



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

**“SISTEMA DE CONTROL DE LAS EVALUACIONES DE LOS  
BECARIOS PARA LA UNIDAD DE SERVICIOS DE  
CÓMPUTO ACADÉMICO”  
(SICEB)**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERA EN COMPUTACIÓN

P R E S E N T A:

**EVANGELINA ALLIER MENDOZA**

DIRECTOR DE TESIS:  
ING. MARÍA DEL ROSARIO BARRAGÁN PAZ



Ciudad Universitaria México, D.F.

2005



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# *Dedicatorias*

A MI MAMÁ, CAROLINA ALLIER, ESTE TRABAJO ES PARA TI, ME DISTE LA VIDA MISMA, Y NUNCA TENDRÉ PALABRAS SUFICIENTES PARA AGRADECÉTELO, TE HAS DEDICADO A MÍ EN CUERPO Y ALMA, ERES LO MAS PRECIADO QUE TENGO, TE AMO.

A MI ABUELO, GUILLERMO ALLIER NO PUDISTE VERME TERMINAR LA CARRERA, TE EXTRAÑO ESPERO QUE DESDE DONDE TE ENCUENTRES, ESTÉS ORGULLOSO DE TU NIETA, LA CHIARRÍS.

A MI ABUELA, EVANGELINA MENDOZA TENÍAS ILUSIÓN DE VERME TITULADA, ME HUBIERA GUSTADO QUE ESTUVIERAS CONMIGO EN ESTE MOMENTO TAN IMPORTANTE.

A G Y C, EN EL CIELO LES HACÍAN FALTA ÁNGELES, POR LO QUE TUVIERON QUE IR PARA ALLÁ LOS EXTRAÑO, ESTO ES PARA USTEDES COMO SE LOS PROMETÍ MILES DE VECES.

A P Y C, ESTÁN CONMIGO CADA DÍA, LAS ADORO, SON UN MOTIVO PARA DESPERTAR Y SEGUIR VIVA, GRACIAS POR APARECER EN MI VIDA.

# Agradecimientos

A DIOS, POR PERMITIRME VIVIR TODOS ESTOS AÑOS, Y VER MI SUEÑO DE TERMINAR LA UNIVERSIDAD CUMPLIDO AL FIN, ESPERO ME DE MUCHOS AÑOS MAS.

A LA UNAM Y LA FACULTAD DE INGENIERÍA QUE DESDE EL MOMENTO QUE ME ACOGIERON SE VOLVIERON UN SEGUNDO HOGAR DÁNDOME LA OPORTUNIDAD DE REALIZAR UN SUEÑO QUE TUVE DESDE LA INFANCIA.

A MI PADRE, POR DARMME JUNTO CON MI MAMÁ LA OPORTUNIDAD DE VIVIR.

A LA ING. MA. DEL ROSARIO BARRAGÁN POR SU APOYO EN TODO MOMENTO DESDE QUE INGRESÉ A UNICA, HASTA LA REALIZACIÓN DE ESTA TESIS

A UNICA, QUE DESDE EL PRIMER MOMENTO QUE ENTRE ME BRINDÓ MUCHAS OPORTUNIDADES, ENTRE ELLAS, LA DE CONOCER GENTE MUY VALIOSA.

AL DR. JUAN VICENTE MÉNDEZ HUERTA, QUIEN ME BRINDÓ SU AMISTAD MUCHO ANTES DE EMPEZAR LA CARRERA Y UNA VEZ EN ELLA, ME DIÓ SU APOYO HASTA EL FIN.

A MIS AMIGOS DE LA CARRERA: MARISOL, GABRIEL, FARAH, AXCANÁ, JESÚS, LALO, ILP, NADIA QUE ESTUVIERON CONMIGO A LO LARGO DE ESTE CAMINO, QUE INCLUYÓ NOCHES DE DESVELO, EXÁMENES JUNTO, PASEOS, FIESTAS, Y HASTA UNA LARGA HUELGA, PERO SOBRE TODO LA SATISFACCIÓN DE QUE TERMINAMOS Y TARDE O TEMPRANO ESTAMOS LLEGANDO A SER INGENIEROS.

A MIS COMPAÑEROS DE UNICA: ARMANDO, DURÁN, LIDYA, ERICK, ANTONIO, FRANCISCO, MEMO, ANDRÉS, SANDRA POR SU APOYO DURANTE MI ESTANCIA EN LA UNIDAD

A MIS AMIGOS DE MUCHOS AÑOS: ADRIANA, ISAAC, MARIANO, ESTÁN CONMIGO, AUNQUE NO LOS VEA DURANTE AÑOS.

DE NIÑA CUANDO PASABA POR LA UNIVERSIDAD EN COCHE  
CON MI MAMÁ ME JURÉ QUE ALGÚN DÍA CAMINARÍA POR EL  
CAMPUS Y ESTUDIARÍA AQUÍ.....

.....FUE UNA ILUSIÓN DE MUCHOS AÑOS.....

.....EL PRESENTE TRABAJO ES LA PRUEBA DE QUE LOGRÉ  
HACERLO REALIDAD

---

# ÍNDICE

DEDICATORIAS .....	i
AGRADECIMIENTOS .....	ii
ÍNDICE .....	iv
OBJETIVOS .....	vii
<b>MARCO GENERAL.....</b>	<b>1</b>
1. Antecedentes de UNICA .....	1
1.1 Académico .....	1
1.2 Objetivos y Metas .....	3
1.3 Servicios que ofrece .....	3
1.4 Organización y Funciones por Departamento .....	6
1.5 Necesidad de un sistema de Evaluación de Becarios ....	9
1.6 Alcance y justificación del sistema .....	10
<b>1. LA PROGRAMACIÓN ORIENTADA A OBJETOS ...</b>	<b>12</b>
1.1 Características de la Programación Orientada a Objetos ...	13
1.2 Características de los Objetos .....	15
1.3 Elementos de la Programación Orientada a Objetos .....	19
1.3.1 Abstracción .....	19
1.3.2 Encapsulamiento .....	19
1.3.3 Polimorfismo .....	20
1.3.4 Herencia .....	20
1.3.5 Mensajes .....	21
1.4 Modularidad .....	21
1.4.1 La estructura de un módulo .....	22
1.4.2 Reglas de modularización .....	22
1.4.3 Diseño de módulos .....	23

---

<b>2. EL LENGUAJE DE MODELADO UNIFICADO .....</b>	<b>24</b>
2.1 Funcionalidad de UML .....	25
2.2 Descripción de conceptos básicos .....	26
2.3 Fases del desarrollo de un sistema .....	26
2.3.1 Análisis de requerimientos .....	27
2.4 Elementos comunes a todos los diagramas .....	28
2.5 Diagramas dentro de UML .....	33
2.5.1 Diagramas de caso de uso .....	33
<b>3. REQUERIMIENTOS DEL SISTEMA .....</b>	<b>56</b>
3.1 Antecedentes .....	56
3.2 Módulos del Sistema .....	58
3.2.1 Módulo I: Evaluaciones .....	58
3.2.2 Módulo II: Proyectos .....	71
3.2.3 Módulo III: Ex Becarios .....	73
3.2.4 Módulo IV: Reporte Semestral .....	74
<b>4. TECNOLOGÍA EMPLEADA Y MODELADO DEL SISTEMA .....</b>	<b>75</b>
4.1 Tecnología Empleada .....	75
4.1.1 Interfáz Gráfica .....	75
4.1.2 Modelado del Sistema .....	76
4.1.3 Base de datos .....	77
4.2 El Sistema de Inscripción y Control de Cursos y el Sistema de Evaluación de Becarios, su interacción .....	78
4.3 Modelado del Sistema.....	83
4.3.1 Análisis .....	83

---

<b>5. PRUEBAS Y PUESTA EN MARCHA DEL SICEB .....</b>	<b>95</b>
5.1 Las pruebas de software .....	95
5.2 Diseño de casos de prueba .....	96
5.3 Metodología de las pruebas .....	96
5.3.1 Pruebas de caja blanca .....	96
5.3.2 Pruebas de caja negra .....	97
5.4 Plan de pruebas del SICEB .....	97
5.5 Puesta en marcha de la prueba de caja negra .....	97
5.6 Aplicación del plan de pruebas .....	98
<b>6. CONCLUSIONES .....</b>	<b>147</b>
6.1 Contribución de la Tesis .....	147
6.2 Perspectiva a Futuro.....	149
<b>ÁPENDICE A</b>	
Manual de Usuario .....	150
<b>ÁPENDICE B</b>	
Manual de Instalación .....	172
<b>BIBLIOGRAFÍA .....</b>	<b>239</b>

**Objetivos y resultados  
esperados**

The image shows a table with a header row and a vertical line on the right side. The header row contains the text "Objetivos y resultados esperados". The table is defined by a horizontal line at the top, a vertical line on the right, and a horizontal line at the bottom. The text is centered within the header cell.

Objetivos y resultados esperados
----------------------------------

## OBJETIVO GENERAL

**Diseñar un sistema para la evaluación de los becarios de la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería.**

### Objetivos particulares

- I. Contar con un sistema capaz de evaluar a los becarios
- II. Hacer que el sistema interactúe con el ya
- III. Que el sistema interactúe con el sistema de Control de Asesorías (CORA)
- IV. Contar con una interfaz para generar estadísticas
- V. Generar reportes de las actividades de cada becario
- VI. Consulta por becario.
- VII. Que cada jefe de departamento pueda evaluar a sus becarios.

## RESULTADOS ESPERADOS

En la implementación de este sistema se esperan los siguientes resultados:

- ❖ Hacer más rápido y eficiente el proceso de evaluación de becarios.
- ❖ Realizar reportes del desarrollo interno de cada becario.
- ❖ Los departamentos:

- Departamento de Servicios Académicos
- Departamento de Investigación y Desarrollo
- Departamento de Redes y Operación de Servidores
- Área de Seguridad en Cómputo
- Control de Salas de Computo de UNICA

Contarán con su propia interfaz de evaluación para los becarios que se encuentren en cada uno de ellos.

- ❖ Reducir el tiempo de evaluación
- ❖ Interacción con el Sistema de Inscripción y Control de Cursos (SICC) y el CORA

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

## 1.1.- 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 fue creado en 1972 por acuerdo del entonces Director Juan Casillas García de León, quién designó al Ing. Heriberto Olgún Romo para ser su fundador. Entre los principales servicios que brindaba eran: la elaboración de sistemas, programas de apoyo académico-administrativo, asesorías, cursos para alumnos, personal académico y 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 figura. 1.1.1

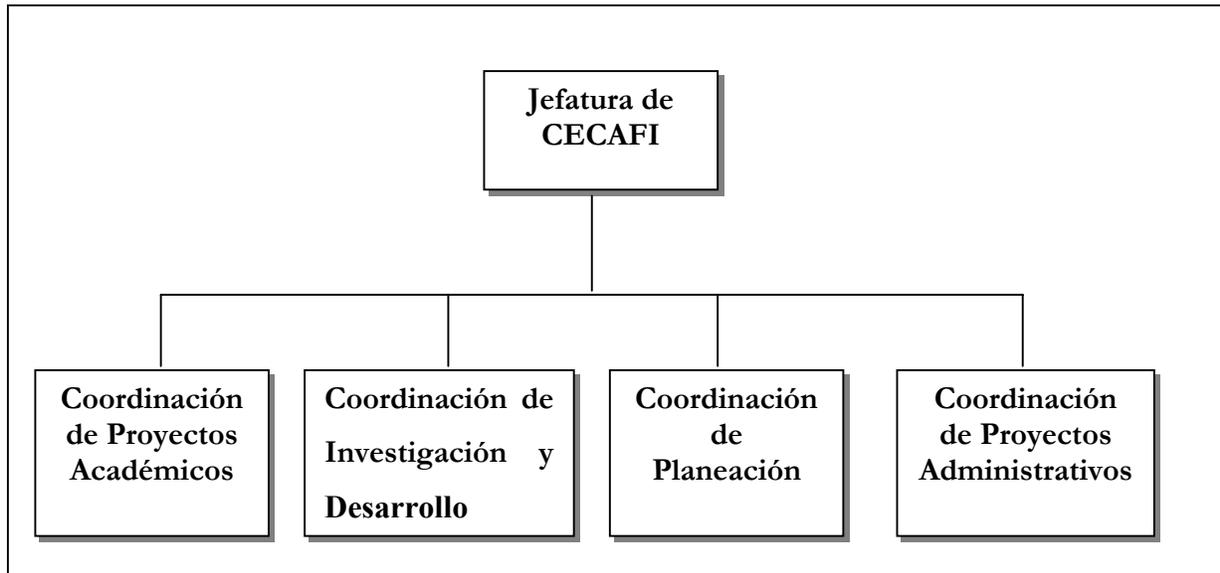
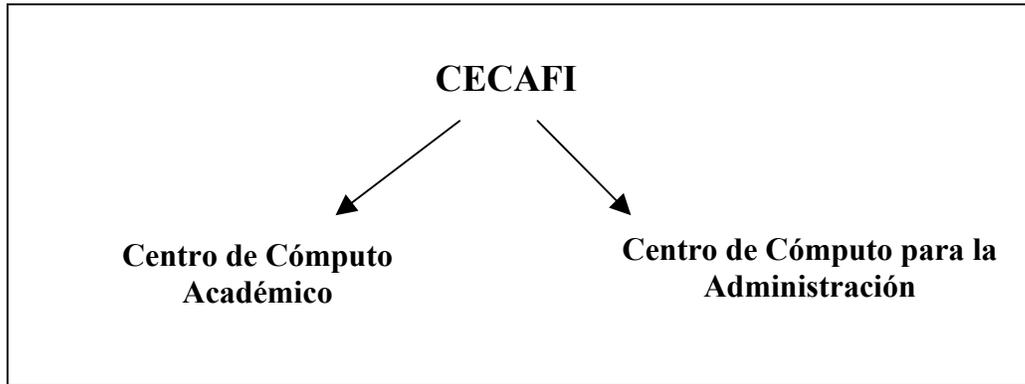


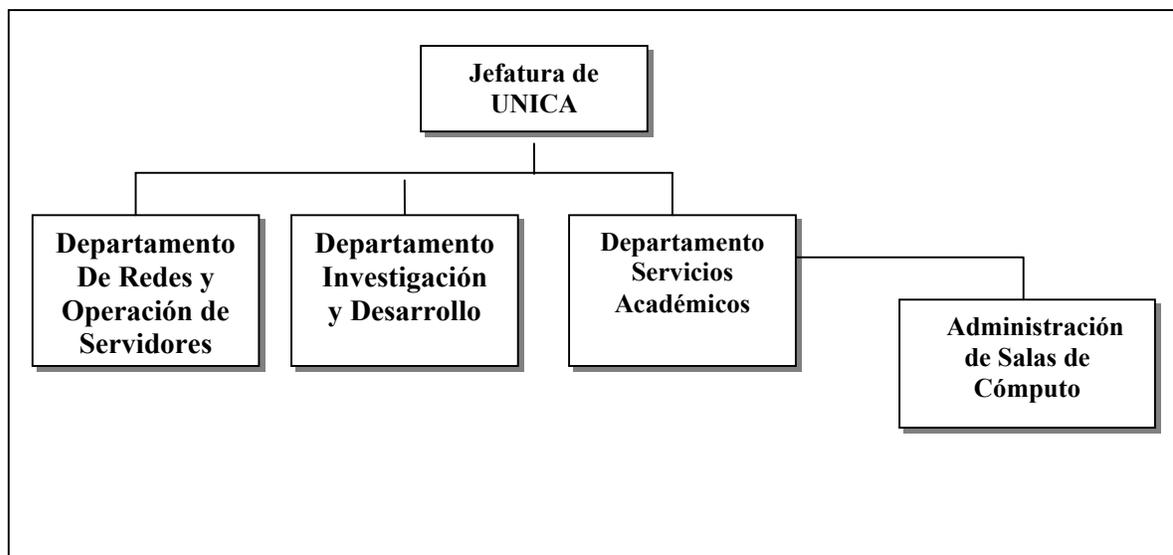
Fig. 1.1.1 Estructura de CECAFI hasta 1992

En 1993 el CECAFI se reestructuró dividiéndose en dos Centros de Cómputo como lo muestra la figura 1.1.2:



*Fig. 1.1.2 Reestructuración de CECAFI 1993*

Hasta entonces el Centro de Cómputo Académico tuvo las funciones de formar recursos humanos y proporcionar servicios en materia de computación; 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.



*Fig. 1.1.3 Departamentos UNICA*

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.

### **1.2.1.- *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.

### **1.2.2.- *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 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 Facultad. Actualmente se ofrecen los siguientes servicios:

### 1.3.1.- Servicios de cómputo.

Para brindar este servicio a los alumnos de ingeniería, UNICA cuenta con cinco salas. Las dos primeras en el edificio principal, las cuales son producto de una reciente remodelación, ya que sólo se contaba con una sola, dos más en el anexo y una en el edificio de posgrado.

- Salas del Edificio Principal.- Ubicadas a un lado de la librería, equipada con computadoras e impresoras para uso de los alumnos, así como estaciones de trabajo de la marca Sun y un área especial para los servidores.
- Sala 2.- Ubicada en el anexo de Ingeniería abajo del auditorio “Sotero Prieto”. Esta dividida en cuatro salas: A, B, C y D.
- Sala 3.- Ubicada en el anexo de Ingeniería (en la antigua biblioteca). Dividida de igual forma en 4 salas: E, F, G y H.
- Sala 4.- Ubicada en el Edificio de Posgrado, primer piso esta sala fue inaugurada en noviembre del 2004 y cuenta con un sistema de internet inalámbrico.

### 1.3.2.- Cursos de computación.

- *Intersemestrales.*- Cada periodo intersemestral la Facultad de Ingeniería ofrece diversos cursos de computación, los cuales son sugeridos 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, sólo 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 de clases. Estos están abiertos al 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 para empresas o personas externas a la UNAM.

1.3.3.- *Formación de recursos humanos.* Al final del 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 dándoles capacitación en cómputo a fin de contar con los mejores candidatos para formar parte de su personal. Al término 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, entre otros.

1.3.4.- *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.

1.3.5.- *Coordinación del mantenimiento preventivo del equipo de cómputo.* La Facultad de Ingeniería tiene contrato con una empresa dedicada a esta tarea, UNICA en conjunto con Secretaría Administrativa realiza un plan de mantenimiento para los equipos y verifica que se cumpla.

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

- 1.3.7.- *Coordinación del inventario de la Facultad de Ingeniería.* UNICA realiza un plan de acción para concentrar la información del inventario de equipo de cómputo con que cuenta la Facultad, coordina a todas las Divisiones y Secretarías para realizar un inventario de sus áreas.
- 1.3.8.- *Elaboración de material didáctico.* En la Unidad se elaboran notas y manuales sobre temas de computación de acuerdo a los cursos impartidos.
- 1.3.9.- *Estudio y aplicación de software.* Implica la investigación y desarrollo del software comercial para las actividades que la Facultad requiere así como la 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 y los Departamentos de: Servicios Académicos, Investigación y Desarrollo, Redes y Operación de Servidores y Seguridad en Cómputo, así como por la Coordinación de Salas de Atención a Alumnos.

En Septiembre del 2004, se instauró el departamento de Seguridad en Cómputo, esto atendiendo a las necesidades de contar con gente especializada, dedicada a la investigación y puesta en marcha de planes de seguridad para prevenir: fuga de información, hackeos contra servidores, estar al tanto de los nuevos virus informáticos, sus vacunas y también para contar con información reciente de lo que ocurre en el mundo de la seguridad en cómputo.

### ***1.4.1 Departamentos que conforman la Unidad***

#### *1.4.1.1.- Departamento de Servicios Académicos*

Se encarga de planear, organizar e impartir los cursos de cómputo que la Unidad ofrece semestre a semestre en las modalidades de: semestrales, intersemestrales, fines de semana, especiales, internos, así como los cursos requeridos para la formación de los alumnos que con candidatos a becarios.

Los cursos para candidatos son realizados siguiendo el 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 importante destacar la 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.

#### *1.4.1.2.- Departamento de Investigación y Desarrollo*

Se encarga de estar a la vanguardia en la tecnología y avances en materia de cómputo, a efecto de estar en la posibilidad de incorporarlos en el menor tiempo posible a los servicios que la Unidad ofrece a los alumnos para de esta forma mantenerlos oportunamente actualizados. También tiene a su cargo el 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 la página Web de UNICA.

#### *1.4.1.3.- 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 de la Web de la Facultad de Ingeniería. Coordina el subcomité de administradores y funge como el área que administra la red de cómputo general de la Facultad ante DGSCA.

Proporciona 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 al semestre.

#### *1.4.1.4.- Coordinación de Salas de Atención a Alumnos*

Su principal función 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 trabajos de investigación, además les proporciona servicio de correo electrónico, acceso a internet y servicios de apoyo en materia de cómputo.

Para tal efecto se cuenta con cinco salas de atención a usuarios, dos en el Edificio Principal productos de una reciente remodelación y acondicionamiento de este espacio, dos en el área sur ubicadas en el Anexo de Ingeniería, una de ellas abajo del auditorio Sotero Prieto, mientras que la otra se encuentra donde anteriormente estaba la antigua biblioteca y una última en el primero piso del edificio de posgrado, en ellas 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, se cuenta además con el servicio de asesoría especializada brindada por becarios, prácticamente durante todo el día.

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

#### *1.4.1.5.- Departamento de Seguridad en Cómputo*

Es el encargado de estar a la vanguardia en cuanto a medidas de seguridad dentro del área de cómputo, con el fin de resguardar la integridad de la información que se tiene en UNICA, así como también dar a conocer los nuevos

virus informáticos que circulan día con día en internet y que son los causantes de pérdidas millonarias en todo el mundo, por ende parte de sus funciones es dar a conocer las vacunas y medidas preventivas que se deben de tomar contra de ellos.

También esta encargado de subsanar las debilidades que pueda tener la red con la que cuenta UNICA, a fin de que los menos virus posibles entren y que sea más difícil que alguna persona realice un hackeo de los servidores.

