalla.

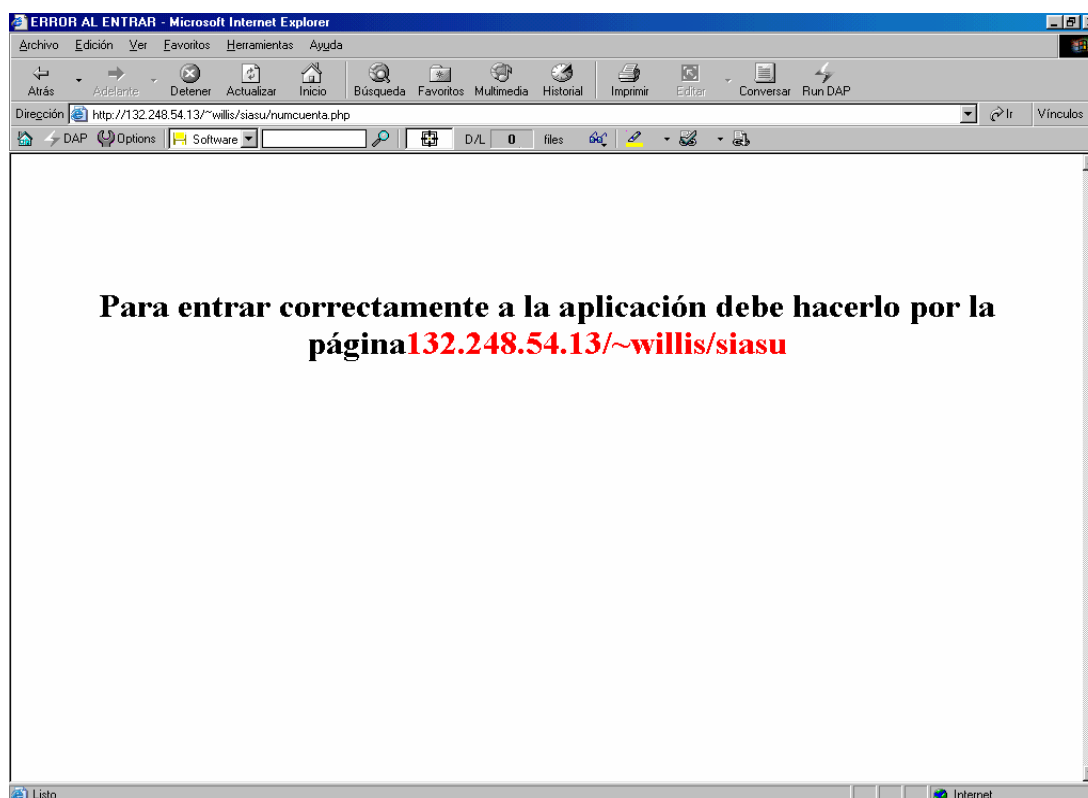


Fig. 11

Fin de Sesión.

En caso de que el usuario decida salir del sistema deberá presionar el botón **Salir** que se muestra en la figura 6. Este finalizará la sesión del usuario en turno y nos llevará a la pantalla de bienvenida (Fig. 1). La imagen inferior (Fig. 12) muestra la ventana de Finalización de la Sesión.

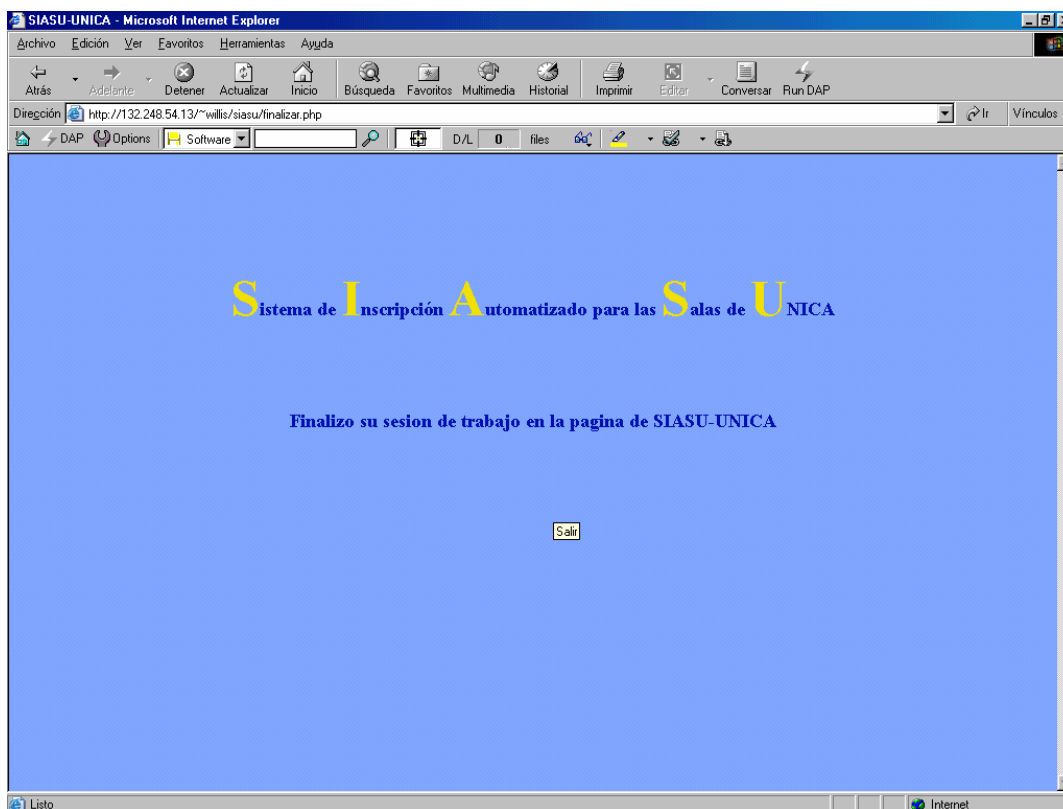


Fig. 12

Comprobante.

El proceso “termina” con la impresión de un comprobante. El usuario deberá presionar el icono que muestra una impresora. Una vez presionado el icono, el sistema nos enviara a la ventana **Número de Cuenta**, y al mismo tiempo mostrará una ventana emergente de impresión en la cuál se deberá teclear el botón Imprimir para poder entregarle el comprobante al alumno. Estas ventanas se muestran en las siguientes figuras (Fig. 13, 14, 15).

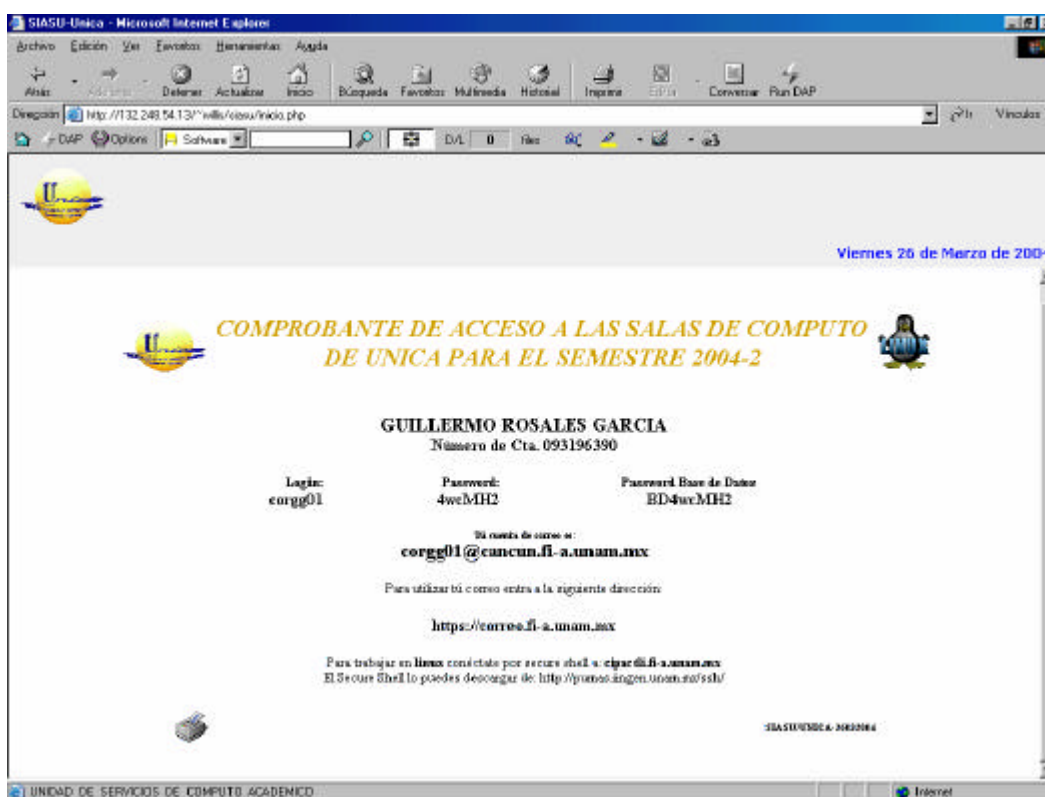


Fig. 13

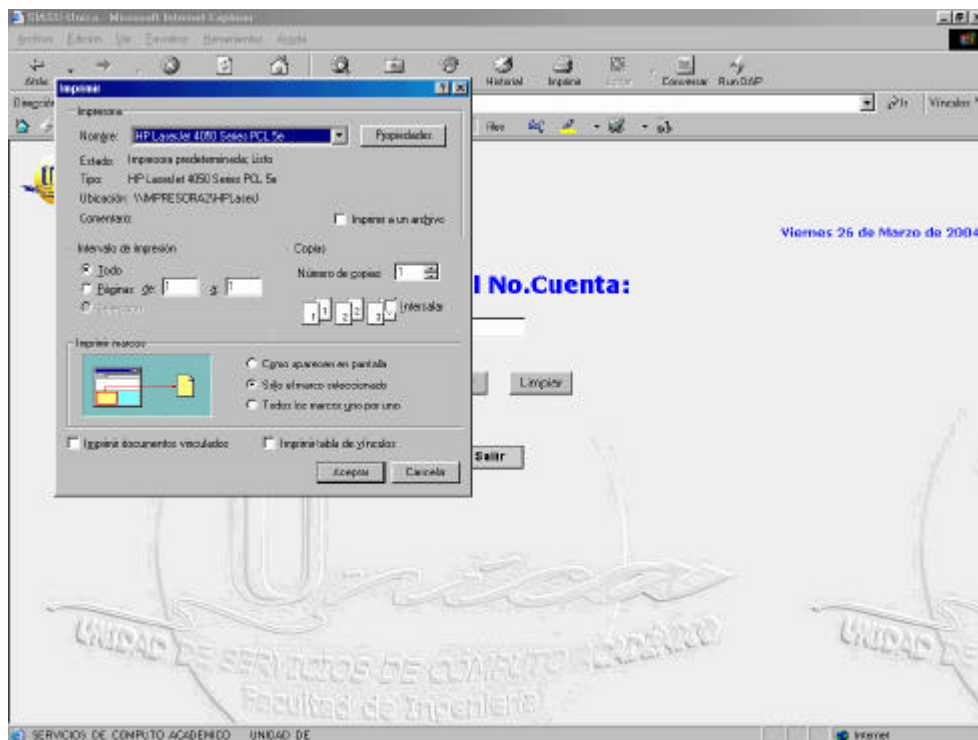


Fig. 14

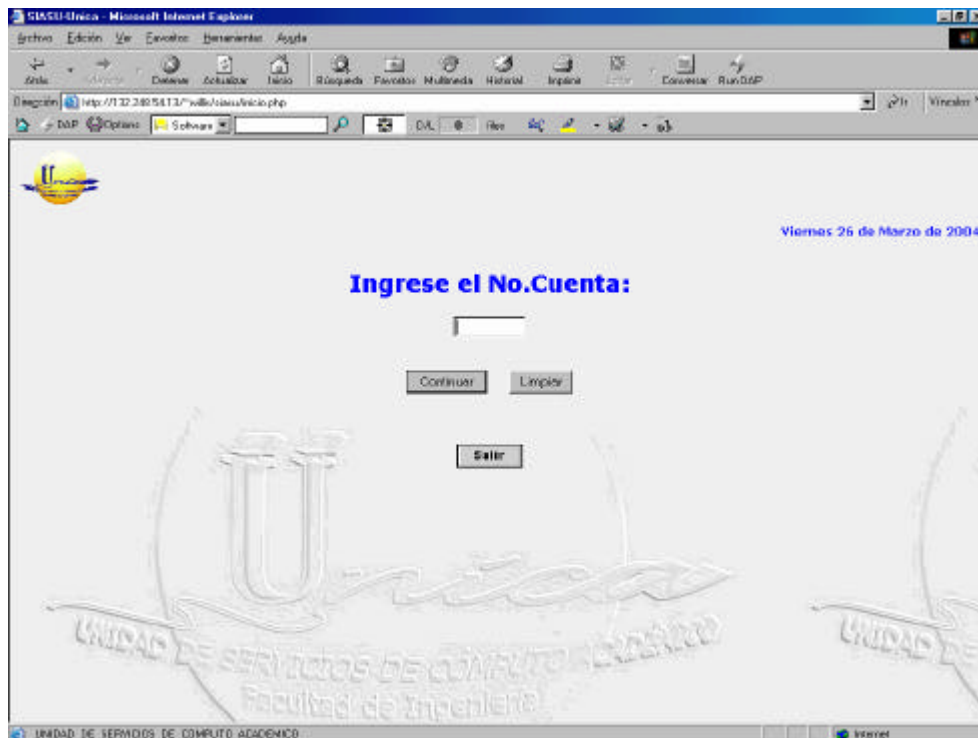


Fig. 15

Consulta.