### ***1. 5.- Necesidad de un Sistema de Evaluación de Becarios:***

La Unidad de Servicios de Cómputo Académico esta conformada en su mayor parte por estudiantes de la Facultad de Ingeniería, que luego de un proceso de admisión pasan a formar parte del “Plan de Formación de Becarios”; para poder permanecer dentro de este plan y por ende dentro de la Unidad, cada fin de semestre se somete a los becarios a una evaluación por parte de su jefe, estas junto con un reporte semestral de actividades que cada becario proporciona son revisadas por el Departamento de Servicios Académicos y finalmente son enviadas al Comité de Becas conformado por las autoridades superiores de la Facultad para considerar la permanencia de los becarios dentro del programa.

Anteriormente la Unidad de Servicios de Cómputo Académico realizaba el proceso de evaluación mediante el manejo de archivos, por lo cual era muy tardado ya que si se deseaba consultar la evaluación de algún becario de semestres anteriores se debía buscar en los archivos con la consiguiente pérdida de tiempo e incluso con el riesgo de pérdida de la información.

De igual forma y como parte complementaria de la evaluación se debe de contar con un registro confiable de los proyectos en los que participa el becario, actualmente estos se hayan en sistemas de archivos por lo que se pensó que el sistema de evaluación debía controlar este aspecto también para poder de forma sencilla y sin estar cambiado de una aplicación a otra, consultarlos.

Cuando un becario ingresa a la Unidad permanece dentro de ella laborando durante cierto tiempo, luego del cuál (por lo general al concluir sus estudios o titularse) se va a trabajar a empresas públicas, privadas o bien a realizar estudios de posgrado, por lo que se solicitó que este sistema también incluyera una parte destinada al control de los ex – becarios con el fin de seguir en contacto con ellos, de manera que algún día puedan regresar a brindar sus conocimientos a los que aun se encuentren laborando en la Unidad.

Dentro de este registro se debe de contar con los datos personales, fecha en que abandonaron la Unidad, empresa a la que fueron a trabajar o bien escuela o país donde realizan los estudios de posgrado.

Con el sistema propuesto se pretende agilizar la evaluación de los becarios al realizarla mediante un programa enfocado para tal uso, así como también contar con un registro de proyectos y registro de ex- becarios de la Unidad.

Se debe resaltar que con la elaboración de este trabajo se da 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” , explorar al máximo el software existente en la Facultad de Ingeniería así como también nos permite empezar a crear sistemas ayudándonos en la Ingeniería de Software y el Modelado de Sistemas, siendo un gran comienzo para empezar a desarrollar habilidades que en un futuro nos serán demandadas como ingenieros en sistemas, sirviendo también de práctica para los conocimientos que se adquirieron dentro de UNICA, pues son los becarios los que realizan estos proyectos.

### ***1. 6.- Alcance y justificación del Sistema***

Con el sistema propuesto se pretende agilizar la evaluación de los becarios, consulta de los proyectos y de los datos de ex – becarios.

Como ya fue mencionado en el punto anterior, el sistema de archivos utilizado actualmente para guardar las evaluaciones es poco confiable y ocasiona pérdidas de tiempo al buscar algún dato, teniendo en cuenta que este es tanto de búsqueda, como de captura y realización de la evaluación debido a que los becarios son el personal que en su mayoría conforman la Unidad; a la fecha se cuentan con 55 becarios, repartidos de la siguiente forma en los departamentos:

- DID: 15
- DROS: 11
- DSA: 9
- DSC: 10
- CSC: 10

Por lo que son 55 evaluaciones que se deben realizar.

Para los proyectos nos encontramos en el mismo caso que las evaluaciones, teniendo incluso poca información al alcance al momento de ser requerida ésta.

Los registros de los ex - becarios son pocos y con datos incompletos, por lo tanto al querer consultarlos falta información para tener completo un registro aceptable.

Para poder realizar lo anterior se necesita un sistema confiable, fácil de utilizar, que evalúe a los becarios en todos los parámetros que actualmente se ocupan, que contenga información detallada y accesible de los proyectos que se realizan y que de manera rápida provea de la información de los ex – becarios.

En resumen el SICEB aportará una mejora a la forma en la que se realizan las evaluaciones, los proyectos y el manejo de ex-becarios en UNICA.

*La programación orientada  
a objetos*

---

---

Capítulo

*1*

## 1.- La Programación Orientada a Objetos

El desarrollo de la Programación Orientada a Objetos (POO) empieza a destacar en la década de los 80, dotando a los programadores de nuevos elementos para el análisis y desarrollo de software. El principio básico es que un sistema se ve como una secuencia de transformaciones y se organiza en función de los objetos que manipulará

Su auge se inicia a raíz de que los programas estructurados comienzan a acarrear problemas de tiempo, costos y confiabilidad ya que los cambios realizados en la evolución de un programa traía consigo un sin número de modificaciones al código escrito, además de que poco a poco se fue introduciendo el concepto de reusabilidad; la programación estructurada pocas veces ofrecía la opción de adaptar o reutilizar partes de código ya hechas para otros programas, debido a que cada función o algoritmo era específico.

Por lo tanto y ante estas situaciones, surge la llamada programación orientada a objetos, la cual busca que los programas que se realicen estén organizados en la misma manera que el problema que trata de modelar es decir, se trata de representar el mundo real tan fielmente como sea posible.

La manera que tiene la POO de trasladar el mundo real a la abstracción (un programa) es visualizando todo aquello que nos rodea como objetos, que pertenecen a una clase y tienen cierto comportamiento.

Los objetos en el mundo real pueden ser una persona, una cosa o un lugar por ejemplo: un perro, una casa, una silla, etc., todos estos objetos tienen atributos que son las características que los describen: sexo, edad, alto, largo, color, raza, etc., y a su vez tienen un comportamiento que nos indica que es lo que hacen esos objetos: estudiar, dormir, caminar, etc.

La POO nos proporciona las siguientes ventajas:

- *Uniformidad:* Ya que la representación de los objetos implica el diseño y la codificación de los mismos.
- *Comprensión:* Tanto los datos que componen los objetos como los procedimientos que los manipulan están agrupados en clases que se corresponden con la información que maneja el programa.

- *Flexibilidad:* Al estar relacionados los procedimientos con los datos que manipulan cualquier cambio que se realice en ellos se verá reflejado automáticamente en cualquier lugar donde aparezcan.
- *Estabilidad:* Se pueden separar aquellos objetos que cambian a lo largo del programa de aquellos que no lo hacen, para identificar las partes que permanecen inalterables.
- *Reusabilidad:* La noción de objeto permite que diferentes programas que traten estructuras de información parecidas, reutilicen aquellos objetos o clases que previamente fueron definidas, acoplándolas a sus necesidades de esta forma el tiempo de desarrollo es recortado, ya que parte del código fue creado previamente.
- *Calidad.* Los diseños suelen tener mayor calidad puesto que se integran a partir de componentes probados que han sido verificados varias veces
- *Mantenimiento más sencillo.* El programador encargado del mantenimiento cambia un método de clase a la vez. Cada clase efectúa sus funciones independientemente de las demás.

## ***1.1.- Características de la Programación Orientada a Objetos***

### ***1.1.1.- Objetos:***

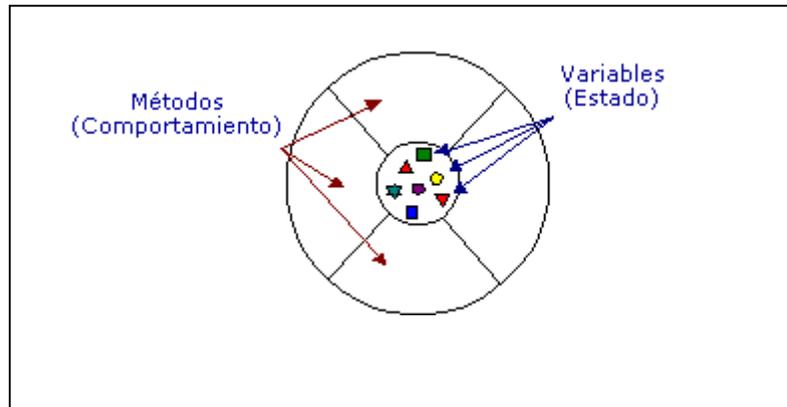
La idea de los lenguajes orientados a objetos es combinar en una sola unidad datos y funciones que operan sobre ellos, dicha unidad es conocida como objeto y dentro de ellos residen los datos y las funciones.

Las funciones conocidas como métodos dentro de la POO son el único medio para acceder a los datos internos del objeto (datos privados), con esto se asegura que los datos están ocultos y que no se pueden modificar accidentalmente. A esto se le llama encapsulamiento de la información.

Un objeto tiene atributos que son los que describen su estado, es decir las características propias de él; por ejemplo en el mundo real se tiene el objeto perro

éste tiene características como lo son: la raza, color y tipo de pelo, tamaño, etc., todas ellas nos describen de forma específica al objeto perro; estos atributos constan de dos partes: un nombre y un valor, pero también tienen acciones que son aquellas que realiza el objeto y que cambian el estado de nuestro perro: puede ladrar, comer, dormir, caminar, etc.

Dentro de la POO se representa a los objetos de la siguiente manera:



*Figura 1.1.1.1 Representación gráfica de un objeto*

Este diagrama nos ayuda a esquematizar que los datos o variables de un objeto se encuentran protegidos por los métodos y que solo al pasar por un método podremos llegar a los datos internos del objeto.

### **1.1.2.- Métodos:**

La manera en la que los objetos se comunican unos con otros y modifican sus datos internos es con los métodos, estos les dicen al objeto como se va a comportar y que es lo que va a hacer, así como el valor que van a tomar sus datos.

Estos representan las acciones que pueden ser realizadas por o sobre un objeto.

Cada método tiene un nombre y un cuerpo que realiza la acción o comportamiento asociado con el nombre del método. El cuerpo del método consta de un bloque de código que ejecuta la acción requerida.

Por lo general se pueden identificar tres tipos de métodos:

- Métodos que manipulan los datos de alguna forma específica (añadir, borrar, cambiar, actualizar, etc.)
- Métodos que realizan un cálculo u operación.
- Métodos que comprueban (monitorean) un objeto frente a la ocurrencia de algún suceso.

### ***1.1.3.- Clases:***

Son las estructuras que definen el comportamiento y los atributos generales de los objetos que pertenecen a ella. Recordando nuestro objeto perro este ese deriva de la clase perro en la cuál se definen los atributos comunes a todos los animales de este tipo como son: 4 patas, 2 orejas, 1 hocico, 1 cola, etc., y también todos sus comportamientos generales: comer, dormir, crecer, ladrar, etc.

De una clase se pueden definir muchos objetos entonces podemos tener no un perro sino muchos; las clases nos definirán modelos o plantillas que describen cómo se construyen ciertos tipos de objetos.

Cuando se crea un objeto a partir de una clase se dice que el objeto es una instancia de la clase.

## ***1.2.- Características de los Objetos***

### ***1.2.1.- Definición de Objetos:***

Un objeto es una abstracción de cosas del mundo real (es decir se lleva el mundo real al mundo de la programación) tomando en cuenta que:

- Todas las instancias tienen las mismas características generales.

- Todas las instancias siguen las mismas reglas.

Cada objeto consta de:

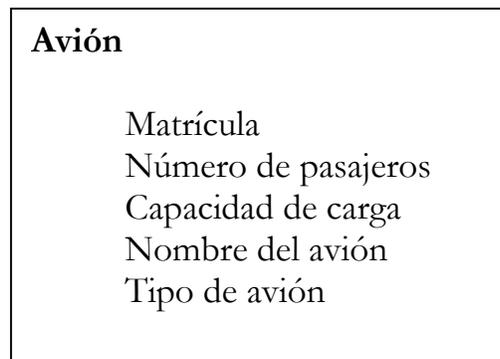
- Atributos: características que lo describen.
- Métodos: que indican como se comporta

### ***1.2.2.- Identificación de objetos***

Cuando se requiere implementar un sistema orientado a objetos se debe identificar aquellos que van a estar involucrados en nuestro sistema. Esto se obtiene examinando la descripción del problema. Los objetos pueden caer dentro de las siguientes categorías para su identificación:

- Cosas tangibles (avión, televisor, computadora, automóvil).
- Roles o papeles (gerente, empleado, cliente, médico, paciente).
- Organizaciones (empresas, división, equipo).
- Incidentes (sucesos y/o eventos).

Una vez identificados a los objetos, es necesario identificar los atributos y las operaciones que actúan sobre ellos.



*Figura 1.2.2.1 Atributos de un objeto*

### 1.2.3.- Duración de los objetos

Los objetos son entidades que existen en un determinado tiempo y para un determinado propósito por ello deben ser creados antes de utilizarse y una vez que han terminado su trabajo ser destruidos; con esto garantizamos que el objeto será creado, trabajará y se eliminará para liberar memoria y no tener miles de objetos sin usarse que solo ocupan espacio dentro de la máquina.

### 1.2.4.- Objetos Compuestos

Una característica que hace a los objetos herramientas muy versátiles y potentes es que pueden contener otros objetos, estos son llamados objetos compuestos.

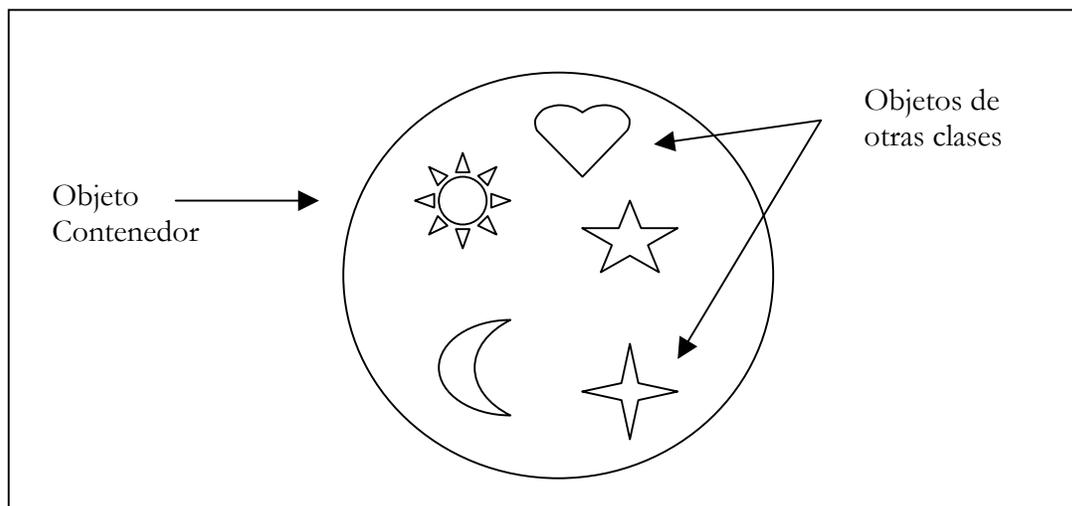


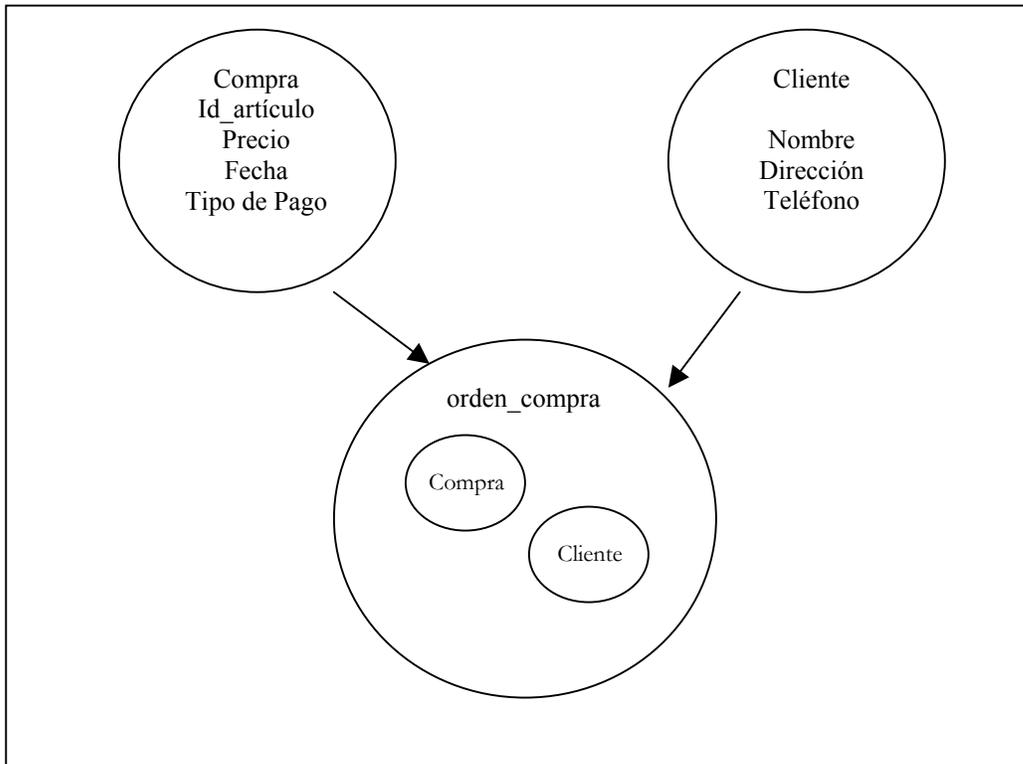
Figura 1.2.4.1 Representación gráfica de un objeto compuesto

Esta característica ofrece dos ventajas importantes:

- Los objetos contenidos pueden cambiar sin afectar al objeto que los contiene.
- Los objetos contenidos están libres para participar en cualquier número de objetos compuestos, no se limitan a estar en uno solo.

Los objetos contenidos en los objetos compuestos pueden por sí mismos ser objetos compuestos y este anidamiento puede ir hasta cualquier profundidad.

Los objetos contenidos tienen una relación entre ellos por ejemplo se tiene un objeto compuesto llamado `orden_compra` donde sus objetos componentes son: un objeto de tipo “cliente”, y un objeto de tipo “compra”:



*Figura 1.2.4.2 Objeto compuesto representación gráfica, especificando lo que contienen los objetos contenidos*

El objeto `orden_compra`, contendrá al objeto “cliente” que tiene los datos propios del cliente: nombre, dirección, teléfono y al objeto tipo “compra” con los datos de la compra: artículo comprado, fecha, monto y tipo de pago.

### ***1.3.- Elementos de la Programación Orientada a Objetos***

#### ***1.3.1.- Abstracción:***

Es la propiedad que permite representar las características esenciales de un objeto sin preocuparse de las restantes (no esenciales).

Una abstracción se centra en la vista externa de un objeto de modo que sirva para separar el comportamiento esencial de la implementación. Dentro de la POO tenemos lo que son las clases éstas se pueden definir como una descripción abstracta de un grupo de objetos

#### ***1.3.2.- Encapsulamiento***

Esta es la propiedad que nos permite asegurar que el contenido de la información de un objeto está oculta del mundo exterior: el objeto A no conoce lo que hace el objeto B y viceversa es por lo general conocido como ocultamiento de la información.

Podemos imaginar la encapsulación como un introducir el objeto dentro de una caja negra donde existen dos ranuras una la entrada y otra la salida al introducir datos por la primera se obtendrá un resultado a la salida sin conocer el proceso que tuvo lugar adentro de la caja.

Este término indica la capacidad que tiene los objetos de ocultar la información que es necesaria para su funcionamiento interno y que nadie más debe de modificar puesto que es incensaría para los demás objetos; apareciendo entonces como una forma de proteger los datos que se manipulan dentro de los objetos.

La encapsulación nos da dos grandes ventajas en los sistemas desarrollados orientados a objetos: mantenibilidad y reusabilidad.

***Mantenibilidad:*** Nos indica que cualquier programa o sistema debe ser fácilmente modificable, los cambios en las condiciones externas (definir una nueva variable) implicarán modificaciones pequeñas en la estructura general del programa.

**Reusabilidad:** Nos indica que partes del programa pueden ser reutilizados en el diseño de otros programas. Esto implica que podemos extraer objetos de un sistema e implantarlos en otro realizando solo las modificaciones necesarias en código del objeto.

### ***1.3.3.- Polimorfismo***

Indica la posibilidad de definir varias veces una operación con el mismo nombre, diferenciándolas únicamente con los parámetros de entrada, por lo tanto podemos tener diferentes métodos dentro del programa llamados igual, pero con argumentos de entrada diferentes. Esta característica los demás lenguajes de programación no los soportan ya que en cuanto se definen dos funciones que se llamen igual el compilador lo rechazará aunque los argumentos de entrada sean diferentes.

Básicamente quiere decir que un mismo nombre puede ser utilizado para varios propósitos levemente distintos.

### ***1.3.4.- Herencia***

Esta propiedad característica de la orientación a objetos permite que nuevas clases sean creadas a partir de otras creadas anteriormente logrando con esto que la nueva clase contenga la misma estructura y métodos de la anterior, pudiendo redefinir algunos o todos, declarar su propia estructura y variables. El objetivo es la reutilización de código ya que con esto el tiempo de desarrollo se decreta considerablemente debido a que se implementan clases que ya existían y rápidamente se tiene la clase que se buscaba.

La herencia se apoya en el significado de este concepto en la vida real. Por ejemplo en la vida cotidiana tenemos la clase principal llamada Animal, que tiene métodos y atributos generales a todos los animales: comer, dormir, caminar, número de patas, si tiene cola, etc., de esta clase general podemos empezar a subdividir en clases más específicas como por ejemplo por el tipo de reproducción: ovípara o vivípara, a su vez estas clases se pueden subdividir en otras clases: los vivíparos tienen a los mamíferos, los ovíparos a las aves, pájaros, reptiles, peces; así sucesivamente podemos ir definiendo subclases cada vez más específicas, cada clase que se subdivide de otra se llama clase hija y la clase de la que se desprende es la clase padre.