Si un alumno, ya registrado en el sistema, solicita se le recuerde su clave de usuarios y su contraseña, la persona encargada podrá hacerlo. Primero necesitará salir del sistema y volver a entrar. En la ventana inicio (Fig. 16) habilitará la opción de **CONSULTA** y presionará el botón **ACEPTAR**, ingresando a la ventana mostrada en la figura 17. Posteriormente solicitará los datos que ahí se piden al alumno y presionará el botón **Continuar**. Si los datos son correctos y el usuario, efectivamente, ya fue dado de alta con anterioridad, el sistema mostrará la ventana indicada en la figura 19, en caso contrario se mostrará la ventana de la figura 18 indicando que el alumno aun no tiene comprobante.

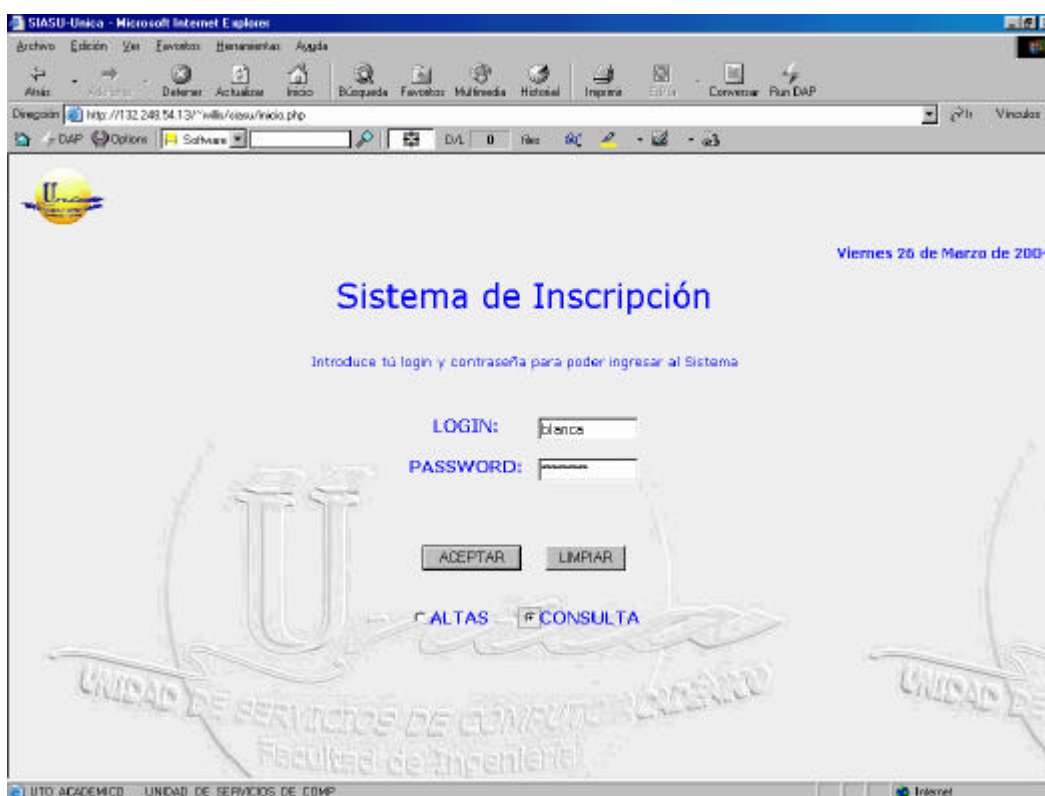


Fig. 16

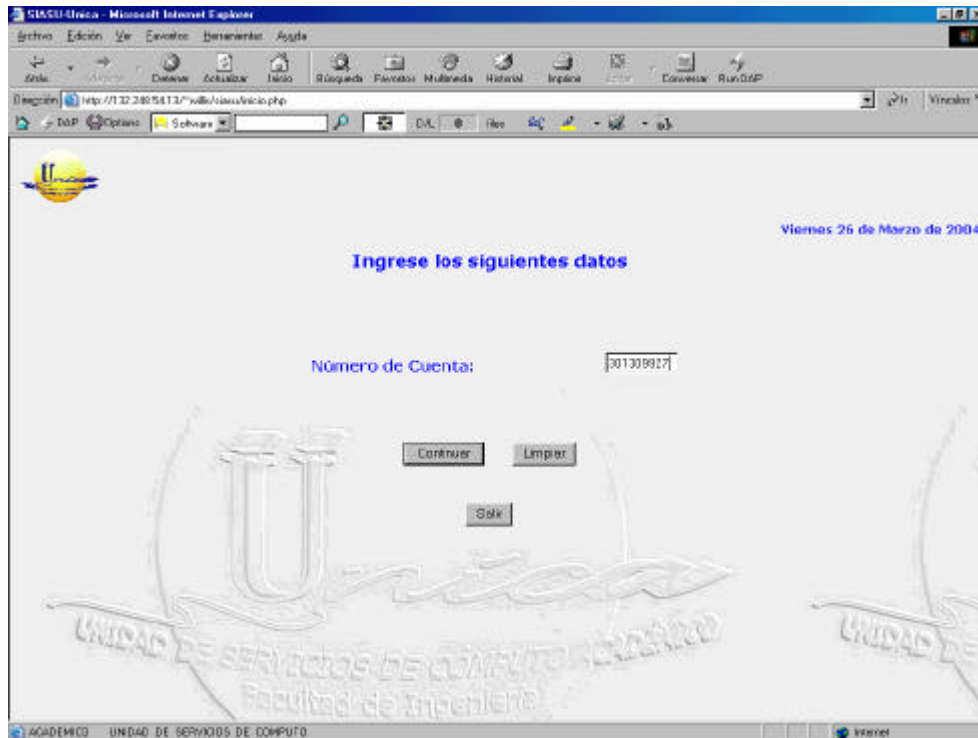


Fig. 17

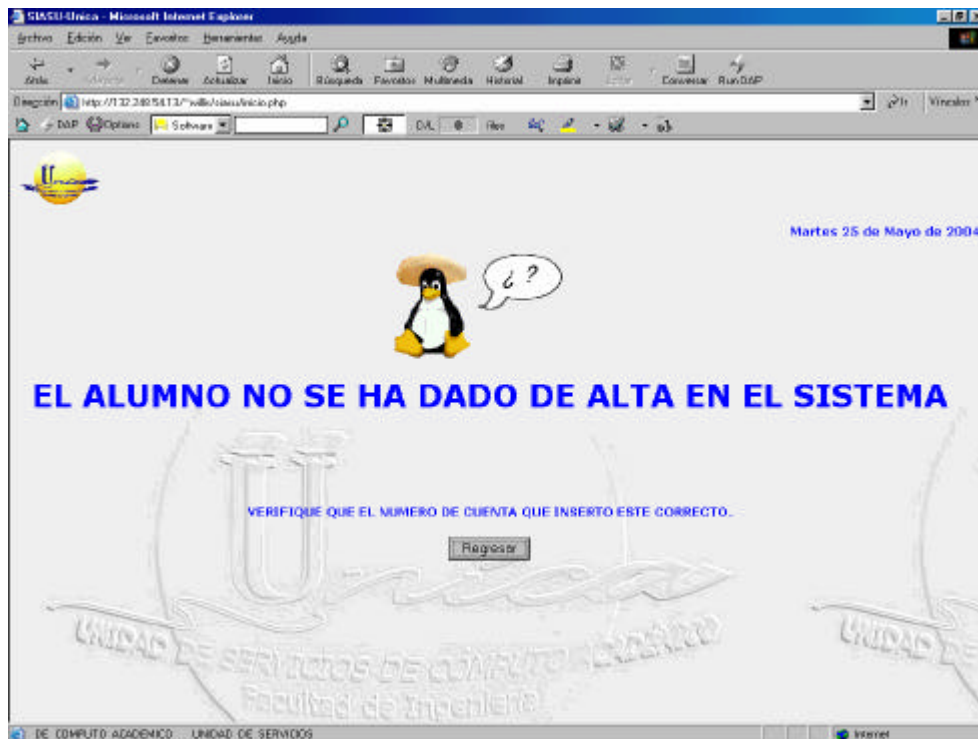


Fig. 18

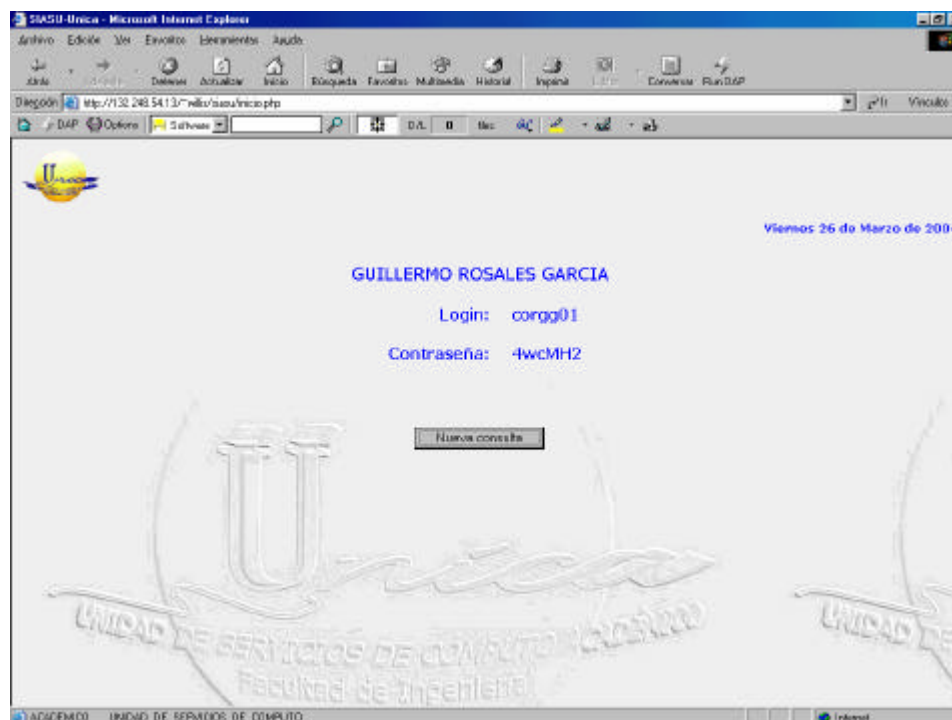


Fig. 19

Deberá de oprimir el botón de nueva consulta para regresar, sin importar si lo que se desea es realizar una nueva consulta o simplemente salir del sistema de consultas. Para salir de esta ventana, será necesario presionar el botón **Salir** que se muestra en la figura 17, enviándonos a la pagina inicial (Fig. 2).

Extras.

Las figuras siguientes (figura 20 y 21) muestran ejemplos de páginas que surgen cuando la actividad en dicha página a sido nula por un tiempo considerable o por intentar darle un “refresco” a la misma; también puede surgir en el caso de que se pretenda regresar a un página anterior sin presionar botones propios de la aplicación, si no por quererlo hacer desde los botones del propio navegador. En cuyo caso se presionara el botón reintentar de la sub-ventana de la figura 21.