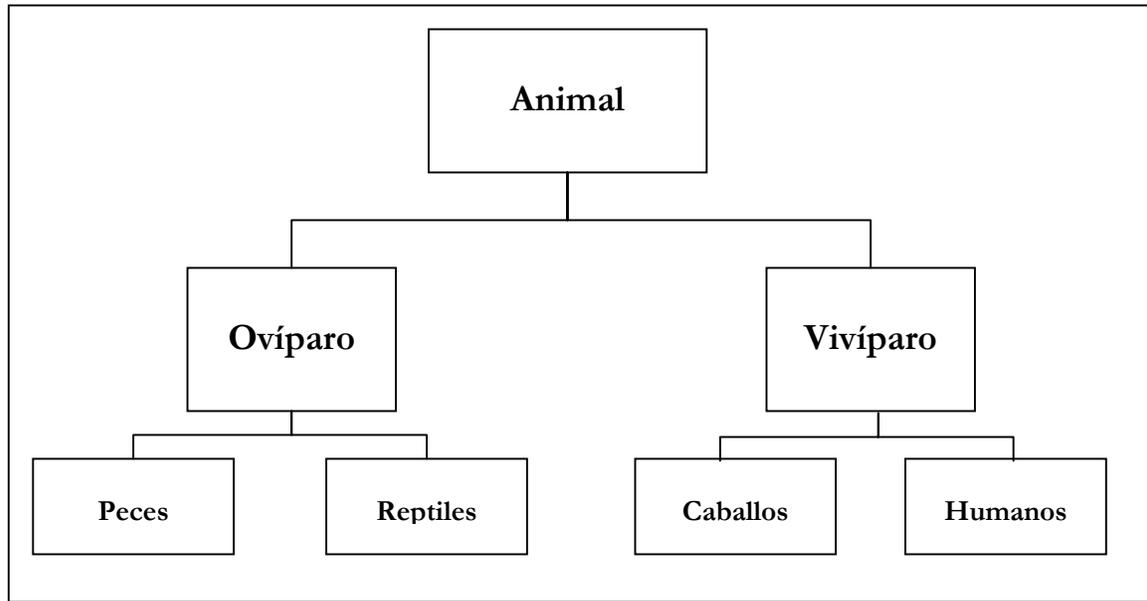


Figura 1.3.4.1 Representación gráfica de la herencia

### 1.3.5.- Mensajes

Un mensaje es una petición de un objeto a otro donde le solicita ejecutar uno de sus métodos. Por convenio el objeto que envía la petición se denomina emisor y el objeto que recibe la petición se denomina receptor.

El conjunto de mensajes a los que responde un objeto se denomina comportamiento del objeto. El mensaje incluye el nombre de una operación y cualquier argumento requerido por esa operación. Con frecuencia es útil referirse a una operación por su nombre sin considerar argumentos.

## 1.4.- Modularidad

Se trata de descomponer un programa en un pequeño número de abstracciones coherentes que pertenecen al dominio del problema y cuya complejidad interna puede ser fácilmente oculta por medio de una interfaz.

En otras palabras, es la posibilidad de subdividir un sistema en piezas más pequeñas llamadas módulos, cada una de las cuales debe ser tan independiente como sea posible considerando el sistema como el todo. Podemos hacer una analogía con un rompecabezas donde el todo está formado por pequeñas piezas

que al acomodarse entre sí dan como resultado la visualización entera de la imagen original antes de fraccionarla.

Con esto al subdividir un sistema grande y complejo en módulos pequeños se reduce la complejidad de cada uno de ellos siendo estos en parte independientes de los restantes y por lo tanto reduciendo también la necesidad de tomar decisiones globales.

#### ***1.4.1.- La estructura de un módulo***

Un módulo se caracteriza fundamentalmente por su interfaz y por su implementación, en donde la primera va a ser la que nos permita comunicarnos con el código interno, será la parte visible hacia el usuario; mientras que la implementación será el código encargado de modificar los atributos y de realizar las conexiones necesarias para acceder a la base de datos.

#### ***1.4.2.- Reglas de modularización***

En primer lugar se debe de recurrir a un método de diseño para poder identificar el alcance del sistema que se va a diseñar y poder separar de él los módulos que sean necesarios para crear el sistema completo, las características de un módulo son las siguientes:

- Debe ser estructurado de tal manera que pueda resolverse de manera casi independiente.
- Debe ser fácil conectar los diferentes módulos.
- Cada módulo tiene un significado específico propio y debe asegurarse que cualquier cambio en su implementación no afecte a su exterior (o al menos que sea un cambio mínimo)
- Se debe de asegurar que los posibles errores o comportamientos erráticos no se propaguen más allá del módulo.

- Cada módulo debe realizar solo una función relacionada con el problema general.

### ***1.4.3.- Diseño de Módulos***

El diseño modular busca la división de un sistema grande en módulos más pequeños y a la vez más manejables, sin embargo esto no es garantía de que el sistema este bien organizado. Por tal razón los módulos deben diseñarse bajo los criterios de acoplamiento y cohesión:

#### ***1.4.3.1.- Acoplamiento***

Éste es una medida del grado de interdependencia entre los módulos es decir, la manera en la que un módulo se ve afectado por otro. Entre menor sea el grado de acoplamiento mejor será, ya que indicará que los módulos son menos dependientes unos de otros. Con esto logramos que en el momento de sustituir un módulo por otro sólo algunos de los restantes serán afectados, además de que es más fácil realizar el seguimiento de un error y por lo tanto aislar el módulo que lo produce.

Sin embargo se debe tener en cuenta que no se pueden realizar módulos de un sistema 100% independientes de los demás debido a que se estaría rompiendo el concepto de sistema (conjunto de elementos que interactúan entre sí para lograr un fin), si los módulos son completamente independientes cada uno de ellos se volvería un sistema pequeño sin interacción con los demás, por lo tanto debemos tener cuidado al crear los módulos de manera que los hagamos los más independientes posible, sin olvidar la interacción que deberá existir entre ellos para que puedan funcionar como un todo.

#### ***1.4.3.2.- Cohesión de módulos.***

Este criterio sugiere que un sistema bien modularizado es aquel en el cual las interfaces de los módulos son claras y simples entonces un módulo que se considera cohesivo es aquel que realiza (idealmente) una sola cosa teniendo poca interacción con los procedimientos que realizan otros módulos del sistema.

Idealmente se debería poder tener módulos altamente cohesivos y débilmente acoplados.

*UML*

Capítulo

2

## 2.- El lenguaje unificado de modelado (UML)

En la Ingeniería es importante el contar con modelos que describan el aspecto y la conducta de "*algo*". Ese "*algo*" puede existir ya sea en un estado de desarrollo o de planeación que es cuando los diseñadores del modelo deben investigar los requerimientos del sistema; a menudo el modelo es dividido en cierto número de módulos cada uno de los cuales describe un aspecto específico del producto o sistema en construcción.

El modelado sirve no solamente para los grandes sistemas aún en aplicaciones de pequeño tamaño lo podemos emplear con grandes beneficios, aunque es completamente cierto que en los sistemas más complejos estos aumentan de manera considerable reduciendo los tiempo de desarrollo en gran medida.

UML es un lenguaje para el modelado de sistemas. Nace en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores entre ellos, el de sus creadores: Grady Booch, autor del método Booch, James Rumbaugh, autor del método OMT e Ivar Jacobson, autor de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero de 1997 y desde entonces ha sido utilizada como un estándar para la creación de modelos.

Entre los beneficios que UML nos provee tenemos:

- Tiempos totales de desarrollo más pequeños y rápidos.
- Modelar sistemas utilizando la orientación a objetos.
- Crear sistemas que luego puedan ser escalados a versiones más complejas.
- Tener un modelo del comportamiento del sistema, que sea entendible por toda la gente involucrada en el proyecto así como personas que posteriormente tengan que modificarlo.

- Mejorar la planeación de sistemas, así como tener una documentación.
- Reutilización de código.

### 2.1.- Funcionalidad de UML

UML permite crear modelos de forma independiente del análisis y el diseño. Estos son utilizados para describir algo y comunicar los resultados de su uso. Además son expresados en un *lenguaje de modelado* que consiste de vistas, diagramas, elementos de modelo y un conjunto de mecanismos generales o reglas que indican cómo utilizar los elementos.

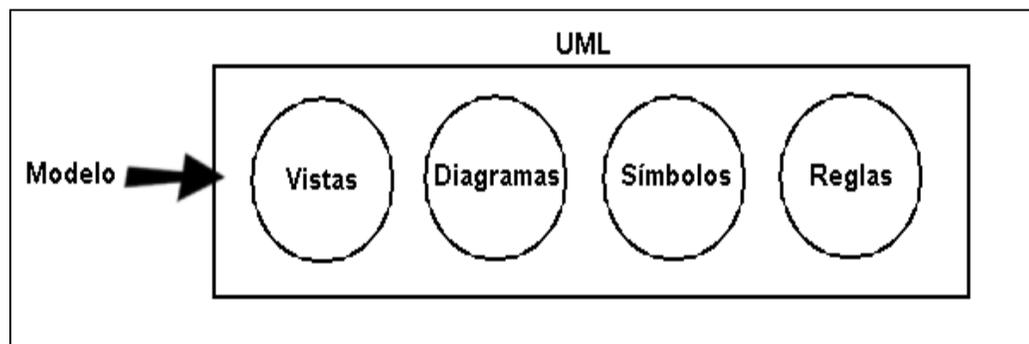


Figura 2.1.1 Representación general de un modelo

#### 2.1.1.- Vistas

Muestran diferentes aspectos del sistema modelado. Es una abstracción que consiste en un número de diagramas y todos ellos en conjunto nos muestran lo que sería el sistema por completo. UML cuenta con diferentes vistas estas son:

- **Caso de uso:** (Use-Case) Muestra la funcionalidad del sistema como la perciben los actores externos.
- **Lógica:** Muestra cómo se diseña la funcionalidad dentro del sistema.

- **De Componentes:** Muestra la organización de los componentes de código.
- **Concurrente:** Muestra la concurrencia en el sistema.
- **De Distribución:** Muestra la distribución del sistema en la arquitectura física con computadoras y dispositivos llamados *nodos*.

## ***2.2.- Descripción de conceptos básicos***

### ***2.2.1.- Diagramas:***

Son las gráficas que describen el contenido de una vista. UML tiene nueve tipos de diagramas que son utilizados para proveer todas las vistas de un sistema: diagramas de caso de uso, de clases, de objetos, de estados, de secuencia, de colaboración, de actividad, de componentes y de distribución.

### ***2.2.2.- Símbolos o Elementos:***

Es todo aquello que se representa en los diagramas: clases, objetos y mensajes, así como las relaciones que existan entre ellos.

### ***2.2.3.- Reglas generales:***

Con ellos podemos incluir comentarios y de esta manera dar información acerca del modelo que lo hagan más comprensible a los desarrolladores.

## ***2.3.- Fases del desarrollo de un sistema***

UML describe cinco fases de desarrollo de sistemas las cuáles son: Análisis de Requerimientos, Análisis, Diseño, Programación y Pruebas.

### ***2.3.1.-Análisis de Requerimientos***

Aquí se utilizan los diagramas de caso de uso para capturar los requerimientos del cliente es decir, qué va a hacer el sistema de acuerdo a su visión; con ellos los actores externos que utilizaran o interactuarán con el sistema son modelados. Estos modelos contienen relaciones y asociaciones entre ellos. Cada diagrama de caso de uso (*use-case*) es descrito en texto y especifica los requerimientos del cliente.

### ***2.3.2.- Análisis***

En esta parte del desarrollo las clases que van a ser modeladas son identificadas con sus relaciones y descritas en un diagrama de clases. Las clases que colaboran entre si para poder ejecutar los casos de uso también se consideran en esta fase a través de los modelos dinámicos en UML. En esta parte de análisis sólo se consideran aquellas clases que podemos identificar del análisis de requerimientos, no aquellas que surgirán durante el desarrollo del sistema.

### ***2.3.3.- Diseño***

Aquí el resultado del análisis es traducido a una solución técnica. Se agregan las clases que proveerán las conexiones a la base de datos así como las de almacenamiento de información.

### ***2.3.4.- Programación***

En esta fase las clases del diseño son convertidas a código en un lenguaje de programación orientado a objetos. Cuando se crean los modelos de análisis y diseño en UML, lo más aconsejable es trasladarlos mentalmente a código.

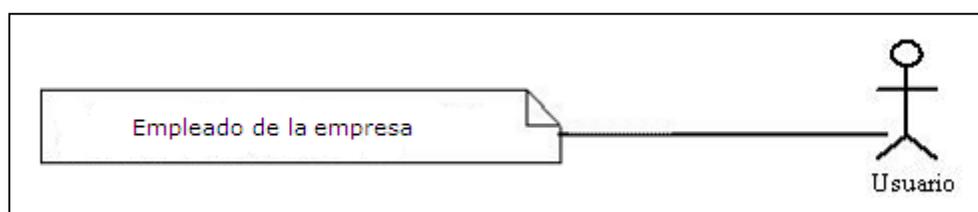
### ***2.3.5.-Pruebas***

Normalmente un sistema es tratado en pruebas de unidades, de integración, de sistema, de aceptación, etc. Las de unidades se realizan a clases individuales o a un grupo de ellas y son típicamente ejecutadas por el programador. Las de integración como su nombre lo dice integran componentes y clases en orden para verificar que se ejecutan como se especificó. Las pruebas de sistema ven al sistema como una "caja negra" y validan que este tenga la funcionalidad que el usuario final espera. Las pruebas de aceptación conducidas por el cliente verifican que el sistema satisface los requerimientos, son similares a las pruebas de sistema.

### ***2.4.- Elementos comunes a todos los diagramas***

#### ***2.4.1.- Notas:***

Sirven para añadir cualquier tipo de comentario a un diagrama o a un elemento del mismo. Es un modo de indicar información en un formato libre cuando la notación del diagrama no nos permite expresarla de manera adecuada. Se representa como un rectángulo con una esquina doblada con texto en su interior. Puede aparecer solamente unido al elemento que esta explicando por una línea discontinua.



*Figura 2.4.1.1 Notación gráfica de UML para las notas*

### 2.4.2.-Dependencias:

Significa que un cambio en el elemento destino puede implicar un cambio en el elemento origen por lo tanto si este cambia se debe de revisar el origen. Se representa por medio de una línea de trazo discontinuo entre los elementos con una flecha que apunta al elemento destino.

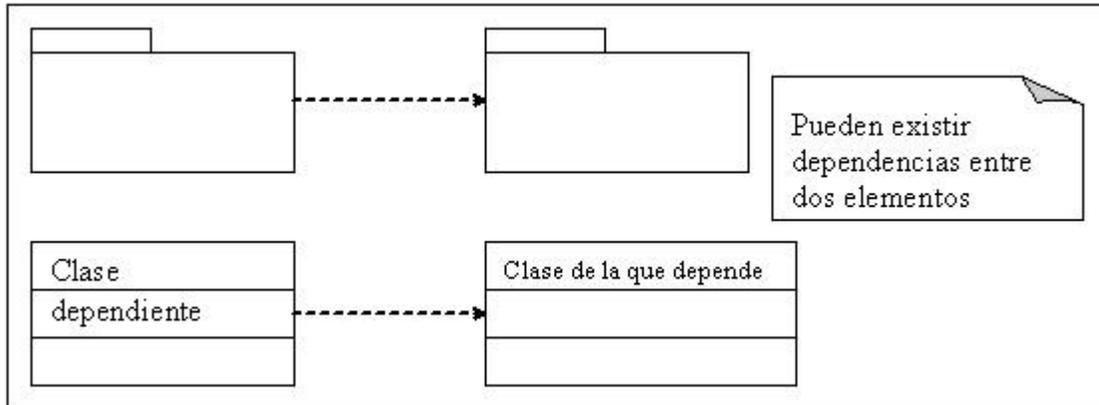


Figura 2.4.2.1 Notación gráfica de UML para las dependencias

### 2.4.3.- Clases:

Se representa mediante una caja subdividida en tres partes. En la superior muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones, o bien se pueden suprimir las divisiones de los atributos y operaciones dejando solamente el nombre de la clase.

### 2.4.4.- Objetos:

Se representan igual que las clases: en la división superior se pone el nombre del objeto junto con el de la clase a la cuál pertenece estos dos subrayados siguiendo la siguiente sintaxis: nombre\_objeto: nombre\_clase.

### 2.4.5.- Asociaciones:

Se representan mediante una línea continua que las une esta puede tener una serie de elementos gráficos que expresan características particulares de la asociación.

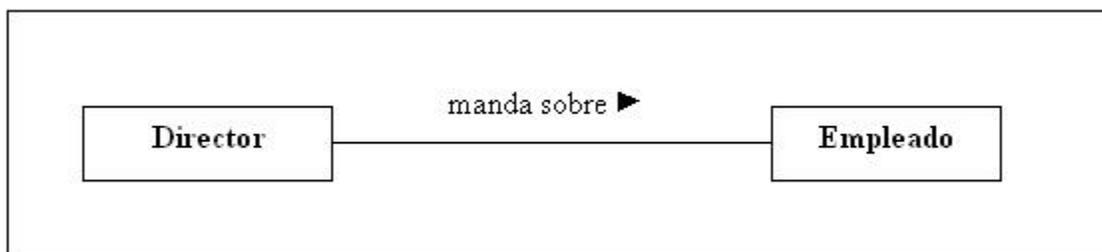


Figura 2.4.5.1 Notación gráfica de UML para las asociaciones

El nombre de la asociación es opcional y se muestra como un texto próximo a la línea. Se puede añadir un pequeño triángulo negro que indique la dirección en la cuál se leerá el nombre de la asociación, por lo general son incluidos en los modelos para aumentar la legibilidad; cuándo la información es muy abundante se pueden suprimir los nombres para evitar que el modelo sea muy grande en cuanto a la información que presenta.

### 2.4.6.- Multiplicidad:

Es una restricción que se le coloca a una asociación para limitar el número de instancias de una clase que pueden tener asociación con una instancia de otra clase. Se puede expresar con un número fijo por ejemplo: 2, con un intervalo de valores: 2:5, con un rango en el cual uno de los extremos es un asterisco que significa un intervalo abierto: 2:\* o bien con una serie de elementos como los anteriores: 2,3,5:7,\*.

### 2.4.7.-Roles:

Es el papel que juega una clase en una asociación; se puede especificar un nombre de rol. Se representa en el extremo de la asociación junto a la clase que desempeña dicho rol.

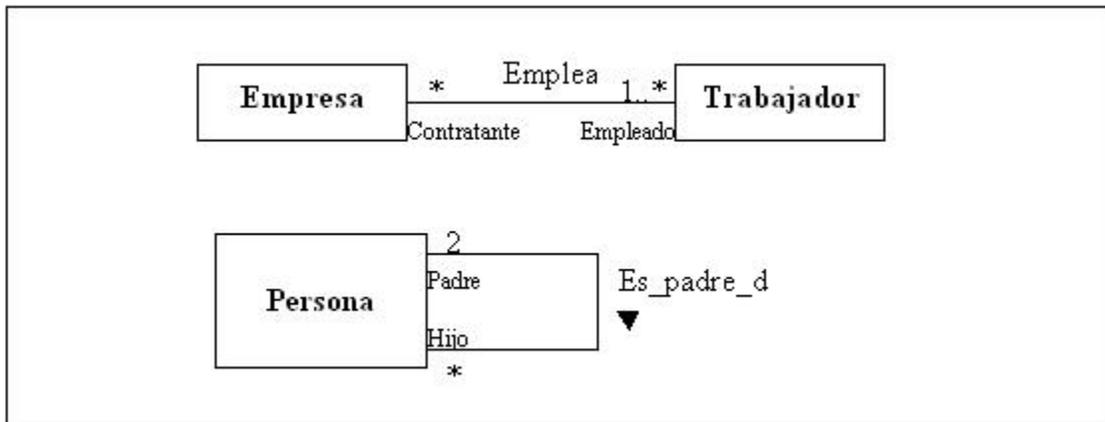


Figura 2.4.7.1 Notación gráfica de UML para los roles

### 2.4.8.-Agregación:

Se representa como un diamante junto a la clase que representa el “todo”

### 2.4.9.-Asociaciones N-Arias

En el caso de una asociación en la que participan más de dos clases estas se unen con una línea a un diamante central.

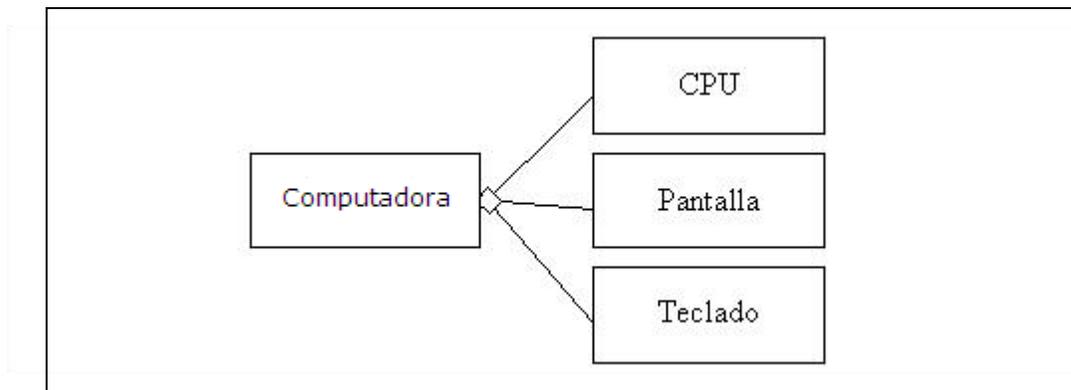


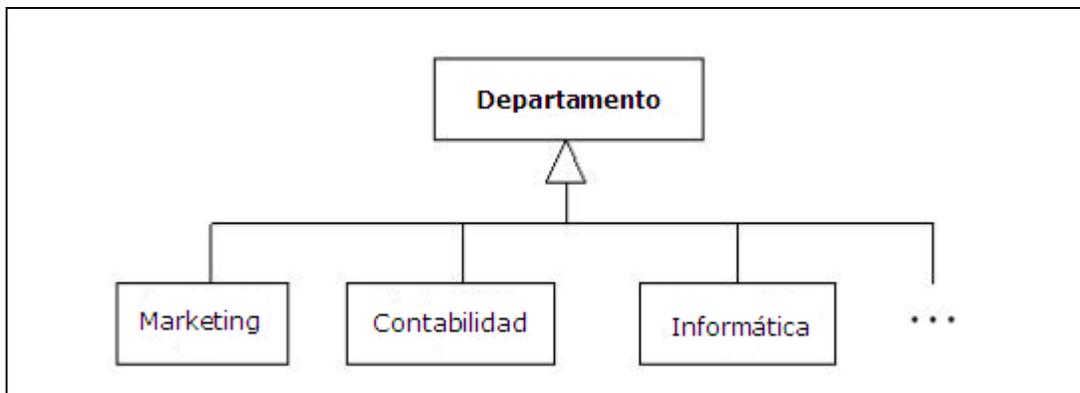
Figura 2.4.9.1 Notación gráfica de UML para las asociaciones N-Arias

### ***2.4.10.- Navegabilidad:***

En un extremo de la asociación se puede indicar la navegabilidad, esto es que es posible navegar o ir desde el objeto de la clase origen hasta el de la clase destino indicando que el objeto de la clase origen conoce a los objetos de la clase destino y por lo tanto puede llamar a alguna de sus operaciones.