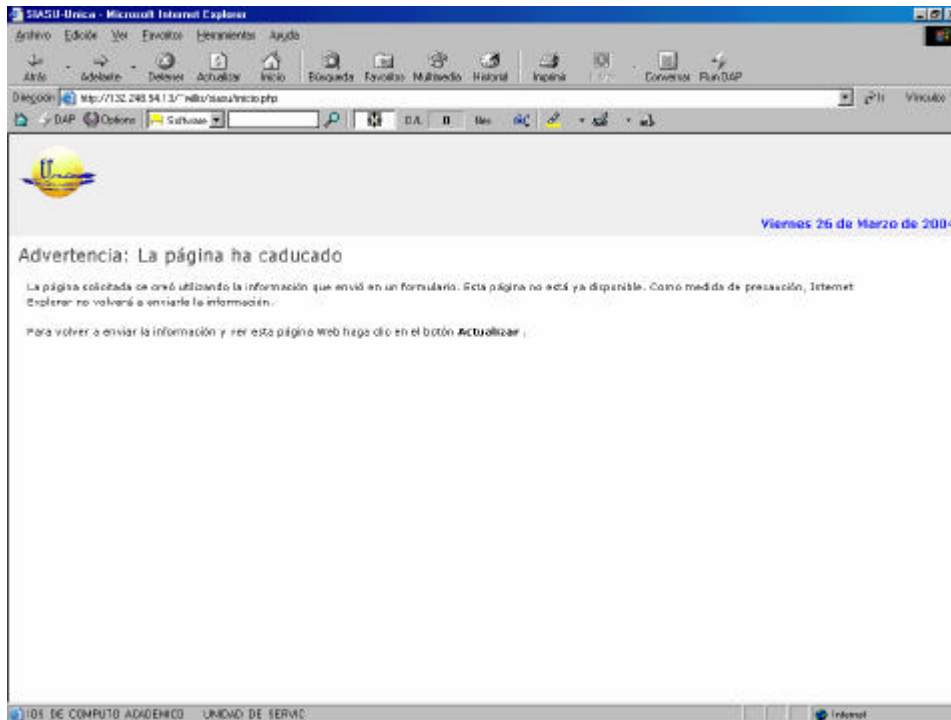


Fig. 20

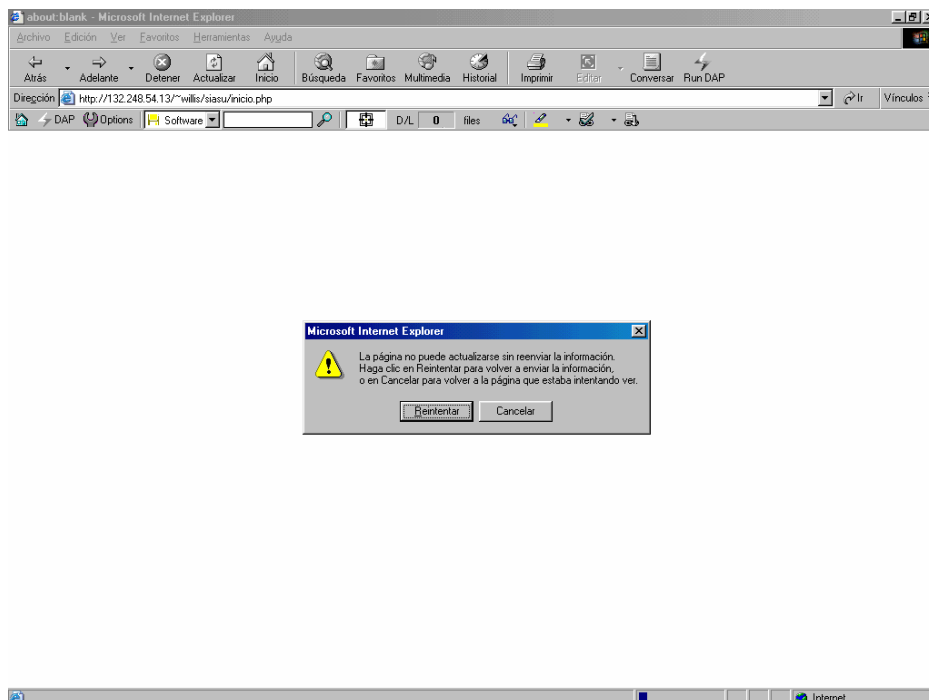


Fig. 21

Glosario

KERNEL:

El Kernel es considerado el núcleo o parte fundamental de un Sistema Operativo. La parte que se carga inicialmente y permanece en memoria, encargada de proporcionar los servicios básicos de gestión de memoria, ficheros y recursos.

TERMINAL:

Una terminal es un punto de introducción y salida de datos de una computadora central. Se les suele denominar "tontos" si carecen de procesador, es decir, dependen por completo del servidor, e "inteligentes" en caso contrario.

API:

Application Program Interface. Interfase de Aplicación del Programa. Es el conjunto de rutinas del sistema que se pueden usar en un programa para la gestión de entrada/salida, gestión de ficheros etc.

ODBC:

ODBC es un intermediario entre bases de datos y aplicaciones, cuya tarea es sostener una conversación de preguntas y respuestas entre dos "sujetos" que no hablan el mismo idioma y que gestionan sus recursos de forma diferente.

JDBC:

JDBC (Java Database Connectivity) La conectividad de la base de datos de Java es un marco de programación para los desarrolladores de Java que escriben los programas que tienen acceso a la información guardada en bases de datos, hojas de cálculo, y archivos "planos".

SERVIDOR:

De manera genérica, es el dispositivo de un sistema que pone ciertos recursos, o resuelve las peticiones de otros elementos del sistema, denominados clientes. Estos recursos pueden ser datos, aplicaciones, impresoras y más.

CLIENTE:

Es un dispositivo conectado en red, que accede a los recursos que otro dispositivo (servidor) pone a su disposición.

PROTOCOLO:

Es un conjunto de normas y/o procedimientos para la transmisión de datos que a de ser observado por los dos extremos de un proceso comunicacional (emisor y receptor).

TCP/IP:

(Transmission Control Protocol / Internet Protocol) Es un estándar de comunicaciones muy extendido y de uso muy frecuente para software de red.

SCRIPTS:

Son pequeños programas o aplicaciones propias de un lenguaje de programación, que ayudan a realizar cierta tarea descrita.

NAVEGADOR:

Es una aplicación o programa de computadora diseñado para facilitar la visualización de páginas Web en Internet. Actualmente es posible mostrar no solo texto, sino también audio, video e incluso ejecutar software.

Creación de una Base de Datos

DIAGRAMA DE FLUJO DE DATOS

Definición:

El diagrama de flujo de datos (DFD) es una representación gráfica que describe los flujos de datos, los procesos que cambian o transforman los datos en un sistema, las entidades externas que son fuente o destino de los datos, y los almacenamientos o depósitos de datos a los cuales tiene acceso el sistema, permitiendo así describir el movimiento de los datos a través del sistema.

En síntesis, el Diagrama de Flujo de Datos describe:

- los lugares de origen y destino de los datos (los límites del sistema),
- las transformaciones a las que son sometidos los datos (los procesos internos),
- los lugares en los que se almacenan los datos dentro del sistema, y
- los canales por donde circulan los datos.

Características:

Es relevante puesto que provee comunicar diferentes modelos para así facilitar el entendimiento entre el usuario y el analista de sistemas; creado a un nivel lógico, ya que no analiza soporte físico y diseñado de manera descendente puesto que se diseña de lo general a lo particular.

Simbología:

Entidad Externa:

Una **entidad externa** consiste en un Sistema o Subsistema que provee información de entrada pero que no está comprendido en el sistema a estudiar.

Son generalmente clases lógicas de cosas o de personas, las cuales representan una fuente o destino de transacciones, como por ejemplo alumnos, profesores, etc., con las que el sistema se comunica.

Al designar alguna cosa o algún sistema como Entidad Externa estamos estableciendo implícitamente que se encuentra fuera de los límites del sistema que estamos considerando, por lo cual no nos interesa la transformación o proceso que se realiza dentro de ellos, es decir que están fuera del control del sistema que se está modelando. Estos sólo nos proveen o solicitan datos del sistema bajo consideración.

Por todo ello, ni el analista ni el diseñador pueden cambiar ni los contenidos ni la forma de trabajo de un terminador.

Para su representación se utiliza el rectángulo:



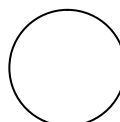
Proceso:

Un **proceso** es la acción o conjunto de acciones que permiten convertir o transformar los datos de entrada en los datos de salida del Sistema. Gráficamente indican aquellos lugares dentro del sistema en donde la información (flujos de datos) que ingresan se procesa o transforman. Es decir, son los procesos que transforman entradas de datos en salidas de información.

Su nombre se coloca mediante una frase imperativa, que consistirá idealmente de un verbo activo seguido por una cláusula objeto, cuanto mas simple mejor. Estas frases imperativas no tienen sujeto; tan pronto como se introduce un sujeto se habrá indicado como deberá realizarse físicamente la función ("El operador ingresará los datos del alumno").

Un proceso puede ser físicamente una salón repleto de alumnos, un procedimiento, o una combinación de actividades manuales y automatizadas.

Es representado por un círculo:



Flujo de datos:

Representa una transferencia de información desde su origen hasta su destino, es decir que representa una estructura de datos en movimiento de una parte del sistema a otro.

Un flujo muestra las interfaces entre los elementos del DFD. Puede imaginarse como una tubería por donde se envían paquetes de datos, pero deberá tener una descripción de su contenido la cuál deberá elegirse de forma que sea lo más útil posible a los usuarios que revisen el DFD.

La flecha indica la dirección del flujo:

**Almacén o archivo:**

Representa un archivo lógico en donde se agregan o de donde se extraen datos. Es una estructura de datos, pero estática.

Puede ser físicamente un archivo de tarjetas, una microficha, un archivo, o un archivo en cinta o disquete.