### ***2.4.11.-Herencia:***

Se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre”, si una clase tiene varias clases hijas pero no son necesarias dentro del diagrama se indican con puntos suspensivos.



*Figura 2.4.11.1 Notación gráfica de UML para la Herencia*

### ***2.4.12.-Elementos Derivados:***

Es aquel cuyo valor se puede calcular a partir de otros elementos presentes en el modelo pero que se incluye por motivos de claridad o como decisión del diseño. Se representa con una barra diagonal (“/”) precediendo al nombre del elemento derivado.



Figura 2.4.12.1 Notación gráfica de UML para los elementos derivados

## 2.5.-Diagramas dentro de UML

### 2.5.1.- Diagrama de Casos de Uso (Use- Case )

Muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Los elementos de un diagrama de casos de uso son:

#### 2.5.1.1.- Sistema

Es descrito como una caja; el nombre del sistema aparece arriba o dentro de ella. Ésta también contiene los símbolos para los casos de uso del sistema.

#### 2.5.1.2.- Actores

Un actor es alguien o algo que interactúa con el sistema; es quien lo utiliza. Por la frase "interactúa con el sistema" se debe entender que el actor envía al sistema mensajes o bien recibe información del mismo. En pocas palabras el actor lleva a cabo los casos de uso. Puede ser una persona u otro sistema que se comunica con el que se está modelando.

Un actor es un tipo (es decir, una clase) no una instancia e implica un rol, gráficamente se representa con la figura de "*stickman*".

#### 2.5.1.2.1.- Encontrando a los actores de un diagrama de casos de uso.

Es posible obtener a los actores de un diagrama de casos de uso a través de las siguientes preguntas:

- ✚ ¿Quién utilizará la funcionalidad principal del sistema (actores primarios)?
- ✚ ¿Quién necesitará soporte del sistema para realizar sus actividades diarias?
- ✚ ¿Quién necesitará mantener, administrar y trabajar el sistema?
- ✚ ¿Qué dispositivos de hardware necesitará manejar el sistema?
- ✚ ¿Con qué otros sistemas necesitarán interactuar el que se va a desarrollar?
- ✚ ¿Quién o qué tiene interés en los resultados que el sistema producirá?

#### 2.5.1.3.- Casos de Uso

Un caso de uso representa la funcionalidad completa tal y como la percibe un actor. En UML es definido como un conjunto de secuencias de acciones que un sistema ejecuta y que permite un resultado observable de valores para un actor en particular. Gráficamente se representan con una elipse y tiene las siguientes características:

- Un caso de uso siempre es iniciado por un actor.
- Un caso de uso provee valores a un actor.
- Un caso de uso es completo.

### 2.5.1.3.1.- Encontrando casos de uso

El proceso para encontrar casos de uso inicia encontrando al actor o actores previamente definidos. Por cada actor identificado hay que realizar las siguientes preguntas:

- ✚ ¿Qué funciones del sistema requiere el actor? ¿Qué necesita hacer el actor?
- ✚ ¿El actor necesita leer, crear, destruir, modificar o almacenar algún tipo de información en el sistema?
- ✚ ¿El actor debe ser notificado de eventos en el sistema o viceversa?
- ✚ ¿Qué representan esos eventos en términos de funcionalidad?
- ✚ ¿El trabajo diario del actor podría ser simplificado o hecho más eficientemente a través de nuevas funciones en el sistema? (Comúnmente, acciones actuales del actor que no estén automatizadas)

Otras preguntas que nos ayudan a encontrar casos de uso pero que no involucran actores son:

- ✚ ¿Qué entradas/salidas necesita el sistema? ¿De dónde vienen esas entradas o hacia dónde van las salidas?
- ✚ ¿Cuáles son los mayores problemas de la implementación actual del sistema?

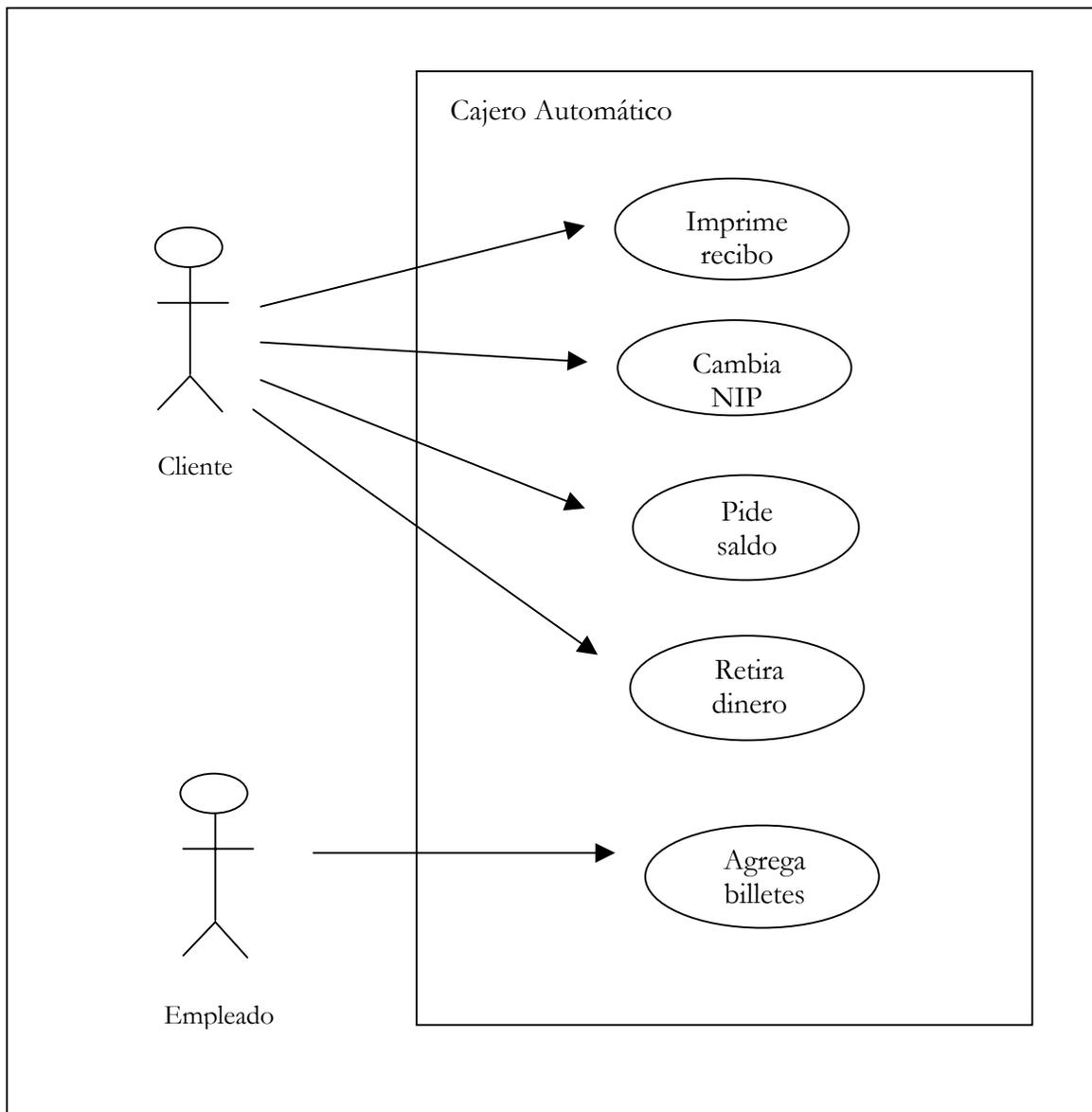


Figura 2.5.1.3.1.1 Diagrama de caso de uso para un cajero automático.

### ***2.5.1.4.- Diagramas de Interacción:***

Ellos muestran un patrón de interacción entre objetos. Hay dos tipos de diagramas, ambos basados en la misma información pero cada uno enfatiza un aspecto particular:

#### *2.5.1.4.1.- Diagramas de secuencia*

Muestra la interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, mientras que en el horizontal se colocan los objetos y actores participantes en la interacción sin un orden prefijado. Cada objeto o actor tiene una línea vertical y los mensajes se representan mediante una flecha entre los distintos objetos. El tiempo fluye de arriba abajo, se pueden colocar etiquetas (como restricciones del tiempo, descripción de las acciones) junto a las transiciones a las que se refieren.

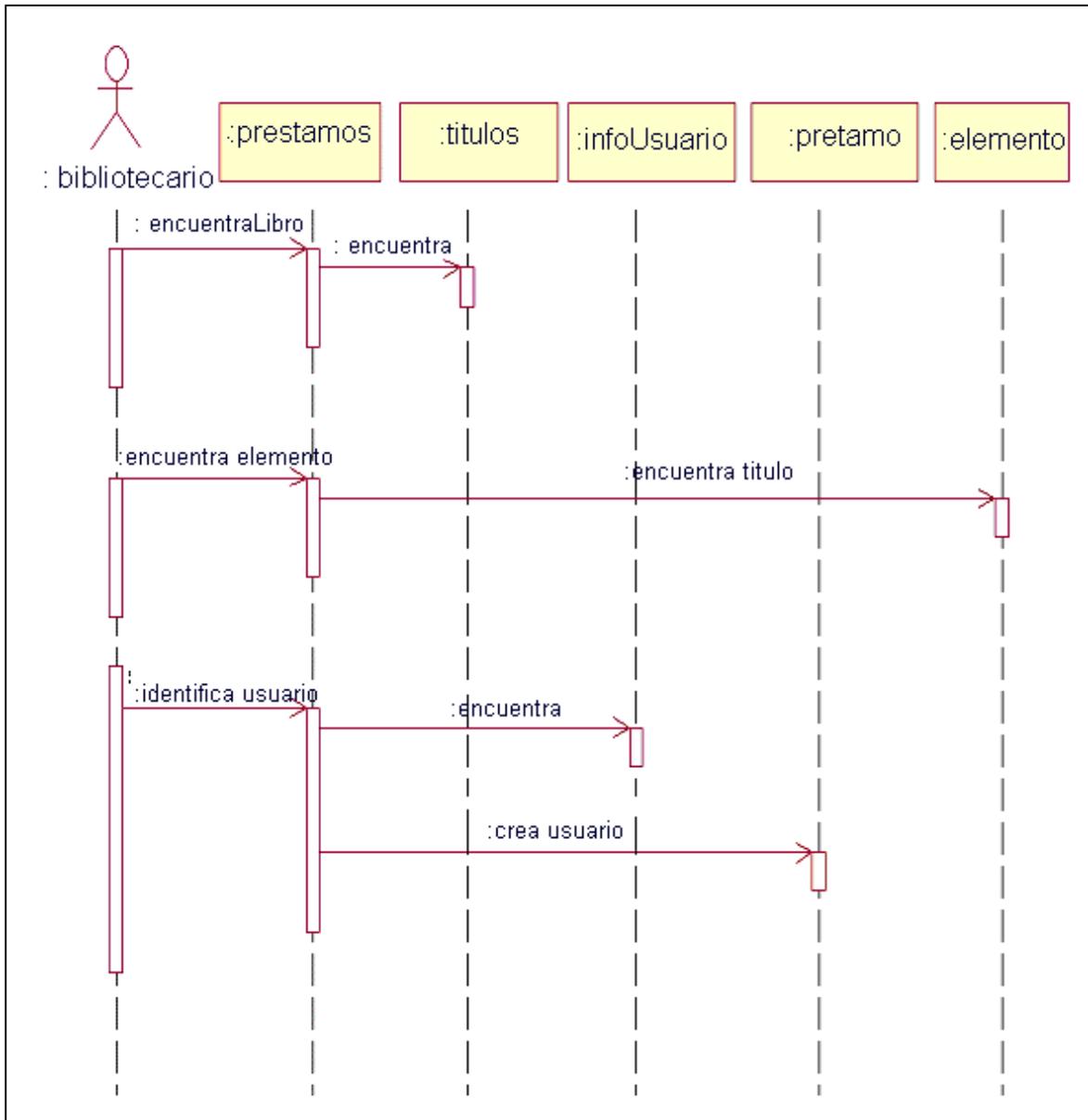


Figura 2.5.1.4.1.1 Representación de los diagramas de secuencia

#### 2.5.1.4.2.- Diagramas de colaboración

Muestra una interacción organizada basándose en los objetos que toman parte en ella y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los diagramas de secuencia, estos muestran las relaciones entre los roles de los objetos. Las secuencias de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

En su representación muestra a una serie de objetos con los enlaces entre los mismos y con los mensajes que se intercambian. Los mensajes son flechas que van junto al enlace por el que “circulan” que contienen el nombre del mensaje y los parámetros (si existen) entre paréntesis. Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el que inicia el diagrama que no lleva número de secuencia.

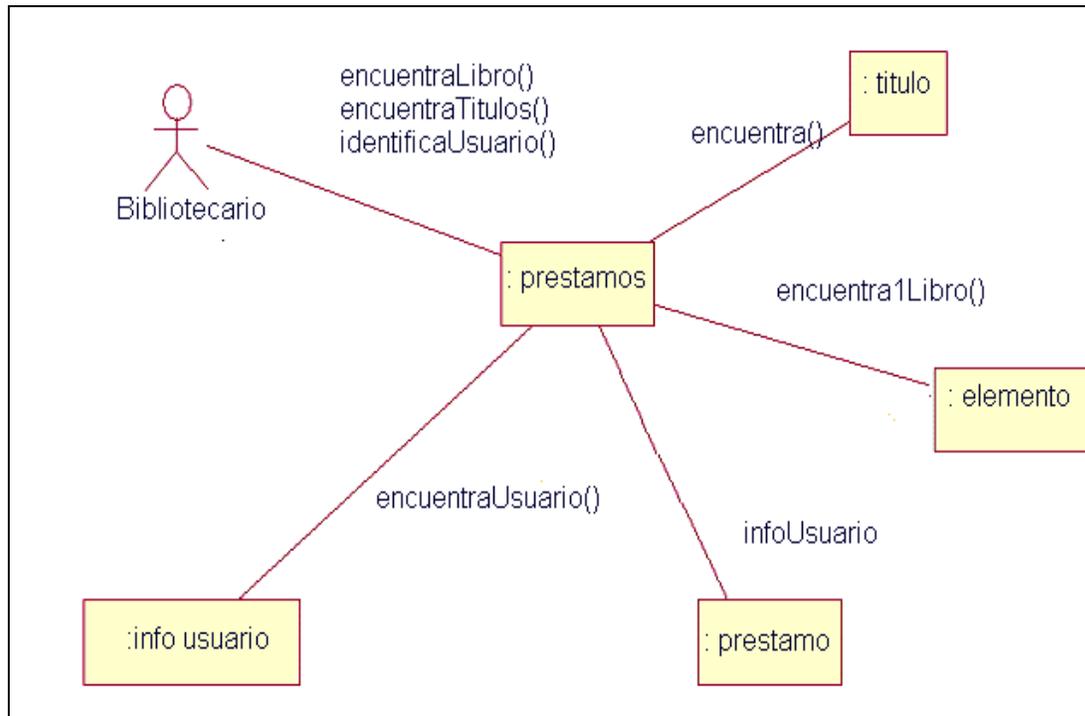


Figura 2.5.1.4.2.1 Representación de los diagramas de colaboración

### 2.5.1.5.- Diagramas de Clases:

Para este diagrama se toman como base los de secuencia y de colaboración por lo que se manejarán los objetos que ahí se consideraron pero ahora a nivel de clases. Además se pueden agregar nuevas clases que no existían. Este diagrama es un tipo de modelo estático describiendo el sistema en términos de clases y relaciones.

El diagrama de clases muestra sólo las clases, sin embargo puede existir una variante de este diagrama que muestra las instancias, es decir el diagrama de objetos.

Para poder definir las clases UML sugiere seis características selectivas que debe utilizar el analista para considerar una clase candidato en el modelo de análisis:

- *Información almacenable*: La clase será útil durante el análisis sólo si la información sobre el mismo ha de ser almacenada, transformada, analizada o manejada de algún modo. La información puede referirse a conceptos que deberán estar siempre registrados en el sistema, eventos o transacciones que ocurren en un momento específico.
- *Sistema externo*. Si se tiene un sistema externo a este, entonces es de interés en la etapa de modelado. Los sistemas externos deberán ser vistos como clases que el sistema contendrá o con los cuales interactuará.
- *Patrones, librerías de clases o componentes*. Si se tienen patrones, librerías de clases o componentes de proyectos anteriores, generalmente éstos son clases candidatos.
- *Dispositivos que el sistema maneja*. Dispositivos técnicos que maneja el sistema se convertirán en clases que manejarán esos dispositivos.
- *Partes organizacionales*. Especialmente en modelos de negocio, todas las partes que representan a la organización serán clases candidatos.
- *Roles de actores*. Los roles de actores serán vistos como clases por ejemplo: usuario, operador del sistema, administrador, cliente, etc.

En este diagrama las clases se representan en una caja rectangular dividida en 3 compartimentos:

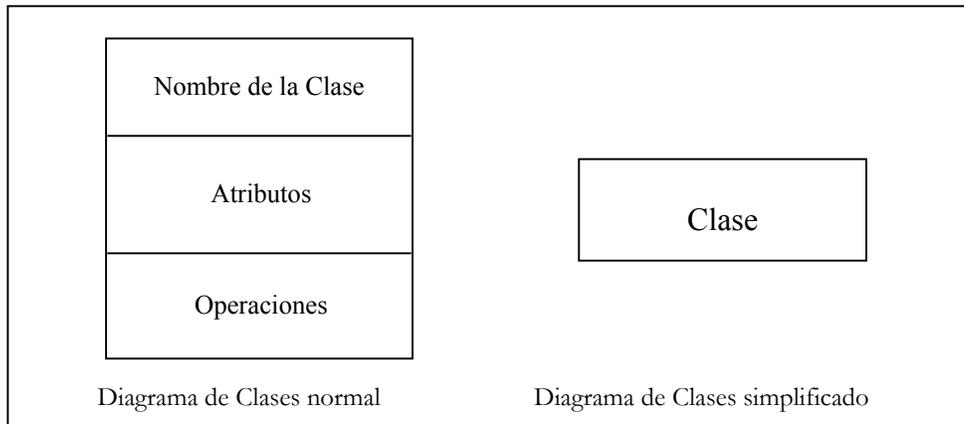
1. En la parte superior se coloca el nombre de la clase el cual, por lo general va en negritas y centrado es conveniente que el nombre se deduzca del dominio del problema y sea lo menos ambiguo posible, no debe tener prefijos o sufijos.

2. En la parte del medio se encuentran los atributos que describen las características de los objetos. Los atributos correctos capturan la información que describe e identifica a una instancia de la clase sólo deben incluirse los atributos que sean de interés dentro del sistema a modelar. Un atributo tiene un tipo, que por lo general son los tipos primitivos de datos dentro de un lenguaje de programación: *boolean*, entero, flotante, cadena, etc. y también tienen una visibilidad, pudiendo ser:

- a) públicos: Se pueden utilizar y visualizar fuera de la clase ( signo + adelante del nombre ).
- b) Privados: No se pueden acceder desde otras clases ( signo – adelante del nombre ).
- c) Protegidos: Son atributos que no se pueden modificar, se utilizan en notaciones de generalización y especialización.

Si ningún signo está indicado, quiere decir que la visibilidad no está definida. Aquellos atributos que pertenezcan a la clase y que por ende sean comunes a todos los objetos deberán ir subrayados para mejor identificación.

3. En la inferior se encuentran las operaciones o métodos de la clase que son los utilizados para manipular los atributos (modificar el estado de la clase) o realizar alguna otra acción, son funciones a las que se llama o invoca siendo interiores a la clase y se pueden aplicar sobre objetos de esa clase. Al igual que los atributos, los métodos pueden tener cierta visibilidad y en ellos opera de la misma forma que para los atributos.



*Figura 2.5.1.5.1 Representación de las clases*

En un diagrama simplificado de clases la segunda y tercer banda o compartimiento son omitidos mostrando por lo tanto sólo el nombre de la clase.

Las clases se relacionan entre sí a través de las relaciones que son las líneas rectas entre dos de ellas y constan de los roles, estos son las frases que se encuentran en la relación.

Tanto los atributos como los métodos cuentan con una sintaxis. Para los atributos es la siguiente:

*visibilidad nombre : tipo = valor\_inicial {lista\_de\_posibles\_valores}*

*por ejemplo:*

+ *cantidad: entero (publico de tipo entero)*  
 - *cantidad: entero (privado de tipo entero)*

para los métodos, la sintaxis es la siguiente:

*visibilidad nombre (lista\_parámetros) : tipo\_de\_retorno {lista\_de\_posibles\_valores}*

*por ejemplo:*

+ aterrizar ( velocidad: entero)  
dibujar ( )

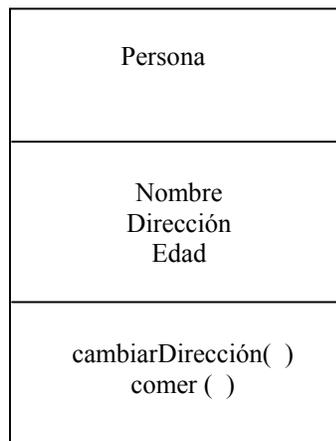
Donde la lista de parámetros consta de la siguiente sintaxis:

*nombre : tipo = valor\_default*

*por ejemplo:*

*x: integer = 3*

Y los parámetros van separados por comas.



*Figura 2.5.1.5.2 Representación de las clases simplificadas*

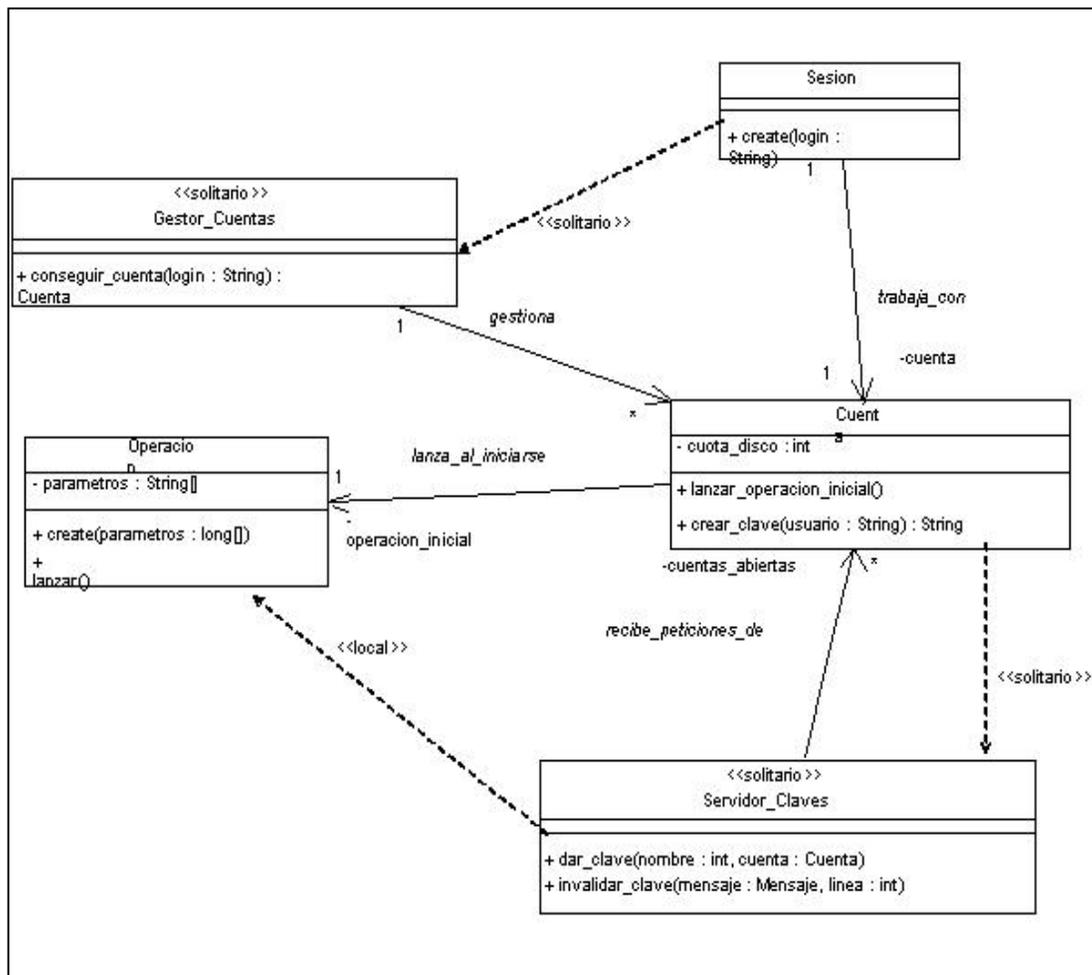


Figura 2.5.1.5.3 Representación de las clases

### 2.5.1.6.- Relaciones en los Diagramas de Clase

Dentro de un diagrama de clases se tiene una parte muy importante: las relaciones entre ellas. En UML se tienen las siguientes relaciones:

- **Asociación:** Es una conexión entre clases; es bidireccional lo cual indica que los objetos se conocen entre sí perteneciendo a clases diferentes, estas se representan mediante una línea que une a las dos clases, el nombre de la asociación se escribe sobre ella:

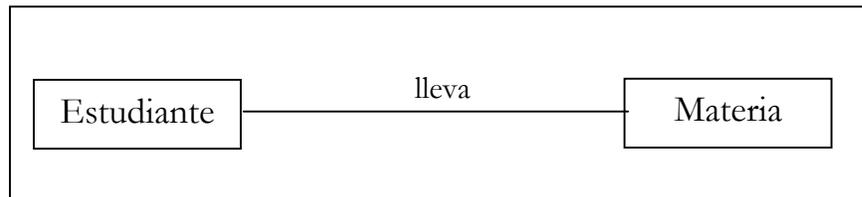


Figura 2.5.1.6.1 Relación de asociación

Cada asociación es una dependencia bidireccional a menos que se especifique lo contrario.

- **Generalización:** Es el relacionar un elemento muy general a uno más específico en donde este pudiera sólo contener información adicional. Esta relación implica que existen clases que son muy generales y que de ellas se derivan clases que son más específicas. Por lo general en un lenguaje de programación orientado a objetos la generalización se conoce como herencia y se representa por una punta de flecha y una línea que se dirige de la clase especializada a la general. Al leer la relación de generalización tendríamos que es una relación es/un, por ejemplo el perro ES / UN animal:

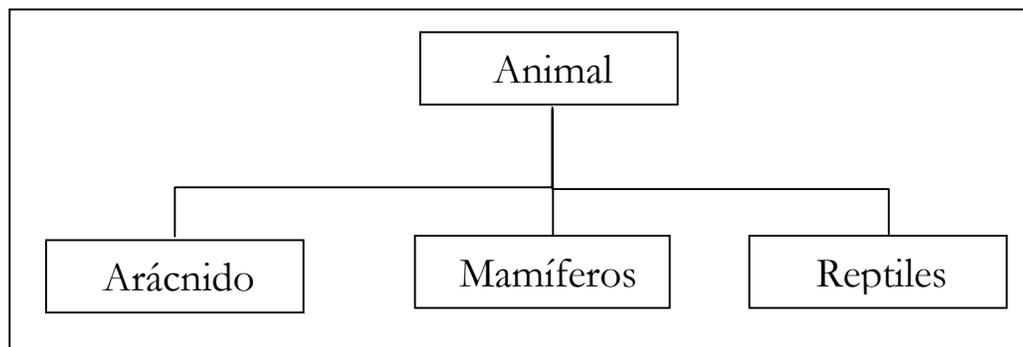


Figura 2.5.1.6.2 Relación de generalización

- **Agregación:** Es una relación que representa a los objetos compuestos; como ya se mencionó en el primer capítulo un objeto compuesto es aquel que se compone de otros objetos. La agregación permite describir modelos del mundo real que se

componen a su vez de otros modelos y así sucesivamente alcanzando el nivel de profundidad que uno desee o requiera. Por lo general al leer una relación de agregación tendríamos una relación de tiene / un, por ejemplo: una casa TIENE / UN techo, TIENE / habitaciones, etc.

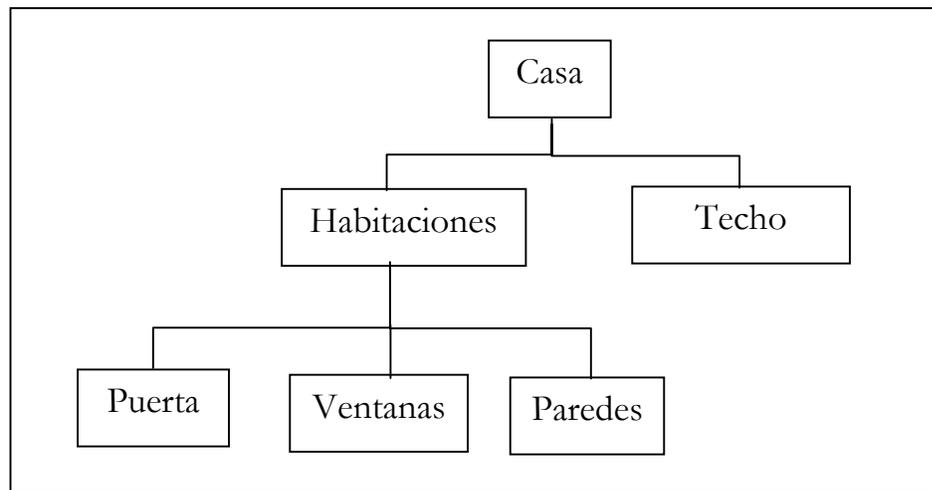


Figura 2.5.1.6.4 Relación de agregación

- **Dependencia:** Es cuando tenemos una relación entre dos clases, una siendo dependiente de la otra lo que implica que cualquier cambio en el elemento independiente afectará de alguna manera al dependiente más no viceversa.

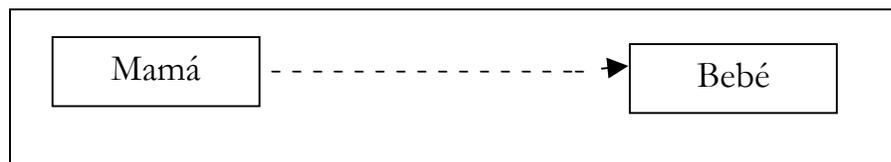


Figura 2.5.1.6.5 Relación de dependencia

- **Refinamiento:** Es la relación entre dos descripciones de la misma clase pero en niveles diferentes de abstracción.

### 2.5.1.7.- Diagrama de Estados:

Posteriormente se realiza el *diagrama de estados* el cual captura el ciclo de vida de los objetos, subsistemas y sistemas. Dicho diagrama determina los estados que un objeto puede tener y cómo los eventos llegan a afectar a los estados a través del tiempo. Dicho de otro modo, muestra los estados por los que pasa un elemento (caso de uso, objeto o todo el sistema) a lo largo del tiempo, en el se indica que eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que se generan.

Todos los objetos pasan a un estado siendo éste el resultado de actividades previas ejecutadas por el objeto. Ese estado está determinado por los valores de los atributos de este objeto y sus relaciones con otros objetos. Una clase puede tener un atributo que especifique el estado.

Gráficamente, los estados se representan en rectángulos con esquinas redondeadas, las líneas entre dos estados se llaman *transiciones*. Las transiciones constan de una sintaxis la cual es la siguiente:

$$\textit{nombre\_evento (parámetros) [ condición ] / acción\_o\_consecuencia}$$

(los parámetros van separados por comas). La condición es una expresión que tiene que considerarse para que el evento se genere, la acción o consecuencia es una expresión que se debe efectuar después del evento y puede ser un incremento o un decremento.

En el interior del rectángulo se coloca el nombre del estado, las transiciones se representan como una flecha que va desde el estado origen al destino, la caja puede estar dividida en dos en la parte de arriba aparece el nombre del estado y en la segunda que es opcional pueden aparecer acciones de entrada, de salida o acciones internas.

Una entrada sigue la sintaxis: *entrada/acción\_asociada*; donde *acción\_asociada* es el nombre de la acción que se realiza al entrar a ese estado

. Una salida aparece como: salida/acción\_salida, donde esta última será la acción que se ejecutará cuando se salga de ese estado.

La acción interna es aquella que se ejecuta cuando se recibe un determinado evento o mensaje estando en ese estado, pero su ejecución no causa transición.

El diagrama de estados puede representar ciclos continuos o una vida finita en la que hay un estado inicial (de creación), estados intermedios (pueden o no existir) y un estado final (de destrucción). El estado inicial se representa como un círculo y el final como dos círculos uno dentro del otro.

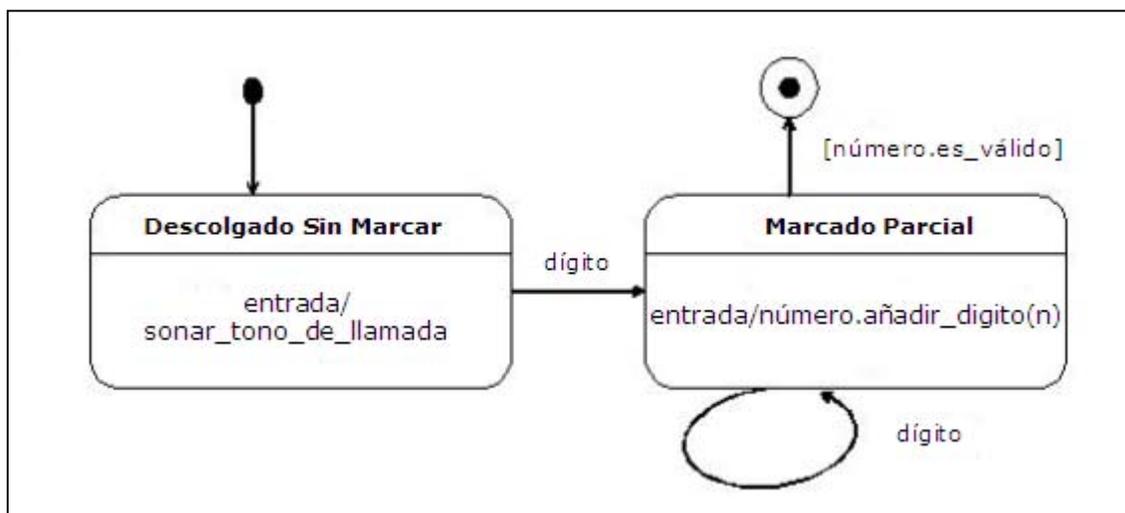


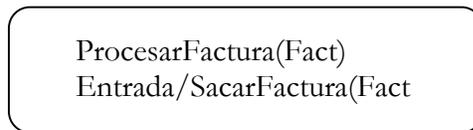
Figura 2.5.1.7.1 Diagrama de estados.

### 2.5.1.8.- Diagrama de actividades

El diagrama de actividades de UML lo podemos comparar en su estructura con los diagramas de flujos utilizados en programación. Nos va indicando paso a paso la secuencia que siguen los eventos para llegar a un resultado final. Este diagrama se centra principalmente como su nombre lo indica en las actividades que se deben llevar a cabo.

El contenido de un diagrama de actividades básicamente es:

- *Estados de actividad:* Se representan por un rectángulo redondeado un tanto angosto y ovalado en cuyo centro se especifica la actividad o la acción, se considera que un estado de actividad es instantáneo y no puede ser interrumpido.



*Figura 2.5.1.8.1 Representación de un estado actividad*

- *Estados de acción:* Estos si pueden descomponerse en más sub-actividades representadas a través de otros diagramas, si pueden ser interrumpidos y tardan un cierto tiempo en completarse. Aquí podemos tener actividades como entradas y salidas del estado en cuestión.
- *Transiciones:* Reflejan el paso de un estado a otro o de una actividad a otra, se representa con una flecha que va de la actividad que lo propicia a la que le sigue.

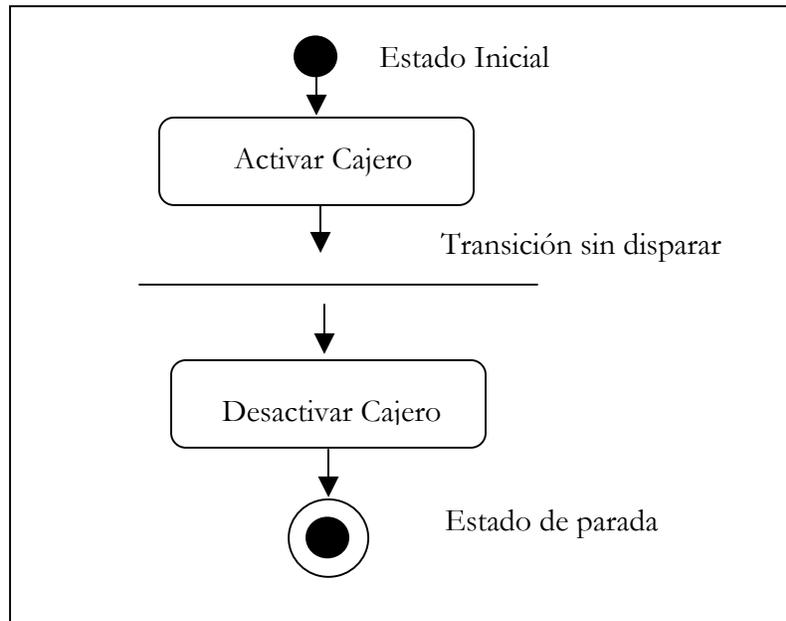
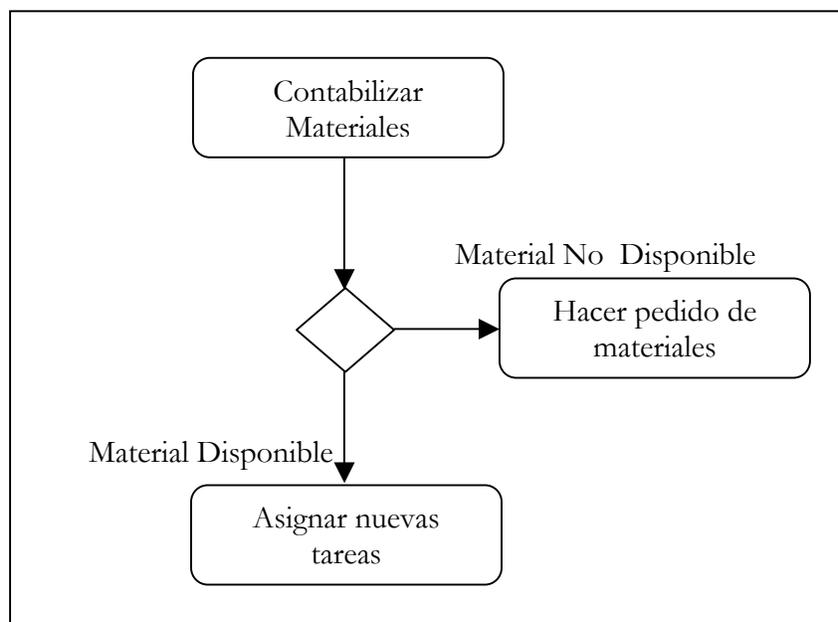


Figura 2.5.1.8.2 Representación de una transición.

- *Bifurcaciones:* Representan caminos alternativos que se pueden seguir al momento de efectuar una transición antes de pasar por una bifurcación se debe de evaluar una expresión *booleana* (verdadera o falsa) para poder saber que camino tomar.



*Figura 2.5.1.8.3 Representación de una bifurcación*

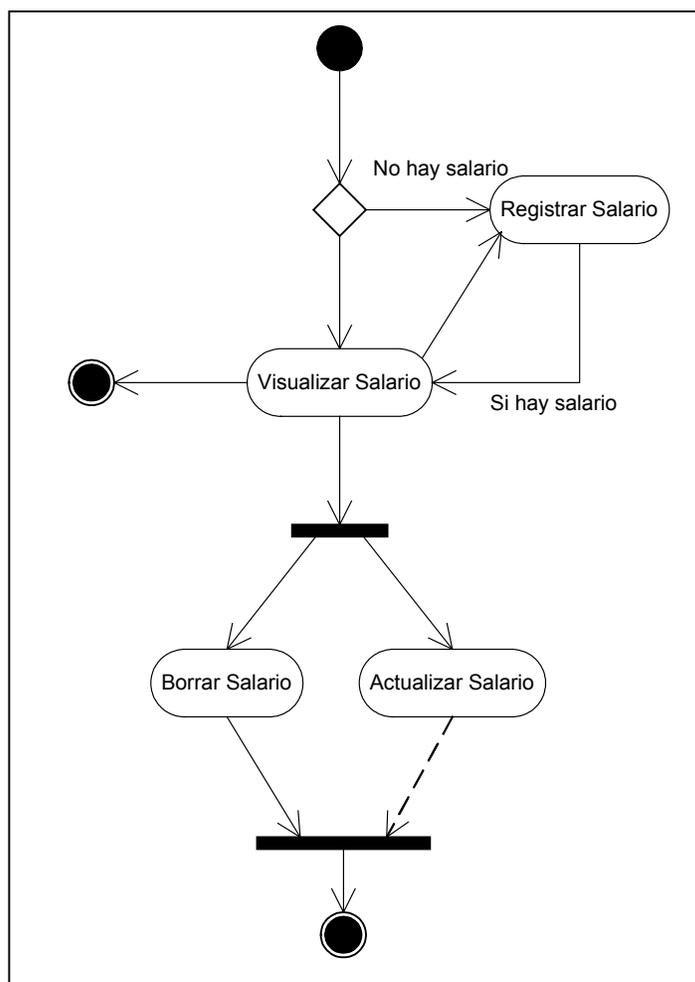
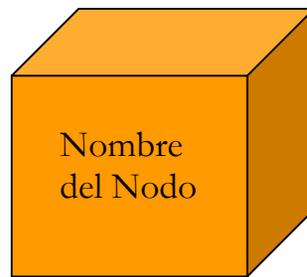


Figura 2.5.1.8.4 Ejemplo de un diagrama de actividades con bifurcaciones, estados y transiciones.