Gráficamente se utiliza un rectángulo sin uno de los bordes laterales:



Deberá elegirse el nombre que sea más descriptivo para el usuario, que identifique los paquetes de datos que contiene.

MODELO ENTIDAD – RELACIÓN

Un sistema manejador de base de datos es un conjunto de archivos interrelacionados y una serie de programas que permiten a varios usuarios tener acceso a estos archivos y modificarlos. Uno de los objetivos principales de un sistema de base de datos es proporcionar a los usuarios una visión abstracta de la información, es decir contar con la información que necesitan pasando por alto la forma en como estos datos se almacenan o se publican.

Modelos de datos

Es necesario definir el concepto de modelo de datos para poder describir la estructura de una base de datos. Este modelo de datos es un grupo de herramientas conceptuales que nos ayudan a describir los datos, sus relaciones, su semántica y sus limitantes. Existen varios modelos de datos diferentes, pero para este trabajo nos basaremos sobre uno en específico, el modelo entidad-relación. Decidimos basarnos en este modelo por que, durante la elaboración de este y varios proyectos más, nos hemos dado cuenta de que es un modelo de datos conveniente para el diseño de bases de datos.

MODELO ENTIDAD-RELACIÓN

Este modelo de datos se basa en una apreciación del mundo real y esta diseñado de forma abstracta para poder ser aplicado. Dicho modelo consiste en un conjunto de objetos llamados *entidades*, y las *relaciones* que entre estos existen.

Una entidad puede ser cualquier objeto o evento que a parte de distinguirse de otros por ciertas particularidades (atributos) que lo describen, es también una fuente de datos. No necesita ser algo “tangible” como lo es el concepto de un libro o una computadora, puede ser también una abstracción como una fiesta o un concepto.

Se le llama *conjunto de entidades* al conjunto de entidades del mismo tipo.

Los atributos son características propias de cada uno de las entidades, y para cada atributo existe un rango de valores permitidos al cuál se le conoce como *dominio* del atributo.

Por ejemplo, el dominio para cierto atributo podría ser conjunto de todas las cadenas de texto de cierta longitud.

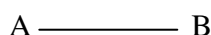
De este modo cada entidad posee un conjunto de atributos los cuales se representan por las parejas: nombre del atributo y valor del dato.

Las relaciones son asociaciones entre varias de estas entidades, y al igual que las entidades, un *conjunto de relaciones* no es más que un grupo de relaciones del mismo tipo.

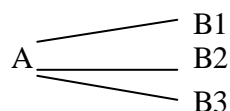
Pero este modelo no solo se basa en entidades y sus relaciones; se cuenta además con ciertas condiciones que se deben cumplir para poder llevar a cabo la relación entre las entidades, y una de estas es la cardinalidad de mapeo, la cuál expresa el número de entidades con las que puede asociarse otra entidad por medio de un conjunto de relaciones.

Existen varios tipos de relaciones, los cuales cumplen con la cardinalidad de mapeo:

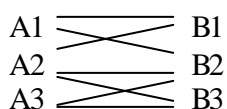
Uno a uno (1:1). Una entidad esta relacionada con otra sola entidad y esta última entidad esta relacionada únicamente con la entidad anterior.



Uno a Muchas (1:M). Una entidad esta asociada a varias entidades de otro conjunto, pero cada una de estas entidades esta asociada a una sola entidad del primer conjunto



Muchos a muchos (M:M). Una entidad esta relacionada con varias entidad de otro conjunto, y una entidad de ese conjunto se puede encontrar asociada a cualquier entidad del primer conjunto.



La cardinalidad de mapeo adecuada estará dada por la manera en que nosotros hayamos modelado nuestro sistema (o mundo).

Las llaves son datos que se encuentran en un registro y que se emplean como identificadores para éste, de esta forma podemos distinguir la entidades y las relaciones unas de otras. Regularmente se manejan dos tipos de llaves:

Llave primaria, es un o varios atributos que, juntos, permiten identificar en forma única y de manera exclusiva a una entidad dentro del conjunto de entidades. Se simboliza por las letras PK, por sus siglas en ingles (Primary Key). Esta debe cumplir con cierta reglas como lo son:

- No acepta valores nulos
- No pueden existir el mismo valor en más de un atributo
- No puede cambiarse el valor
- Puede ser asignada por el usuario administrador o por el propio sistema dependiendo de la aplicación.

Si se desea que las entidades o tablas tengan la “fuerza” para no tener que depender de otras tablas o entidades (entidades dominantes y entidades subordinadas), es necesario que cada una de las entidades cuente con una llave primaria.

Llave foránea, es un atributo para cierta entidad que no corresponde a su llave primaria pero que si lo es para otra entidad.

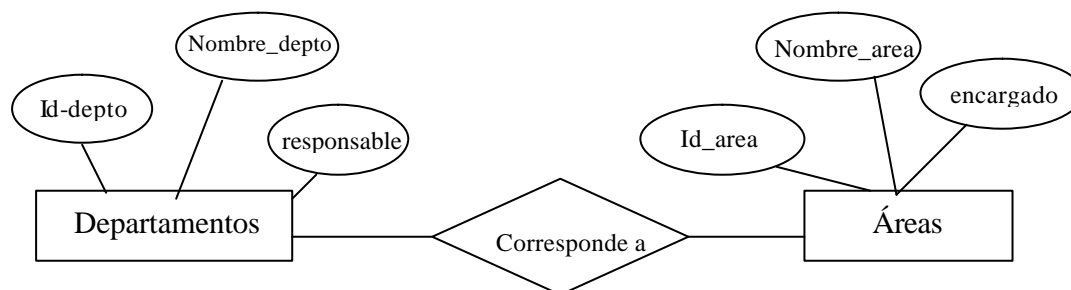
Diagrama Entidad-Relación

La estructura de una base de datos se puede expresar de manera gráfica, y para el modelo entidad-relación existe una técnica que nos ayuda a representar gráficamente la estructura lógica de la base de datos. A esta representación se le conoce como *diagrama entidad-relación*.

Básicamente, un diagrama entidad-relación se integra con los siguientes componentes:

- Entidades, estas se representan por medio de un rectángulo que contiene en su interior el nombre de dicha entidad.
- Atributos, los cuales están representados por una serie de elipses que se enlazan a la entidad correspondiente, el nombre del atributo esta envuelto por la elipse.
- Relaciones, están representadas por rombos los cuales se ligan con las distintas entidades. En su interior se coloca el nombre de la relación.
- Las uniones, correspondientes a enlazar las distintas entidades ya sea con sus atributos o con las relaciones que las afectan, se representan por medio de líneas.

Una representación de este diagrama Entidad-Relación se ilustra en la siguiente figura:



En base a una representación gráfica nosotros podemos diseñar lo que sería nuestra base de datos. Una base de datos que esta apoyada en un diagrama entidad-relación puede ser representada por un conjunto de *tablas*. De esta forma para cada entidad se cuenta con una tabla única que poseerá el mismo nombre asignado a su entidad correspondiente y, a su vez, contendrá un cierto número de columnas todas estas con nombres únicos.