### 2.5.1.9.- Diagramas de Despliegue y de Componentes

Nodo: Es un elemento físico que pertenece al mundo real y que existe en tiempo de ejecución, representa un recurso computacional que generalmente tiene alguna memoria y capacidad de procesamiento. Los nodos sirven para ayudarnos a modelar la topología del hardware sobre el cuál se ejecuta el sistema. El nodo debe de tener un nombre asignado que lo distinga del resto de ellos se representan gráficamente como un cubo:



*Figura 2.5.1.9.1 Representación gráfica de un nodo*

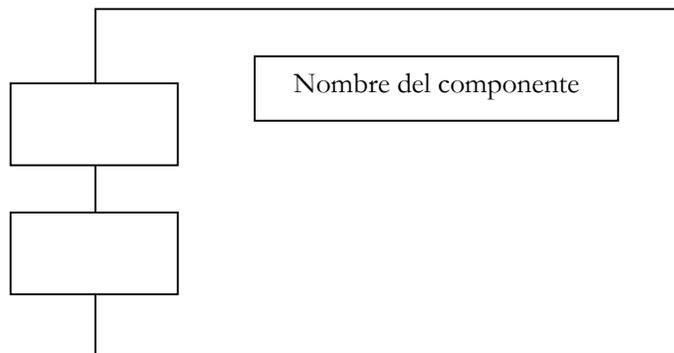
### 2.5.1.10.- Diagrama de Componentes

Describe componentes de software y las dependencias entre ellos, representando la estructura del código. Los componentes de software pueden ser: de código, binarios (que son los generados por la compilación de los anteriores) y los ejecutables.

Este diagrama muestra un conjunto de componentes y sus relaciones de manera gráfica a través del uso de nodos y de arcos por lo general contiene:

- Componentes.
- Interfaces.
- Relaciones de dependencia, generalización y asociaciones.

- Paquetes o subsistemas.
- Instancias de alguna clase.

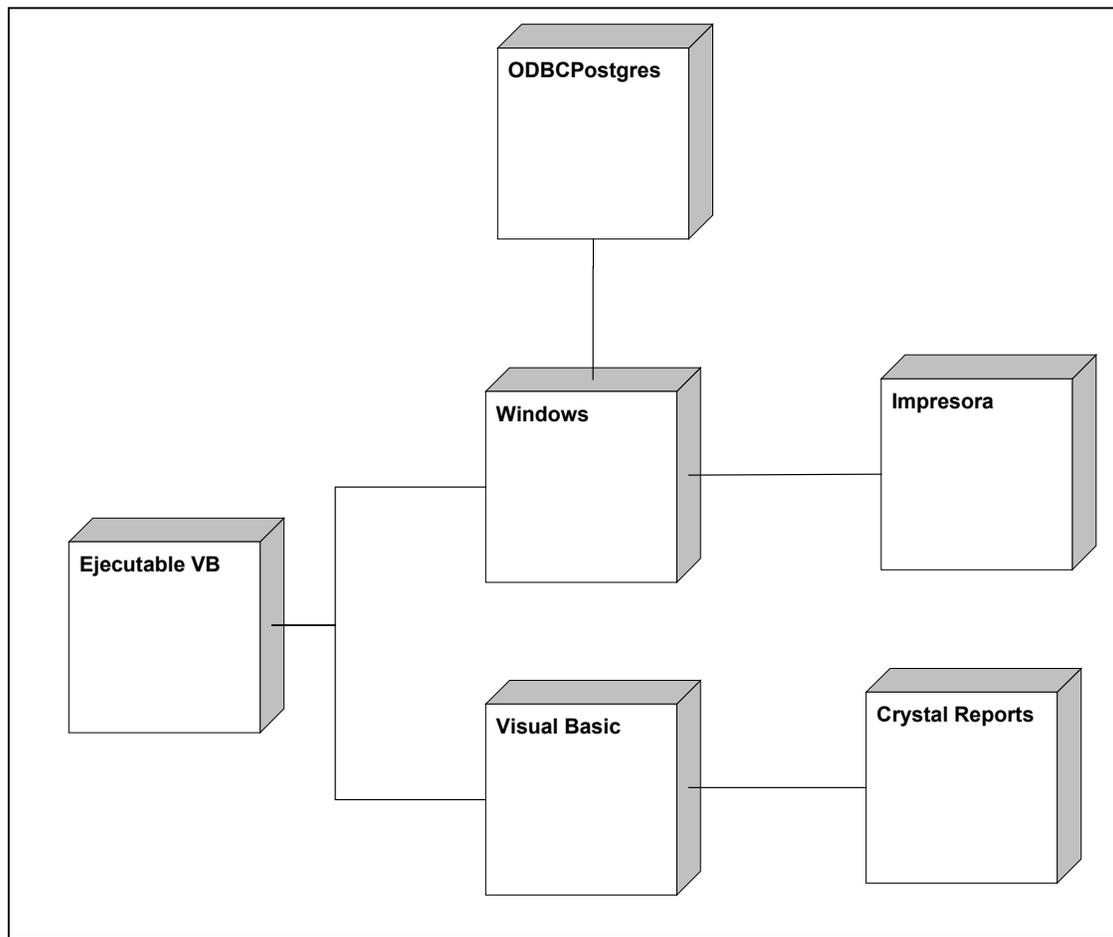


*Figura 2.5.1.10.1 Representación gráfica de un componente*

#### **2.5.1.11.- Diagrama de despliegue o de distribución.**

Contiene los nodos y las conexiones que muestran la arquitectura del sistema en tiempo de ejecución a través de procesadores, dispositivos y los componentes de software. Esta es la última descripción física de la topología del sistema describiendo la estructura de las unidades de hardware y el software que se ejecuta en cada unidad.

Si nodo representa a una instancia en lugar de a una clase, el nombre va subrayado. Las conexiones se representan con líneas continuas y contienen el nombre y el estereotipo de la conexión. El nombre es el identificador de la misma y el estereotipo indica el protocolo de comunicaciones entre los dos nodos implicados.



*Figura 2.5.1.11.1 Diagrama de despliegue con más de un nodo.*

*Requerimientos del  
sistema*

---

---

Capítulo

3

## **3.- Especificaciones del Sistema de Evaluación de Becarios (SICEB)**

### ***3.1.- Antecedentes***

La Unidad de Servicios de Cómputo Académico entre los muchos servicios que ofrece se encuentra el de Plan de Becarios en donde alumnos de la Facultad son escogidos luego de un examen general para tomar cursos intensivos de programación durante un semestre, luego del cual deben presentar una exposición y posteriormente, si cumplieron con el perfil aspirado por la Unidad durante los cursos y los aprobaron satisfactoriamente, son incorporados al equipo de Becarios con el que cuenta la Unidad para formar parte de uno de los 4 departamentos, desarrollando diferentes actividades de acuerdo a donde sean asignados.

Al pasar a formar parte del equipo de becarios de UNICA adquieren ciertas responsabilidades las cuáles deben de ser evaluadas por su jefe inmediato para analizar el desempeño que tuvieron; estas evaluaciones se realizan justo al inicio de un nuevo semestre y se evalúa el inmediato anterior, esto debido a que como personal activo de UNICA se trabaja también en el periodo intersemestral es decir, se inician y finalizan labores no de acuerdo al calendario académico sino al administrativo por lo cuál es necesario realizar esta evaluación al inicio del siguiente semestre marcado en el calendario administrativos y toman en cuenta toda actividad realizada a lo largo del semestre como lo son: asesorías dadas, materias inscritas, aprobadas y reprobadas, desempeño en el trabajo, proyectos en los que participa, cursos impartidos y tomados, etc.

Esta evaluación se realiza con el propósito de determinar que becarios continuaran laborando en la Unidad para el siguiente semestre, quienes serán dados de baja del programa por un desempeño deficiente o bien quien se hace acreedor a alguna amonestación como por ejemplo el bajar de etapa.

A la par de la evaluación cada becario debe de entregar en una fecha que su jefe le indica el reporte de actividades de lo que hizo en el semestre inmediato anterior, desglosando detalladamente toda actividad para que estas se tengan en consideración al realizar la evaluación.

Estas evaluaciones junto con el reporte semestral correspondiente son enviadas al Comité de Becas formado por directivos de la Facultad quienes son los que toman las decisiones con respecto a los becarios.

Hasta el momento las evaluaciones han sido realizadas a mano y guardadas en archivos lo cuál dificulta la realización de búsquedas, generación de estadísticas y para llevar un control eficiente del desempeño de los becarios, ya que de ser necesario entregar reportes de estos datos el proceso se vuelve laborioso y complicado pues se deben de revisar una por una las evaluaciones que se tengan en los archivos y de ahí los generar los reportes. En cuanto a la evaluación representa un tiempo muy largo para realizarlas.

Como ya se ha mencionado anteriormente otra de las actividades que realizan los becarios es el desarrollo de proyectos actualmente los registros con lo que se cuentan son muy escasos y poco funcionales por el mismo hecho de que son realizados a mano y guardados en archivos o bien realizados en la computadora llegando a tener varios archivos esparcidos de los cuáles es difícil obtener información para la realización de reportes.

Debido a esta necesidad, se incorporó una sección en este sistema para la administración y control de los proyectos.

Esto permitirá tener un registro detallado de todos lo proyectos que se tienen en la Unidad y podrá ser visualizado por todos los jefes de los departamentos para que estén al tanto de los proyectos que se llevan a cabo.

Aunado a esto se tiene la necesidad de crear un registro de ex-becarios esto con el propósito de llevar un registro de quienes han pertenecido al plan y guardar datos relevantes de su desempeño dentro del mismo. Por esta razón se decidió incorporar una sección dedicada a este propósito.

Por lo tanto una vez analizadas las necesidades actuales de la Unidad de Servicios de Cómputo Académico, se definieron los siguientes módulos que conforman el Sistema de Evaluación de Becarios:

- Módulo I: Evaluación de Becarios
- Módulo II: Proyectos
- Módulo III: Ex becarios
- Módulo IV: Reporte Semestral de Actividades.

Cabe mencionar que si bien el sistema contará con estos cuatro módulos el presente trabajo estará enfocado al módulo I: Evaluación de Becarios, los restante son explicados para presentar una vista completa de lo que llegará a ser este sistema.

### **3.2.- Módulos del Sistema**

#### ***3.2.1.- Módulo I: Evaluaciones***

El departamento encargado de condensar las evaluaciones de todo el personal es el Departamento de Servicios Académicos con base al siguiente formato:



Facultad de Ingeniería  
Secretaría General  
Unidad de Servicios de Control Académico



EVALUACIÓN DE BECARIOS				
Nombre del Becario:		P.N. Cuenta:		Proyecto actual:
Departamento:		Semestre que cursa:		Período:
Ejemplar actual:		Ejemplar propuesta:		Evaluación:

Características	Desarrollo	Cantidad	%	Evaluación
Rendimiento académico	* Materias inscritas			
	* Materias aprobadas			
	* Materias reprobadas			
Asistencia y Puntualidad En UNICA	* Asistencia			
	* Retardos			
	* Faltas			
Iniciativa	* Disposición para colaborar			
	* Realización de proyectos			
	* Compromiso con UNICA			
Cursos	* Sábados y Domingos			
	* <del>Jubilados</del>			
	* Internos			
	* Especiales			
	* Prebeatos			
	* Elaboración de manuales			
	* Cursos asistidos			
Calidad en el trabajo	* Nuevas propuestas			
	* Responsabilidad			
	* Contabilidad			
Proyectos	* Actitud de servicio			
	* Líder de proyectos			
	* Participación de proyectos			
Asesorías	* *			
	* Asistencia			
	* retardos			
	* Faltas			

Comentarios:


Tabla 3.2.1.1 Formato de evaluación de becarios

En la parte superior del formato, se cuenta con un recuadro destinado a llevar la información personal del becario que es la siguiente:

<b>Datos del Becario</b>	<b>Información</b>
Nombre del Becario	
No. De Cuenta	Asignado por la UNAM
Promedio Actual	Dentro de la Facultad
Departamento	Al que pertenece dentro de UNICA
Semestre que Concluyó	Semestre que se va a evaluar
Periodo	Fecha que abarca el semestre
Etapa actual	Etapa en la que se encuentra el becario
Etapa propuesta	Etapa que ocupará el siguiente semestre
Evaluación	Calificación otorgada de acuerdo al desempeño.

*Tabla 3.2.1.2 Datos personales del Becario*

Dentro del campo para evaluar el desempeño del becario se encuentran los siguientes campos:

<b>Campo de Evaluación</b>	<b>Información</b>
Rendimiento Académico	<ul style="list-style-type: none"> <li>• Materias Inscritas</li> <li>• Materias Aprobadas</li> <li>• Materias Reprobadas</li> </ul>
Asistencia y Puntualidad en UNICA	<ul style="list-style-type: none"> <li>• Asistencia</li> <li>• Retardos</li> <li>• Faltas</li> </ul>
Iniciativa	<ul style="list-style-type: none"> <li>• Elaboración para colaborar</li> <li>• Elaboración de propuestas</li> <li>• Compromiso con UNICA</li> </ul>
Cursos	<ul style="list-style-type: none"> <li>• Intersemestrales</li> <li>• Semestrales</li> <li>• Sábados y Domingos</li> <li>• Prebecarios</li> <li>• Elaboración de Manuales</li> <li>• Especiales</li> <li>• Cursos Asistidos</li> <li>• Propuestas</li> </ul>
Proyectos	<ul style="list-style-type: none"> <li>• Líder de proyectos</li> <li>• Oportunidad de Avance</li> <li>• Participación en proyectos</li> </ul>
Calidad en el trabajo	<ul style="list-style-type: none"> <li>• Responsabilidad</li> <li>• Confiabilidad</li> <li>• Actitud de servicio</li> </ul>
Asesorías	<ul style="list-style-type: none"> <li>• Asistencia</li> <li>• Retardos</li> <li>• Faltas</li> </ul>

*Tabla 3.2.1.3 Datos de la evaluación de los becarios*

Al final de la evaluación se tiene un campo denominado comentarios, que se destina para realizar observaciones o notas sobre la evaluación y sobre el becario.

Actualmente este es el formato para realizar las evaluaciones dentro de la Unidad, para este sistema se han agregado otros puntos que son importantes, tanto en los datos generales como la creación de evaluaciones especiales de cada departamento.

### **3.2.1.1.-Puntos relevantes agregados a la evaluación**

De los datos personales del becario se agregaron:

- a) **Generación:** Cada becario pertenece a una generación cuando ingresa (Generación 42, Generación 43) este dato debe de ser guardado una sola vez: cuando se evalúa por primera vez a un becario.
- b) **Fecha de ingreso:** Compuesta por el año, mes y día en que el becario comenzó a laborar en la Unidad, este dato también debe ser guardado la primera vez que un becario es evaluado.
- c) **Notificaciones:** Estas se sucederán en el momento que el becario no cumpla con alguna responsabilidad dentro de UNICA.

### **3.2.1.2.- Evaluaciones Especiales para cada Departamento**

#### *3.2.1.2.1.- Evaluación del DID*

El Departamento de Investigación y Desarrollo (DID) es el encargado de mantenerse a la vanguardia en cuanto a software para web, multimedia, desarrollo y bases de datos. Es este departamento el que se encarga de llevar un control y seguimiento de los proyectos que se realizan en la Unidad, así como de sugerir el software más adecuado para el desarrollo de los mismos.

El DID aparte de la evaluación general requiere evaluar el desempeño de las distintas áreas que lo conforman siendo estas:

- **Web:** Encargado de mantener actualizada la página de la Unidad, así como también de la investigación de software multimedia como lo es: Macromedia Studio (Flash MX, Dreamweaver, Fireworks), 3Dstudio Max, entre otros.
- **Base de Datos:** Encargada de administrar las bases de datos con que cuenta, así como los servidores donde están alojadas las mismas.
- **Desarrollo:** Área dedicada a la investigación de nuevo software, uso del mismo y desarrollo de sistemas en diferentes lenguajes de programación de alto nivel como lo son: Java, Visual Basic 6.0, Visual Basic .NET, entre otros

Estas tres áreas del departamento se intercalan entre si para el desarrollo de sistemas ya que los participantes pertenecen a cada una de ellas así como también de ser necesario a otros departamentos dependiendo de las necesidades y requerimientos de los proyectos.

Los puntos clave dentro de la evaluación especial de estas tres áreas son:

- Disposición para colaborar de los integrantes
- Realización de propuestas (de acuerdo a su área )
- Compromiso (con su área y el departamento )

Para poder calificar estos tres puntos se tienen 4 niveles, cada uno de ellos con un puntaje:

Nivel	Puntaje para la evaluación
Siempre	100% El becario participa activamente
Muchas veces	75% El becario participa por lo general
Casi siempre	50% El becario participa regularmente
Pocas veces	0% El becario no participa

*Tabla 3.2.1.2.1.1 Puntajes para la evaluación*

### *3.2.1.2.2.-Evaluación del DROS*

El Departamento de Redes y Operación de Servidores es el encargado como su nombre lo indica de dar mantenimiento y soporte a los servidores con los que cuenta la Facultad de Ingeniería y que se encuentran a cargo de UNICA, el más importante de ellos es el servidor conocido como Cancún, en este se encuentran alojadas las cuentas de correo electrónico de cada alumno y profesor de la Facultad, siendo renovadas semestre a semestre.

También está encargado de dar soporte a distintas dependencias de la Facultad en cuanto al Sistema Operativo Microsoft Windows en cualquiera de sus versiones esto es realizando respaldos de información en caso de que una máquina falle, mantenimiento preventivo y reparación de aquellas fallas que puedan ser subsanadas por el personal de este departamento, así también cuenta con personal capacitado para la administración de máquinas que trabajen bajo la plataforma de software libre UNIX o LINUX con los diferentes entornos de trabajo (Gnome, Fedora) este mismo personal es el que está encargado de la administración y uso de las cinco estaciones de trabajo Solaris con que se cuenta, ubicadas en las salas del edificio principal, junto a la librería.

Debido a lo anterior, la evaluación especial de DROS, se enfoca a los siguientes puntos:

1. Número de atención de solicitud de eventos de soporte técnico atendidos:
  - a. Número de atendidos totalmente
  - b. Número de atendidos parcialmente

2. Realización de Respaldos de Servidores
  - a. Número de respaldos realizados
  - b. Cantidad de información en MB respaldada
3. Número de sistemas para la administración al que le da mantenimiento
  - a. Cantidad
  - b. Actualización del sistema
4. Número de Aportaciones al Sistema de conocimientos de Administración en Cómputo.
  - a. Cantidad
5. Número de atención a incidentes de operación de servidores y redes fuera del horario hábil.
  - a. Cantidad de eventos.
  - b. Número de horas invertidas.

#### *3.2.1.2.3.-Evaluación del DSC*

El Departamento de Seguridad en Cómputo recientemente creado es el encargado de mantener vigilada la seguridad de la red interna así como de mantener informado a todos los que laboran en UNICA sobre los nuevos virus informáticos, las actualizaciones y parches para los equipos y también de la prevención y detección de ataques a los servidores.

Para su evaluación el DSC requirió el siguiente punto:

1. Atención a incidentes de seguridad
  - a. Atención satisfactoria (proceso realizado y concluido por completo)
  - b. Atención insatisfactoria (una parte del proceso no se concluyó)

En cuanto a los departamentos de Servicios Académicos y Control de Salas, hasta el momento no han manifestado el deseo de tener una evaluación especial.

### **3.2.1.2.- Rangos y Criterios para la Evaluación**

Dado que dentro de este sistema se evalúa el desempeño que tuvieron los becarios dentro de la unidad a lo largo del semestre se le dio un porcentaje significativo a los puntos que se evalúan para poder obtener una calificación de acuerdo al desempeño. Estos rangos son:

#### *1.- Rendimiento Académico*

Se evalúan los siguientes puntos:

- Materias inscritas
- Materias aprobadas
- Materias reprobadas
- 

Cada uno de ellos es obtenido mediante el historial académico del becario; para considerar que un becario cumplió con el rendimiento académico se tiene que:

- Número de materias inscritas = Número de materias aprobadas
- Si el porcentaje es  $\geq$  a 80 %  $\Rightarrow$  Cumplió
- Si el porcentaje es  $<$  a 80 %  $\Rightarrow$  que no cumplió

Esto implica que el becario sólo puede tener un margen de una sola materia reprobada al semestre. En caso de que tenga dos o más materias se tomará caso especial e implicará una notificación en la evaluación.

## 2.- *Asistencia y Puntualidad:*

### 2.1) Asistencias

El becario debe tener un mínimo de un 90% de ellas, donde:

número de asistencias = número de días que tiene el semestre administrativo (sin contar periodos vacacionales y días no laborables)

### 2.2) Retardos

Se tomará como tal cuando si se busca al becario a sus horas 2 veces y éste no está.

### 2.3) Faltas

- El Ing. Barranco Vite designara como se tomarán.

## 3.- *Iniciativa*

- Disposición para colaborar
- Realización de propuestas
- Compromiso con UNICA.

Cada uno de estos puntos contarán con cuatro opciones para evaluar y cada uno de ellos tendrá un puntaje que es el siguiente:

- Siempre.....100
- Muchas veces.....75
- Algunas veces.....50
- Muy pocas veces.....0

## 4.- *Cursos*

- Intersemestrales
- Internos
- Especiales

- Sábados y Domingos
- Semestrales
- Prebecarios

Para evaluar los cursos, se tomarán los datos de los cursos que haya dado el becario de acuerdo a los datos que se encuentran en el Sistema de Inscripciones y Control de Cursos (SICC) a excepción de los cursos a Prebecarios, este número será introducido por cada jefe de Departamento.

El dato es numérico e indica la cantidad e cursos que impartió el becario a lo largo del semestre.

#### *5.- Manuales*

Cada jefe de departamento colocará el número de manuales que el becario terminó durante el semestre; no se incluyen aquellos que hayan sido terminados a medias o bien que se encuentren en revisión.

#### *6.- Cursos asistidos*

Se tomarán los datos del SICC en donde el becario aparezca como alumno y haya obtenido una constancia del curso es decir lo aprobó.

#### *7.- Nuevas propuestas*

- Queda a criterio de cada jefe de departamento.

#### *8.- Calidad en el Trabajo*

- Responsabilidad
- Confiabilidad
- Actitud de Servicio

Al igual que la iniciativa, se podrá elegir entre las cuatro mismas opciones con el mismo puntaje.

- Siempre.....100
- Muchas veces.....75
- Algunas veces.....50
- Muy pocas veces.....0

#### *9.- Proyectos:*

##### *9.1) Líder de proyectos*

Se pondrá el número de proyectos en los cuales el becario fue o es líder

##### *9.2) Participación en proyectos*

Se pondrá el número de proyectos en los cuales el becario participó o está participando

##### *9.3) Oportunidad de avance*

Se indica que tanto avance tuvo en relación con los proyectos que está realizando el becario, dentro de los siguientes rangos y puntajes:

- Oportuno.....100
- Regular.....75
- Mínimo.....50
- Ninguno.....0

#### *9.4) Resultados*

Se indica en que momento el becario entregó lo que se le pidió del proyecto teniendo los mismos rangos y puntajes del punto anterior.

#### *10.- Asesorías*

Así como dentro del rubro asistencia y puntualidad en UNICA, aquí también se toman en cuenta los mismos puntos:

- Asistencias
- Faltas
- Retardos

Ellos serán obtenidos del Sistema del Control de las Asesorías una vez que esté funcionando, mientras tanto serán introducidos por el usuario.

Solamente se permiten tres faltas injustificadas, todas aquellas que sean por motivos inesperados o fuera del control del becario deben ser notificadas y si procede mandar un justificante al DSA. El becario que tenga más de tres faltas a las asesorías, será acreedor a una notificación dentro de su evaluación.

#### *11.- Puntos especiales para los departamentos:*

##### *11.1.- Departamento de Seguridad:*

El departamento de seguridad desea el siguiente punto para evaluar a los becarios que están en el:

- Atención a incidentes de seguridad

El cuál contará con dos opciones para evaluar al becario:

- Atención satisfactoria (todo el proceso realizado y concluido completamente)
- Atención insatisfactoria (no se concluyó el proceso)

### *11.2.- Departamento de Investigación y Desarrollo*

Se incluirán puntos para evaluar a los becarios que se encuentran en las siguientes áreas del departamento:

- Desarrollo
- Base de datos
- WEB

Cada uno con:

- Disposición para colaborar
- Realización de propuestas
- Compromiso

Usando los mismos puntos y rangos que se utilizan en el punto de Iniciativa y Calidad en el trabajo.

### *11.3.- Departamento de Redes y Operación de Servidores.*

Básicamente en esta evaluación se deben de indicar cada uno de los campos que se mencionan en el punto 3.2.1.2.2, ellos son numéricos.

### ***3.2.2.- Módulo II: Proyectos***

Como ya se mencionó en la Unidad de Servicios de Computo Académico se llevan a cabo diferentes proyectos estos con el propósito realizar una automatización de funciones, los proyectos son realizados tanto por el personal que se encuentra contratado como por los becarios siendo estos últimos la mayor parte de los equipos de trabajo.

A cada proyecto se le asigna un líder que será la persona que se mantendrá en contacto con el cliente (por lo general el DSA, aunque pudiera cambiar) quien es el que tiene los requerimientos del sistema. A la par se asigna un equipo de trabajo para el proyecto que tendrá la obligación de desarrollar el sistema de acuerdo a los requerimientos establecidos y en el tiempo acordado.

Como actualmente el registro de los proyectos se lleva a cabo de forma manual, se pretende que este módulo subsane las carencias que este sistema de archivos provoca, pudiendo generar reportes y visualizar proyectos de manera fácil y rápida.

Los puntos que se registran para los proyectos son:

**NOMBRE:** A cada proyecto se le asigna un nombre de acuerdo a lo que va a realizar.

**SIGLAS:** De acuerdo al nombre se extraen las siglas con las que el proyecto se conocerá

**FECHA DE INICIO:** Cuando inicia el sistema siguiendo el formato de: año/mes/día

**FECHA DE FIN:** Fecha en la cuál el sistema debe de ser entregado y liberado, con el mismo formato que la fecha de inicio: año/mes/día.

**AVANCE:** Cantidad en porcentaje de acuerdo a cuanto se ha realizado del sistema

**ETAPA:** Fase en la cuál se encuentra el proyecto que puede ser:

*INICIO:* Se discute la necesidad de un sistema para automatizar algún(os) proceso(s)

*PLANEACION:* Se analizan los requerimientos, se define el software y se realiza el modelado del sistema

*DESARROLLO:* Se programa todo aquello que se modelo durante la fase de planeación

*PRUEBAS:* Se realizan pruebas de usuario, para observar cualquier anomalía que se presente y subsanarla antes de la liberación

*LIBERACION:* El sistema es entregado a los usuarios finales, sin errores, cumpliendo con todos los requerimientos.

**LIDER:** Se captura el nombre del que será el responsable de la realización del proyecto

**PARTICIPANTES:** Nombre de las demás personas involucradas dentro del proyecto.

### ***3.2.3.- Módulo III: Ex becarios***

Una vez que un alumno de la Facultad fue seleccionado como becario, permanecerá dentro de la Unidad realizando diversas actividades una vez que decide marcharse pasa a formar parte del grupo de Ex becarios, actualmente no existe un registro formal de aquellas personas que estuvieron laborando en UNICA.

Por esta razón se decidió implementar un módulo donde se pudiera visualizar a aquellas personas que laboraron en la Unidad, teniendo los siguientes datos:

- Nombre completo del Ex becario.
- Departamento al que perteneció.
- Fecha en la que ingresó.
- Fecha en la que se fue.
- Razón por la que se fue.
- Empresa (si es el caso) a la que se fue a laborar.

### ***3.2.4.- Módulo IV: Reporte Semestral***

Como se mencionó en el Módulo I, junto con la evaluación se envía al comité el Reporte Semestral que cada becario elabora y entrega a su jefe inmediato.

Con este módulo se pretende crear un registro de los reportes semestrales que un becario ha realizado a lo largo de su estancia en UNICA, para poder servir como comparativo de las actividades realizadas semestre a semestre por el.

Este módulo será dividido en dos partes:

- a) Registro del Reporte vía Internet
- b) Visualización del Reporte dentro del SICEB.

El registro del reporte pretende hacerse mediante una interfaz realizada en PHP, con conexión a base de datos esto interactuando con el Módulo II del Sistema de Inscripción y Control de Cursos (SICC ) ya que es este el que permite a cada becario vía Web dar de alta su horario de trabajo en UNICA y modificar sus datos personales; cada inicio de semestre el acceso a este sistema se realiza con un *login* y *password* de usuario se pretende que en el momento que se de de alta el horario para el nuevo semestre exista una opción que le permita al becarios introducir su reporte semestral de actividades para con esto acortar el tiempo y simplificar esta tarea.

Posteriormente al realizar la evaluación correspondiente al semestre que terminó cada jefe de departamento podrá visualizar el reporte que el becario introdujo al sistema, para de esta forma poder tomar en cuenta su desempeño al momento de evaluarlo.

*Modelado del  
SICEB*

Capítulo

*4*

## 4.- Tecnología Empleada y Modelado del Sistema

Para la realización del SICEB se tuvieron que considerar diferentes tecnologías para su desarrollo luego compararse se eligieron las que se describen en las tablas siguientes:

### 4.1.- Tecnología Empleado

#### 4.1.1.- Interfaz Gráfica

Tecnología	Razón
Visual Basic 6.0	Es un entorno de desarrollo de programas de forma gráfica, es de fácil uso además de que el SICEB debía interactuar y posteriormente ser acoplado a otro sistema ya existente, creado en este mismo entorno. Se pueden crear aplicaciones orientadas a objetos en donde las propiedades de ellos están encapsuladas asegurando que puedan ser privados o públicos, accediendo a ellos mediante procedimientos.
Crystal Reports	Una parte fundamental del sistema era el poder generar reportes o informes para el usuario, con base en esto y en la tecnología anterior se eligió a Crystal Reports como herramienta para generarlos ya que está diseñado para trabajar en conjunto con Visual Basic, es un programa amigable que permite crear reportes tanto estáticos como dinámicos de manera rápida y con visibilidad en tiempo de diseño, además que permite personalizar los reportes para generarlos de acuerdo a las especificaciones del cliente.

*Tabla 4.1.1 Tecnología utilizada para la interfaz y programado del sistema*

**4.1.2.- Modelado del Sistema**

<b>Tecnología</b>	<b>Razón</b>
UML	El lenguaje de modelado unificado nos proporciona las herramientas necesarias para modelar sistemas orientados a objetos en su totalidad lo que proporciona una vista del sistema incluso antes de que este sea construido, lo cual ayuda a reducir tiempos de desarrollo y también busca reducir tiempos en cuanto a pruebas y manejos de errores, al diseñar desde el momento que se tienen los requerimientos el funcionamiento del sistema.
VISIO 2000	Nos permite realizar modelos para el diseño de bases de datos de alto desempeño para aplicaciones cliente/servidor esta herramienta nos permite no sólo crear el modelo lógico sino también construye a la par las estructuras de datos físicos con el fin de realizar una optimización ya que cuando el modelo está listo simplemente se selecciona el servidor donde se quiere construir la base y eligiendo las opciones de generación de base Visio automáticamente construye la base en el servidor incluyendo tablas, índices, procedimientos, triggers, y otros componentes.

*Tabla 4.1.2 Tecnología utilizada para el modelado del Sistema*

**4.1.3.- Bases de Datos**

<b>Tecnología</b>	<b>Razón</b>
PostgreSQL	Como manejador de base de datos se eligió éste pues aparte de ser uso gratuito es lo que se conoce como una base de datos relacional es un manejador robusto basado en sentencias SQL para realizar las consultas, de fácil uso y mantenimiento, proporcionando integridad en los datos, herencia de tablas, tipos de datos y vistas entre otras muchas funciones.
ODBC	Siglas de Open Data Base Connectivity (Conectividad abierta de base de datos), nos permite la comunicación con las bases de datos sin importar el servidor o manejador en el que estén es decir si la base cambia de manejador basta cambiar el ODBC para que la aplicación siga funcionando y no se tenga que reprogramar toda. Para el SICEB se utilizó el controlador de ODBC PostgreSQL, que viene en forma de programa que se ejecuta, instala y configura para que se comunique con las bases
ADO	Los Active Data Object, permiten crear sistemas y aplicaciones, capaces de interactuar con las bases de datos, siendo una interfaz de programación a nivel de aplicación para los datos y la información. De los objetos ADO, utilizamos los objetos RecordSet, Connection y Command, para el manejo de los datos.

*Tabla 4.2.3 Tecnología utilizada para el manejo de las bases de datos.*

## 4.2.- *El SICC y el SICEB*

Para poder desarrollar el Sistema de Evaluación de Becarios de acuerdo a los requerimientos y especificaciones del cliente (Capítulo 3) se debía de contar con algunos de los datos personales de los becarios que laboran en UNICA. Como éstos ya se encuentran capturados dentro del Sistema de Inscripción y Control de Cursos (SICC), el problema de obtenerlos se reducía a realizar una consulta a esta base de datos.

Pero ¿cómo está constituido el SICC?

El SICC se desarrolló por la necesidad de contar con un sistema que controlara:

- Los cursos que la unidad ofrece a los alumnos de la Facultad, así como a personas ajenas a ella.
- Los datos completos de los becarios que laboran en UNICA, siendo conocidos dentro del sistema como instructores pues son ellos los que imparten los cursos.
- El horario de trabajo del instructor (becario) dentro de la unidad es decir, las horas en las que se le puede encontrar en su cubículo.
- Los alumnos inscritos en los cursos, estos pueden estar registrados en más de uno a la vez, además de que se indica si fueron confirmados o no para tomarlo y si obtuvieron constancia o no.
- Registro de las calificaciones de los alumnos.
- Creación de las constancias tanto de los alumnos que aprobaron el curso como de los instructores que lo dieron.
- Manejo de los cursos: altas, bajas, en los diferentes periodos: semestrales, intersemestrales, sábados y domingos, especiales.
- Generación de estadísticas sobre los tipos de alumnos (alumno de la Facultad, beca, externo o bien personal o alumno de otra dependencia de la UNAM), tipos de grupos (abiertos, cancelados en el periodo).
- Administración de salas (características, descripción de ellas, equipo que tiene, ubicación, capacidad, dar de alta nuevas salas).

- Manejo de los costos de un curso dependiendo del nivel: básico, intermedio y avanzado.

Estas son las funcionalidades del SICC, éste fue desarrollado en Visual Basic 6.0 con conexión a Base de datos PostgreSQL y generando los reportes en Crystal Reports.

La parte que nos interesa sobre todo del SICC es la correspondiente a los instructores puesto es en esa tabla donde se guardan sus datos personales.

Ella se encuentra dentro de la base de datos llamada BDSICC y que esta alojada en el servidor de la Unidad llamado “Tork”. Para poder acceder a estos datos fue necesario pedir al administrador del servidor permiso para poder crear un usuario que pudiera tener acceso de sólo lectura a esta base. Es importante aclarar que el SICEB no modifica ningún registro de esta base, en el momento de borrar alguna evaluación la acción es llevada a cabo directamente de la base de datos del SICEB, sin alterar ningún dato del SICC.

List of relations			
Schema	Name	Type	Owner
public	Untitled	table	siceb
public	alumno	table	sicc
public	alumno_id_alumno_seq	sequence	sicc
public	antecedente	table	sicc
public	costo_grupo	table	sicc
public	curso	table	sicc
public	curso_antecedente	table	sicc
public	experiencia	table	sicc
public	experiencia_id_experiencia_seq	sequence	sicc
public	grupo	table	sicc
public	grupo_alumno	table	sicc
public	grupo_id_grupo_seq	sequence	sicc
public	horario_instructor	table	sicc
public	instructor	table	sicc
public	instructor_grupo	table	sicc
public	instructor_id_instructor_seq	sequence	sicc
public	nivel_curso	table	sicc
public	nombre	table	siceb
public	periodo	table	sicc
public	personal	table	sicc
public	propuesta	table	sicc
public	sala	table	sicc
public	seccion_sala	table	sicc
public	tipo_alumno	table	sicc
public	tipo_grupo	table	sicc
public	tipo_personal	table	sicc
public	tipo_personal_id_tipo_seq	sequence	sicc

Tabla 4.2.1 Contenido de la base de datos del SICC

Column	Type	Modifiers
id_instructor	integer	not null default nextval('instructor_id_instructor_seq'::text)
nombre	character(50)	not null
apellidos	character(50)	not null
no_cuenta	character(9)	
rfc	character(10)	not null
id_tipo	integer	not null
departamento	character(30)	not null
calle	character(50)	not null
colonia	character(50)	not null
delegacion	character(50)	not null
cp	character(6)	not null
telefono	character(10)	not null
telefono_op	character(15)	
email	character(50)	
fecha_ingreso	date	
semestre	integer	
hora_aprob	boolean	
activo	boolean	default true

Tabla 4.2.2 Contenido de la tabla de instructores

En la tabla 4.2.2 podemos apreciar los campos que se encuentran almacenados dentro de la tabla instructor, de ellos analizando los requerimientos se observó que solamente se necesitan los siguientes campos:

- Id\_instructor : Es el identificador único e irrepitible para cada instructor
- Nombre del instructor
- Apellidos
- Número de cuenta
- Departamento: Al que pertenecen dentro de UNICA
- Activo: Este campo nos servirá para discriminar a aquellos becarios que siguen trabajando de los que ya se han ido, su utilidad primordial será en el momento de realizar las consultas.

Estos datos serán lo que se desplegarán en la sección de datos personales de la evaluación.

Básicamente la interacción con el SICC y el SICEB es en el momento de cargar las evaluaciones y también en el momento de mostrar los datos personales de los becarios que faltan de ser evaluados.

Para la opción de cargar antes de ir por las evaluaciones se realiza una consulta a la base de datos del SICC para traer los datos personales de los becarios que se encuentran activos de acuerdo al usuario (departamento) que haya entrado al sistema (DID, DROS, DSA, DSC, DCS ).

```
select id_instructor,nombre,apellidos,departamento,no_cuenta from instructor where activo = 't'  
and departamento = "" & usuario & "";
```

En el *query* o consulta anterior observamos que pedimos los datos de los becarios de la tabla instructor del SICC donde activo sea verdadero y el departamento coincida con una variable llamada usuario que es la que almacena el departamento que entró al sistema. Por ejemplo si el jefe del DROS se encuentra utilizándolo, la consulta quedaría de la siguiente manera:

```
select id_instructor,nombre,apellidos,departamento,no_cuenta from instructor where activo = 't'  
and departamento = " DROS ";
```

La variable “usuario” se define al entrar al sistema en el momento de introducir el nombre de usuario y contraseña del mismo.

Una vez realizada esta consulta, se procede a seleccionar cada uno de los becarios para poder traer sus evaluaciones de la base de datos del SICEB.

### **4.3.- Modelado del Sistema**

#### **4.3.1.-Análisis**

##### *4.3.1.1.- Modelo Entidad Relación*

Para el SICEB y con base en los requerimientos se creó el modelo entidad relación de la base de datos que alojaría toda la información del sistema. Para esto y con ayuda de la herramienta de modelado conocida como Visio 2000, se generó el modelo el cual incluye dos tablas que no son propias del sistema: la tabla de cursos y la de instructores ambas pertenecientes al SICC.

Las tablas que conforman el modelo son:

- **Instructor:** No pertenece a la base de datos del SICEB, sin embargo contiene datos que son necesarios para las relaciones entre las tablas, por lo tanto es incluida además de que parte de su información es empleada en el sistema.
- **Evaluación:** En esta tabla se almacenan los datos de las evaluaciones de todos y cada uno de los becarios, la llave de esta tabla es compuesta formada por el identificador único de cada becario y el identificador del periodo al que pertenece la evaluación.
- **Periodo:** Almacena la información referente a las fechas de inicio y fin del periodo.
- **Cursos:** Pertenece al SICC de ella se obtienen el número de cursos semestrales, intersemestrales, especiales, de sábados y domingos que dió el becarios.
- **DROS Especial:** Guarda la información de la evaluación especial de este departamento.
- **DSC Especial:** Guarda la información de la evaluación especial del Departamento de Seguridad en Cómputo.
- **Gen\_becario:** Guarda la información de la generación y fecha de ingreso del becario a la Unidad.

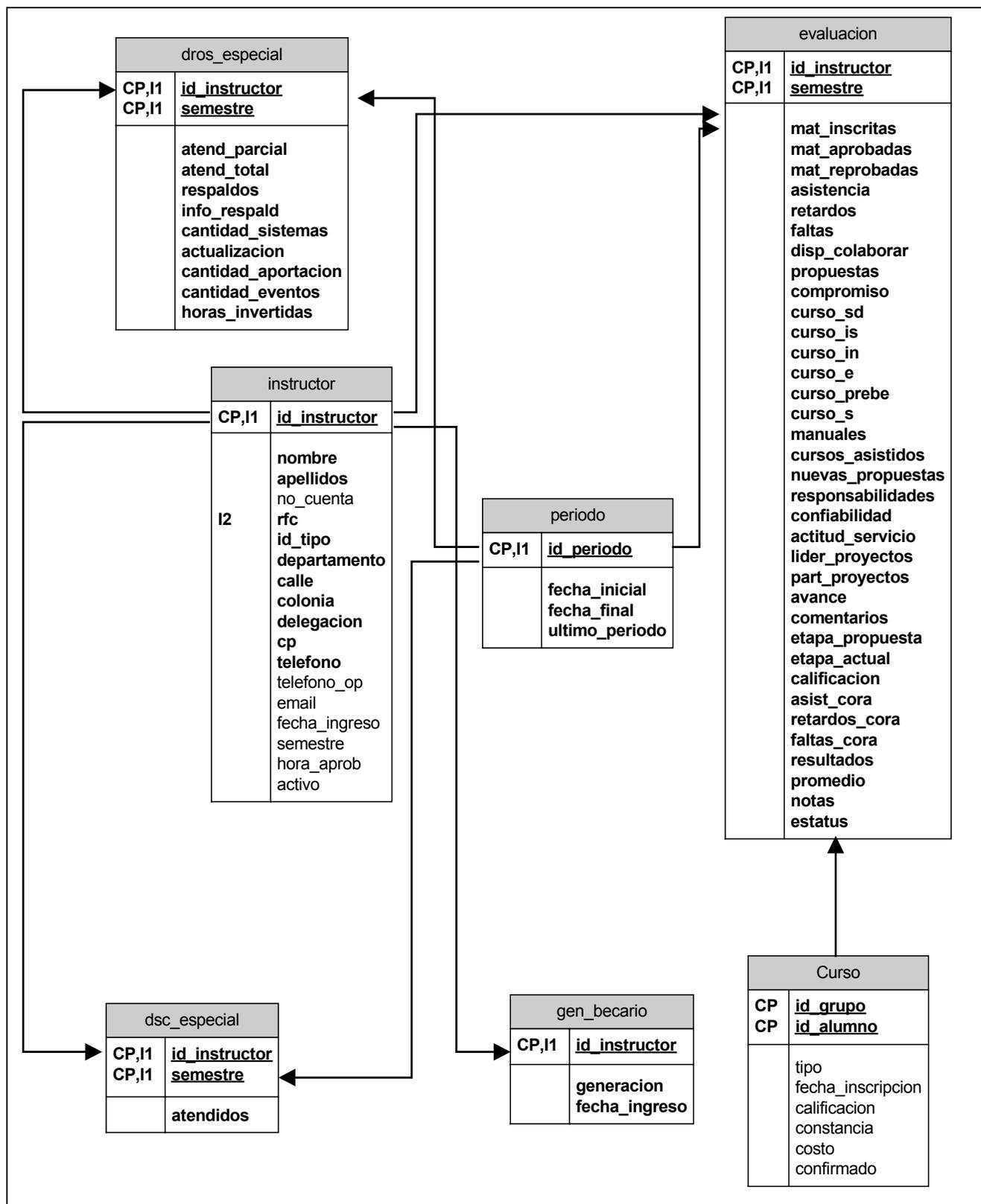


Figura 4.3.1.1.1 Diagrama Entidad Relación

#### 4.3.1.2.- Diagramas Casos de Uso

Estos tienen la finalidad de capturar los objetivos del sistema y de manera general las especificaciones de los requerimientos dentro de ellos podemos identificar a los casos de uso y a los actores.