Departamentos

Id_depto	Nombre_depto	Responsable
8976	Termodinámica	Ing. Marian Aburto
8642	Calculo II	Ing. Margarita Carrera
8542	Algebra	Ing. Juan Gabriel García
8037	Dinámica	Ing. Pedro Hugo Soto
8123	Física	Ing. Marlene Silvestre

Normalización.

Las reglas de Normalización están encaminadas a eliminar redundancias e inconsistencias de dependencias en el diseño de las tablas.

Existen formas de normalizar una base de datos. Lo que a continuación se encuentra son pasos progresivos para normalizar (también conocidas como formas normales FN), de modo que se logre crear una base de datos funcional y eficiente.

Se requiere una base de datos con la información de usuarios, y los datos a guardar son el nombre, la escuela, la dirección de la escuela, y algún e-mail, o bien una URL. Obtendríamos una tabla como la que se muestra:

USUARIOS

Nombre	Escuela	Dirección_escuela	URL1	URL2
Marian	Preparatoria N.5	Clza. del Hueso s/n	marian@prepa5.unam.mx	www.unam.mx
Eric	C. C. H. Sur	Blv. de la Luz s/n	erik@cchsur.unam.mx	www.unam.mx

Esta es una tabla prototipo a la cuál no se le han aplicado alguna de las reglas de normalización, por lo que diremos que la anterior tabla esta en nivel de Formalización Cero.

Si se solicitara una tercera URL se tendría que añadir otro campo/columna a nuestra tabla y tener que reprogramar toda la entrada de datos de nuestro código PHP. Pero lo que se desea es crear un sistema funcional que pueda crecer y adaptarse fácilmente a los nuevos requisitos.

Se revisaran las reglas del Primer Nivel de Formalización-Normalización (F/N), y las aplicaremos a la tabla anterior.

Primer nivel de F/N.

1. Eliminar los grupos repetitivos de las tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

Es notorio que estamos rompiendo la primera regla cuando repetimos el campo url2. Y analizando la tercera regla, no contamos con una llave primaria. La regla tres básicamente significa que tenemos que poner una campo, en este caso, tipo contador autoincrementable para cada registro. De otra forma, tendríamos problemas si nos encontramos con dos usuarios llamados Marian y queremos diferenciarlos.

Una vez aplicado el primer nivel de F/N nos encontraríamos con la siguiente tabla:

USUARIOS

id_user	nombre	escuela	direccion_escuela	url
1	Marian	Preparatoria N.5	Clza. del Hueso s/n	marian@prepa5.unam.mx
1	Marian	Preparatoria N.5	Clza. del Hueso s/n	www.unam.mx
2	Erik	C.C.H. Sur	Blv. de la Luz s/n	erik@cchsur.unam.mx
2	Erik	C.C.H. Sur	Blv. de la Luz s/n	www.unam.mx

Ahora que se tiene solucionado el problema en la limitante del campo url, nuestra tabla está en el primer nivel de F/N. Sin embargo se genera otros problemas, ya que cada vez que introducimos un nuevo registro en la tabla **usuarios**, tenemos que duplicar el nombre de la escuela y del usuario. No sólo nuestra Base de Datos crecerá muchísimo, sino que será muy fácil que ésta se corrompa si escribimos mal alguno de los datos redundantes. Por tal motivo se aplica el segundo nivel de F/N.

Segundo nivel de F/N

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Hemos separado el campo url en otra tabla, de forma que podemos añadir más en el futuro sin tener que duplicar los demás datos. En este caso tenemos que usar una clave primaria para relacionar estos campos:

USUARIOS

id_user	nombre	Escuela	direccion_escuela
1	Marian	Preparatoria N.5	Clza. del Hueso s/n
2	Erik	C.C.H. SUR	Blv. de la Luz s/n

URLs

id_url	relac_id_user	url
1	1	marian@prepa5.unam.mx
2	1	www.unam.mx
3	2	erik@cchsur.unam.mx
4	2	www.unam.mx

En este momento se tienen tablas separadas y la clave primaria en la tabla **usuarios**, id_user, esta relacionada ahora con la clave externa en la tabla **urls**, relac_id_user. El problema radica al dar de alta a más alumnos ya que el nombre de la escuela y su dirección estarían duplicándose, esta es otra situación que puede inducirnos a introducir errores en nuestros datos. Se procede a aplicar el tercer nivel de F/N:

Tercer nivel de F/N.

1. Eliminar aquellos campos que no dependan de la clave.

Se aprecia que el nombre de la escuela y su dirección no tienen nada que ver con el campo id_user, así que tienen que tener su propio id_escuela:

USUARIOS

Id_user	nombre	relac_id_escuela
1	Marian	1
2	Eric	2

ESCUELAS

id_escuela	escuela	direccion_escuela
1	Preparatoria N.5	Clza. del Hueso s/n
2	C.C.H. SUR	Blv. de la Luz s/n

URLs

id_url	relac_id_user	url
1	1	marian@prepa5.unam.mx
2	1	www.unam.mx
3	2	erik@cchsur.unam.mx
4	2	www.unam.mx

Ahora se tiene la clave primaria `id_escuela` en la tabla **escuelas** relacionada con la clave externa `relac_id_escuela` en la tabla **usuarios**, y con esto es posible añadir un número ilimitado de usuarios para la Preparatoria N.5 mientras que sólo tenemos que insertar el nombre 'Preparatoria N.5' una vez.

Las tablas de usuarios y urls pueden crecer todo lo que quieran sin duplicación ni corrupción de datos.

Si se analiza el campo `url` se aprecia que existe duplicación de datos. Esto es aceptable si la entrada de datos de este campo se solicita al usuario en nuestra aplicación para que teclee libremente su url, y por lo tanto es sólo una coincidencia que Marian y Eric teclearon la misma url.

El problema crece si en lugar de entrada libre de texto se utilizara un menú desplegable con cierto numero de urls predefinidas. Lo que seguiría es llevar el diseño de la Base de Datos al siguiente nivel de F/N, el cuarto, muchos desarrolladores lo pasan por alto porque depende mucho de un tipo muy específico de relación, la relación 'muchos -a-muchos'.

Relaciones entre los Datos

Retomando la cardinalidad de mapeo, se tienen 3 tipos de relaciones entre los datos: uno-a-uno, uno-a-muchos y muchos-a-muchos.

En el ejemplo del Tercer Nivel de F/N, en el que se tiene a un usuario relacionado con varias urls. Se quiere llegar a una estructura para permitir que varios usuarios estén relacionados con varias urls y así tendremos una relación muchos-a-muchos. Veamos como quedarían nuestras tablas antes de seguir con este planteamiento:

USUARIOS

id_user	Nombre	relac_id_escuela
1	Marian	1
2	Eric	2

ESCUELAS

id_escuela	escuela	direccion_escuela
1	Preparatoria N.5	Clza. del Hueso s/n
2	C.C.H. SUR	Blv. de la Luz s/n

URL

id_url	url
1	marian@prepa5.unam.mx
2	erik@cchsur.unam.mx
3	www.unam.mx

RELACION_URL

id_relacion	relac_id_url	Relac_id_user
1	1	1
2	1	3
3	2	2
4	2	3

Para disminuir la duplicación de los datos se ha creado una tabla que sólo tiene claves externas y primarias, relacion_url. Este proceso nos llevará al Cuarto Nivel de F/N. De forma tal que se han removido las entradas duplicadas en la tabla urls creando la tabla relacion_url, la cuál expresa exactamente la relación que ambos, Marian y Eric, tienen entre cada uno de ellos, y entre ambos, es decir las urls. El Cuarto Nivel de F/N. supone:

Cuarto Nivel de F/N.

1.- En las relaciones muchos-a-muchos, entidades independientes no pueden ser almacenadas en la misma tabla.

Este cuarto nivel solo es aplicable para este tipo de relaciones, por lo que la mayoría de los desarrolladores pueden ignorar esta regla de forma correcta. Pero es muy útil en ciertas situaciones como esta.

Hemos optimizado nuestra tabla urls eliminado duplicados y hemos puesto las relaciones en su propia tabla, esto es la finalidad de la Normalización.

EL DICCIONARIO DE DATOS

Esta parte pretende representar la información que ha sido señalada en el diagrama de flujo de datos. Es posible decir formalmente que:

El Diccionario de Datos (DD) dentro de un Manejador de Bases de Datos contiene el directorio de todos los elementos de información en la base de datos y también la descripción de los datos.

El Diccionario de Datos es un sistema de información, el cuál va a contener la información acerca de los elementos en la base de datos, tales como sus atributos, su uso y sus definiciones estándar; sin incluir los valores de los objetos de datos.

La naturaleza de esta información debe estar en dos áreas: la conceptual y la de implementación.

La notación de un diccionario de datos facilita la representación de los datos. Una de las características del diccionario de datos es que debe contener la explicación de todos los términos utilizados. Dichas especificaciones deben ser precisas.

La base de datos se expande a medida que se desarrollan e integran las aplicaciones. Se introducen nuevos campos de datos y además puede ser necesario modificar los elementos utilizados durante el diseño de la base de datos. El recurso que proporciona información uniforme y centralizada acerca de todos los datos se llama diccionario de datos (DD).

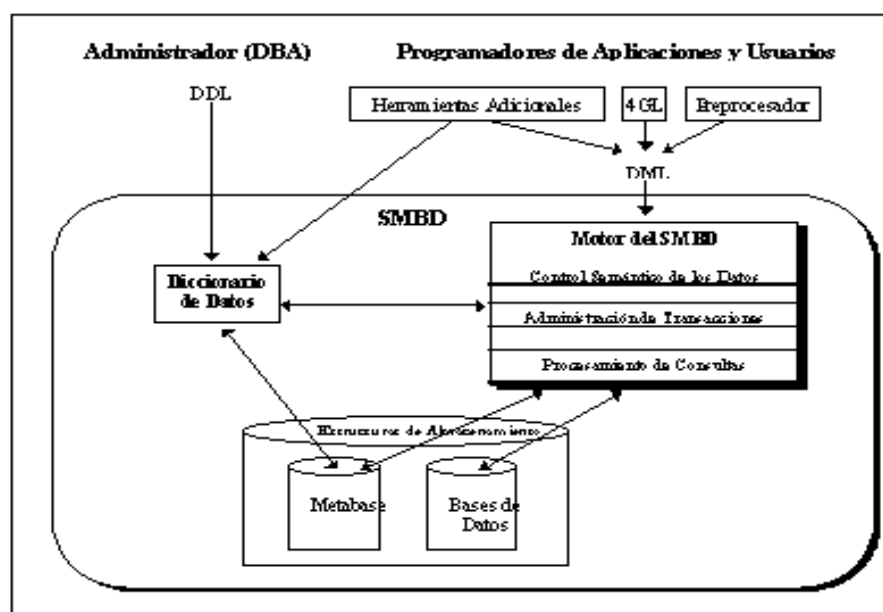
La administración de los datos almacenados en un Sistema Manejador de Bases de Datos (SMBD) requiere a su vez de un sistema de administración de los elementos que definen a la base de datos y sus relaciones constitutivas. El Diccionario de Datos (DD) contiene la información acerca de los elementos que constituyen la base de datos (tablas, vistas, índices, entre otros), tales como sus atributos, su uso y sus definiciones estándar. Estos son almacenados también en una base de datos, la cual lleva el control de las relaciones usadas por

el sistema para la administración de los datos, o sea, el *catálogo del sistema*. La siguiente figura muestra como está implantado el DD dentro del SMBD.

Entre las características representativas del DD deben estar:

- presentar un rápido acceso a la base de datos
- capacidad de generación de directorio
- lenguaje de definición de datos
- control de acceso y seguridad de datos
- control de presentación
- características de mejoras a la independencia de datos
- independencia a los dispositivos
- capacidades de localización de datos
- modelos de datos multi-sistemas y un soporte de consultas catalogadas.

Entre las entidades y relaciones que deben ser soportadas están las áreas, archivos, ítems, trabajos, procedimientos, programas, registros, reportes, esquemas, conjuntos, sub-esquemas, sistemas y transacciones.



Tradicionalmente se definen tres tipos de esquemas dentro de una base de datos:

- Esquema Conceptual.- Es la visión completa, lógica, de la base de datos, como es vista por el Data Base Administrator (DBA). Contiene todas las definiciones del diccionario de datos y todas las restricciones asociadas existentes.

- Esquema Físico.- Es la visión técnica completa de la base de datos como es vista por el sistema operativo. Contienen todas las descripciones físicas de los archivos. Implementa el área para los mapeos hacia archivos, la implementación de los accesos a archivos (hashing, indexado, etc.) y las interrelaciones entre archivos (apuntadores, arreglos de apuntadores, índices secundarios, etc.).

- Esquema externo o sub-esquema.- Es la parte que el usuario o un programa ven de la totalidad de la base de datos. Los sub-esquemas son definidos en el DD por el DBA y el SMDB lo usa, para comprobar que una llamada tenga el permiso de acceso

BIBLIOGRAFÍA

Tesis: “Diseño de un sistema de control de documentos para UNICA ”

Beatriz Barrera Hernández

Págs. 164

Creación de aplicaciones Web con PHP 4

Tobias Ratschiller y Till Cerken

Prentice Hall

Madrid, España 2001

Referencias:

<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/analisis.asp>

www.monografias.com/trabajos7/resi/resi.shtml

<http://redii.iimas.unam.mx/redii/Ciclo2000/REDII2000/>

[GamboaRodriguez/RESUMEN.htm](http://www.gamboaRodriguez.com/RESUMEN.htm)

<http://www.cosaslibres.com/sisope.html>

<http://usuarios.lycos.es/kalimero666/zip/asp.pdf>

<http://www.phpbuilder.com/>

<http://www.php.net>

<http://www.desarrolloweb.com/articulos>