Dentro del SICEB, se identificaron a los siguientes actores:

- **Usuario:** Dentro del SICEB tendremos cuatro usuarios normales es decir que no tienen privilegios de administrador estos serán los Departamentos de: Investigación y Desarrollo, Redes y Operación de Servidores, Seguridad en Cómputo y Control de Salas, contando con el siguiente caso de uso: Manipular Evaluaciones.
- **Administrador:** Será el encargado manipular el sistema para que cada semestre pueda ser utilizado; el administrador será el Departamento de Servicios Académicos este tendrá casos de uso especiales aparte del mencionado para los usuarios esto son: Alta de Periodo, Generación de Reportes y Manipulación de Todas las Evaluaciones.

##### 4.3.1.1.1.-Descripción de los casos de uso:

Manipular Evaluaciones: Dentro de este caso de uso general, encontramos varios particulares que son:

- **Nueva Evaluación:** Servirá para consultar los becarios que faltan de ser evaluados y por consiguiente evaluarlos, incluyendo para los usuarios que así lo requirieron el generar la evaluación especial.
- **Modificar Evaluación:** Servirá para cambiar datos de aquellas que pertenezcan al periodo actual de evaluación.
- **Borrar Evaluación:** Para eliminar los registros que ya no se deseen conservar en el sistema; cuenta con una confirmación.
- **Cargar Evaluaciones:** Para poder traer desde la base de datos todos los registros existentes para el usuario que esté utilizando el sistema (los becarios que pertenezcan a ese departamento y cuenten con al menos una evaluación).

- *Realizar búsquedas:* Para disminuir de acuerdo a ciertos parámetros los registros que se quieren consultar, es decir aquellos que cumplan con cierto(s) criterio(os).

En los casos de uso del administrador tenemos:

- *Alta de Periodo:* Cada nuevo periodo el administrador será el encargado de ingresar en el sistema las fechas de inicio y fin, esto es necesario pues de lo contrario ningún otro usuario podrá evaluar a sus becarios.
- *Generación de Reportes:* Ya que el Departamento de Servicios Académicos es el encargado del control de todos los becarios también se hará cargo de generar los reportes.
- *Manipulación de todas las Evaluaciones:* El administrador también juega el papel de usuario normal, por lo tanto realiza lo que indica el caso de uso: “manipular evaluación”, pero también como administrador puede manipular todas las evaluaciones con la restricción de que no puede evaluar a los becarios de otros departamentos, pero si verlas y modificar las del periodo actual en caso de que sea necesario.

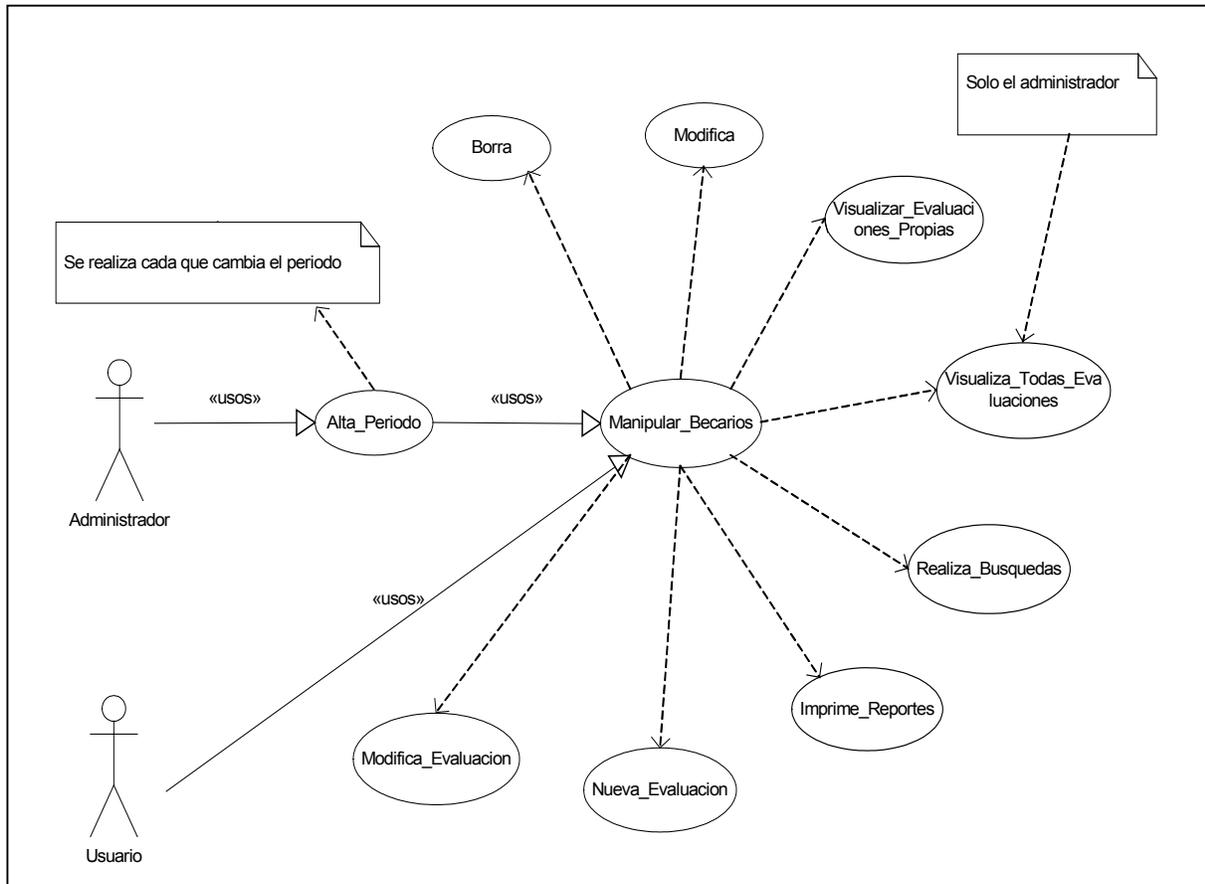


Figura 4.3.1.2.1 Diagrama de casos de uso

### 4.3.1.2.- Diagramas Clases

#### 4.3.1.2.1 Clases que integran el SICEB

Para poder implementar el sistema de forma orientada a objetos, se debieron crear diferentes clases de acuerdo al modelo entidad relación, considerando que cada tabla que en él se encuentre pasa a ser una clase de donde se podrán crear objetos y también que existirán clases especializadas para realizar las conexiones a bases de datos.

#### 1.- CEvaluación:

Esta clase es donde se crearan los objetos del tipo “*evaluación*” en los cuales se almacenarán los datos de la tabla evaluación, pasando entonces a ser atributos del objeto “*evaluación*”.

SICEB::CEvaluacion
-intid_instructor : Integer
-str_semestre : String
-intmat_inscritas : Integer
-intmat_aprobadas : Integer
-intmat_reprobadas : Integer
-int_asistencia : Integer
-int_retardos : Integer
-int_faltas : Integer
-str_colaboracion : String
-str_propuestas : String
-str_compromiso : String
-intcurso_SD : Integer
-intcurso_IS : Integer
-intcurso_IN : Integer
-intcurso_E : Integer
-intcurso_Prebe : Integer
-intcurso_S : Integer
-int_manuales : Integer
-intcurso_asistidos : Integer
-strnuevas_propuestas : String
-str_responsabilidades : String
-str_confabilidad : String
-stractitud_servicio : String
-strlider_proyecto : String
-strpart_proyectos : String
-str_avance : String
-str_comentarios : String
-stretapa_propuesta : String
-stretapa_actual : String
-str_calificacion : String
-intcora_retardos : Integer
-intcora_faltas : Integer
-str_resultados : String
-str_periodo_inicio : String
-str_periodo_fin : String
-intcora_asistencia : Integer
-int_promedio : String
-str_estatus : String
-str_notificacion : String
*get_id() : Integer
*le_id(entrada_id_instructor : Integer)
*get_semestre() : String
*le_semestre(entrada_semestre : String)
*get_intmat_inscritas() : Integer
*le_intmat_inscritas(entrada_intmat_inscritas : Integer)
*get_intmat_aprobadas() : Integer
*le_intmat_aprobadas(entrada_intmat_aprobadas : Integer)
*get_intmat_reprobadas() : Integer
*le_intmat_reprobadas(entrada_intmat_reprobadas : Integer)
*get_int_asistencia() : Integer
*le_int_asistencia(entrada_int_asistencia : Integer)
*get_int_retardos() : Integer
*le_int_retardos(entrada_int_retardos : Integer)
*get_int_faltas() : Integer
*le_int_faltas(entrada_int_faltas : Integer)
*get_str_colaborar() : String
*le_str_colaborar(entrada_str_colaborar : String)
*get_str_propuesta() : String
*le_str_propuesta(entrada_str_propuesta : String)
*get_str_compromiso() : String
*le_str_compromiso(entrada_str_compromiso : String)
*get_int_csd() : Integer
*le_int_csd(entrada_int_csd : Integer)
*get_int_cis() : Integer
*le_int_cis(entrada_int_cis : Integer)
*get_int_cin() : Integer
*le_int_cin(entrada_int_cin : Integer)
*get_int_ce() : Integer
*le_int_ce(entrada_int_ce : Integer)
*get_int_prebes() : Integer
*le_int_prebes(entrada_int_prebes : Integer)
*get_int_semestrales() : Integer
*le_int_semestrales(entrada_int_semestrales : Integer)
*get_int_manual() : Integer
*le_int_manual(entrada_int_manual : Integer)
*get_int_asistidos() : Integer
*le_int_asistidos(entrada_int_asistidos : Integer)
*get_str_nuevas_prop() : String
*le_str_nuevas_prop(entrada_str_nuevas_prop : String)
*get_str_responsabilidad() : String
*le_str_responsabilidad(entrada_str_responsabilidad : String)
*get_str_confabilidad() : String
*le_str_confabilidad(entrada_str_confabilidad : String)
*get_str_servicio() : String
*le_str_servicio(entrada_str_servicio : String)
*get_str_lider() : String
*le_str_lider(entrada_str_lider : String)
*get_str_participacion() : String
*le_str_participacion(entrada_str_participacion : String)
*get_str_avance() : String
*le_str_avance(entrada_str_avance : String)
*get_str_coment() : String
*le_str_coment(entrada_str_coment : String)
*get_str_etapa() : String
*le_str_etapa(entrada_str_etapa : String)
*get_str_etapa_actual() : String
*le_str_etapa_actual(entrada_str_etapa_actual : String)
*get_str_calificacion() : String
*le_str_calificacion(entrada_str_calificacion : String)
*get_str_periodo() : String
*le_str_periodo(entrada_str_periodo : String)
*get_str_periodo_inicio() : String
*le_str_periodo_inicio(entrada_str_periodo_inicio : String)
*get_str_periodo_fin() : String
*le_str_periodo_fin(entrada_str_periodo_fin : String)
*get_int_corasistencia() : Integer
*le_int_corasistencia(entrada_int_corasistencia : Integer)
*get_int_coraretardo() : Integer
*le_int_coraretardo(entrada_int_coraretardo : Integer)
*get_int_corafalta() : Integer
*le_int_corafalta(entrada_int_corafalta : Integer)
*get_str_resultado() : String
*le_str_resultado(entrada_str_resultado : String)
*get_str_promedio() : String
*le_str_promedio(entrada_str_promedio : String)
*get_str_estatus() : String
*le_str_estatus(entrada_str_estatus : String)
*get_str_notas() : String
*le_str_notas(entrada_str_notas : String)

Tabla 4.3.1.2.1.1 Clase Evaluación

## 2.- Cbecario

De ella se obtienen los objetos de tipo “*becario*”, los cuales almacenarán los datos pertenecientes a los becarios.

SICEB::CBecario
-intid_instructor : Integer -str_nombre : String -str_apellido : String -str_dpto : String -strno_cuenta : String -str_generacion : String -str_ingreso : String
+get_ingreso() : String +let_ingreso(entrada fecha : String) +get_generacion() : String +let_generacion(entrada generacion : String) +get_id() : Integer +let_id(entrada id_becario : Integer) +get_nombre() : String +let_nombre(entrada nombre_becario : String) +get_apellido() : String +let_apellido(entrada apellido_becario : String) +get_dpto() : String +let_dpto(entrada dpto_becario : String) +get_cuenta() : String +let_cuenta(entrada no_cuenta : String)

Tabla 4.3.1.2.1.2 Clase Becario

## 3.- CCurso

De esta clase se crearán los objetos “*curso*”, los cuales contendrán los datos del número de cursos que el becario a impartido a lo largo del semestre.

SICEB::CCurso
-intcurso_SD : Integer -intcurso_IS : Integer -intcurso_IN : Integer -intcurso_E : Integer -intcurso_Prebe : Integer -intcurso_S : Integer -intcursos_asistidos : Integer
+get_semestral() : Integer +let_semestral(entrada semestral : Integer) +get_csd() : Integer +let_csd(entrada csd : Integer) +get_cis() : Integer +let_cis(entrada cis : Integer) +get_cin() : Integer +let_cin(entrada cin : Integer) +get_ce() : Integer +let_ce(entrada ce : Integer) +get_prebes() : Integer +let_prebes(entrada prebes : Integer) +get_asistidos() : Integer +let_asistidos(entrada asistidos : Integer)

Tabla 4.3.1.2.1.3 Clase Cursos

#### 4.- CDros\_Especial

De esta clase se obtendrán los objetos “*dros*”, en el cual los atributos serán los que tiene la evaluación especial del DROS.

SICEB::CDros_Especial
-intid : Integer -str_semestre : String -intatend_parcial : Integer -intatend_total : Integer -intrespaldos : Integer -intinfo_respald : Integer -intcantidad_sistemas : Integer -intactualizacion : Integer -intcantidad_aportacion : Integer -intcantidad_eventos : Integer -inthoras_invertidas : Integer
+get_id() : Integer +let_id(entrada id : Integer) +get_semestre() : String +let_semestre(entrada semestre : String) +get_atend_parcial() : Integer +let_atend_parcial(entrada parcial : Integer) +get_atend_total() : Integer +let_atend_total(entrada total : Integer) +get_respaldos() : Integer +let_respaldos(entrada respaldos : Integer) +get_info_respald() : Integer +let_info_respald(entrada info : Integer) +get_cantidad_sistemas() : Integer +let_cantidad_sistemas(entrada sistemas : Integer) +get_actualizacion() : Integer +let_actualizacion(entrada actualizacion : Integer) +get_cantidad_aportacion() : Integer +let_cantidad_aportacion(entrada aportacion : Integer) +get_cantidad_eventos() : Integer +let_cantidad_eventos(entrada eventos : Integer) +get_horas() : Integer +let_horas(entrada horas : Integer)

*Tabla 4.3.1.2.1.4 Clase Dros\_Especial*

#### 5.- CDsc

De esta clase se crearan los objetos “*dsc*”, los cuales guardaran los atributos de la evaluación especial del DSC.

SICEB::CDsc_Especial
-intid : Integer -str_semestre : String -str_atendidos : String
+get_id() : Integer +let_id(entrada id : Integer) +get_semestre() : String +let_semestre(entrada semestre : String) +get_atendidos() : String +let_atendidos(entrada atendidos : String)

Tabla 4.3.1.2.1.5 Clase Dsc\_Especial

6.- colEvaluación

Esta colección contendrá una serie (colección) de evaluaciones ya que un becario puede tener muchas de ellas.

7.- CBecario\_Evaluación

Esta clase contiene como atributos, objetos de otras clases esto se realiza para tener un encapsulamiento de cada becario con su colección de evaluaciones, esta clase es un llamado objeto compuestos.

SICEB::CBecario_Evaluacion
+becario : cbecario
+evaluacion : cevaluacion
+colEvaluacion : colEvaluacion
+coldros : colDros_Especial
+coldsc : colDsc_Especial
-Class_Initialize()
-Class_Terminate()

Tabla 4.3.1.2.1.6 Clase Becario\_Evaluación

8.- Cperiodo

Esta clase sirve para guardar las fechas del periodo que se esta evaluando, sus atributos son:

SICEB::CPeriodo
-intDia_inicio : Integer
-intDia_fin : Integer
-intMes_inicio : Integer
-intMes_fin : Integer
-intHabiles : Integer
+get_dia_inicio() : Integer
+let_dia_inicio(entrada dia_inicio : Integer)
+get_dia_fin() : Integer
+let_dia_fin(entrada dia_fin : Integer)
+get_mes_inicio() : Integer
+let_mes_inicio(entrada mes_inicio : Integer)
+get_mes_fin() : Integer
+let_mes_fin(entrada mes_fin : Integer)
+get_habiles() : Integer
+let_habiles(entrada habiles : Integer)

Tabla 4.3.1.2.1.7 Tabla Periodo

## 9.- colBecario\_Evaluación

Esta colección contendrá a todos los objetos del tipo Becario\_Evaluación.

```
+Add(entrada becario : cBecario, entrada coldros : colDros_Especial, entrada colEvaluacion : colEvaluacion, entrada evaluacion : cevaluacion, entrada coldsc : colDsc_Especial, entrada sKey : String) : CBecario_Evaluacion
+get_Item(entrada vntIndexKey : Variant) : CBecario_Evaluacion
+get_Count() : Long
+Remove(entrada vntIndexKey : Variant)
+get_NewEnum() : IUnknown
-Class_Initialize()
-Class_Terminate()
```

*Tabla 4.3.1.2.1.8 Colección Becario Evaluación*

## 10.- cbdBecario

Esta es una clase especial para realizar específicamente la conexión a la base de datos y ejecutar sentencias dentro de ella en concreto obtener los datos necesarios del becario.

## 11.- cbdEvaluación

Esta sirve para realizar la conexión a la base de datos que contiene las evaluaciones, recibiendo como parámetro, el identificador del becario, este proporcionado por la clase cbdBecario.

## 12.- cbdPeriodo

Esta clase de conexión nos sirve para realizar las consultas a la tabla Periodos, en la cual se guardan las fechas de inicio y fin del mismo.

## 13.- colDros\_especial

Sirve para almacenar los objetos Cdros.

## 14.- colDsc Especial

Sirve para almacenar los objetos CDsc.

4.3.1.3.- Diagramas de Actividad

Se utilizan para modelar los procesos que se llevan a cabo en una organización, es parecido a un diagrama de flujo solamente que en cada estado se guarda una actividad.

4.3.1.3.1 Diagrama de actividades del Administrador

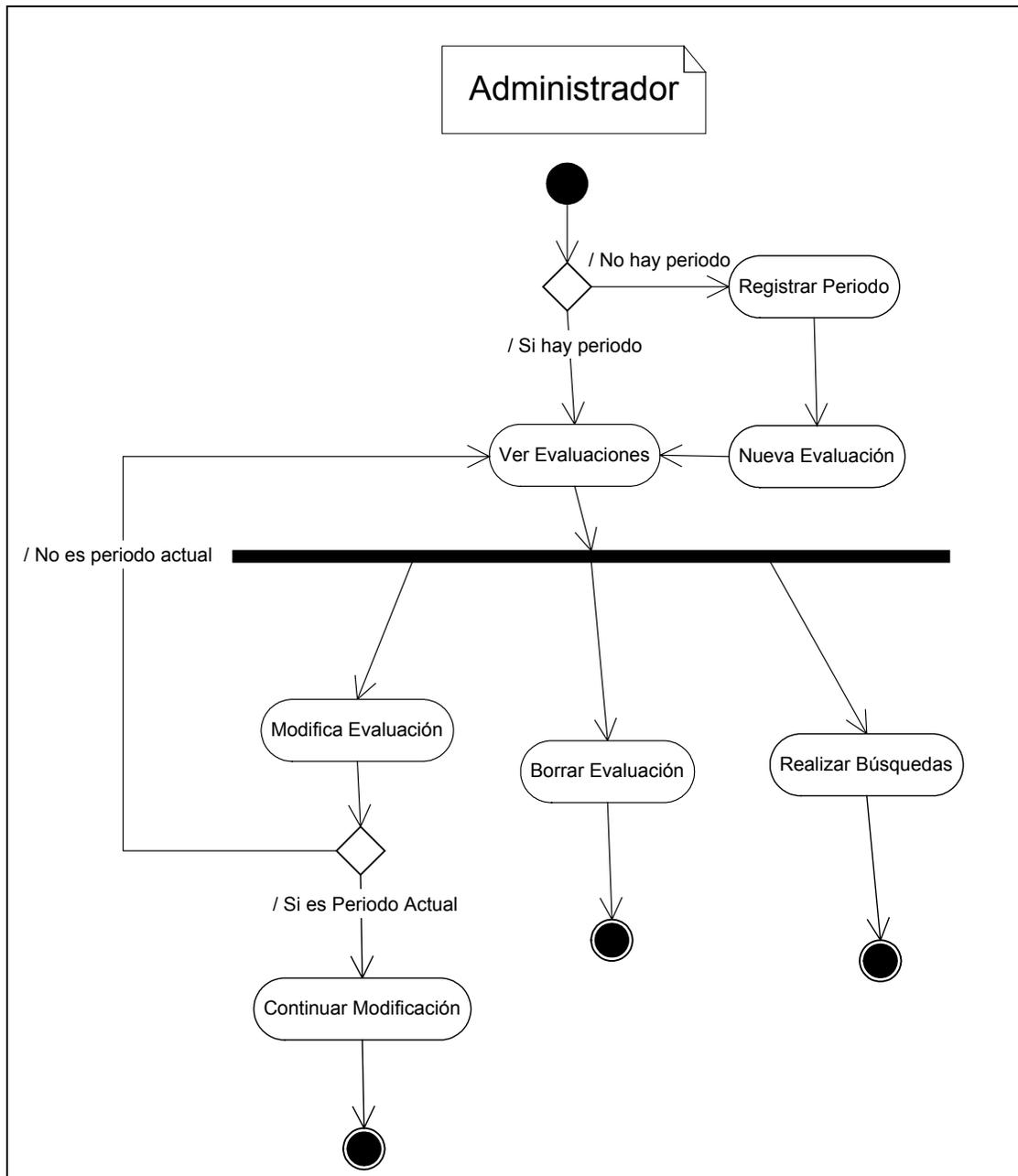


Figura 4.3.1.3.1 Diagrama de Actividades del Administrador

4.3.1.3.2 Diagrama de Actividades del Usuario

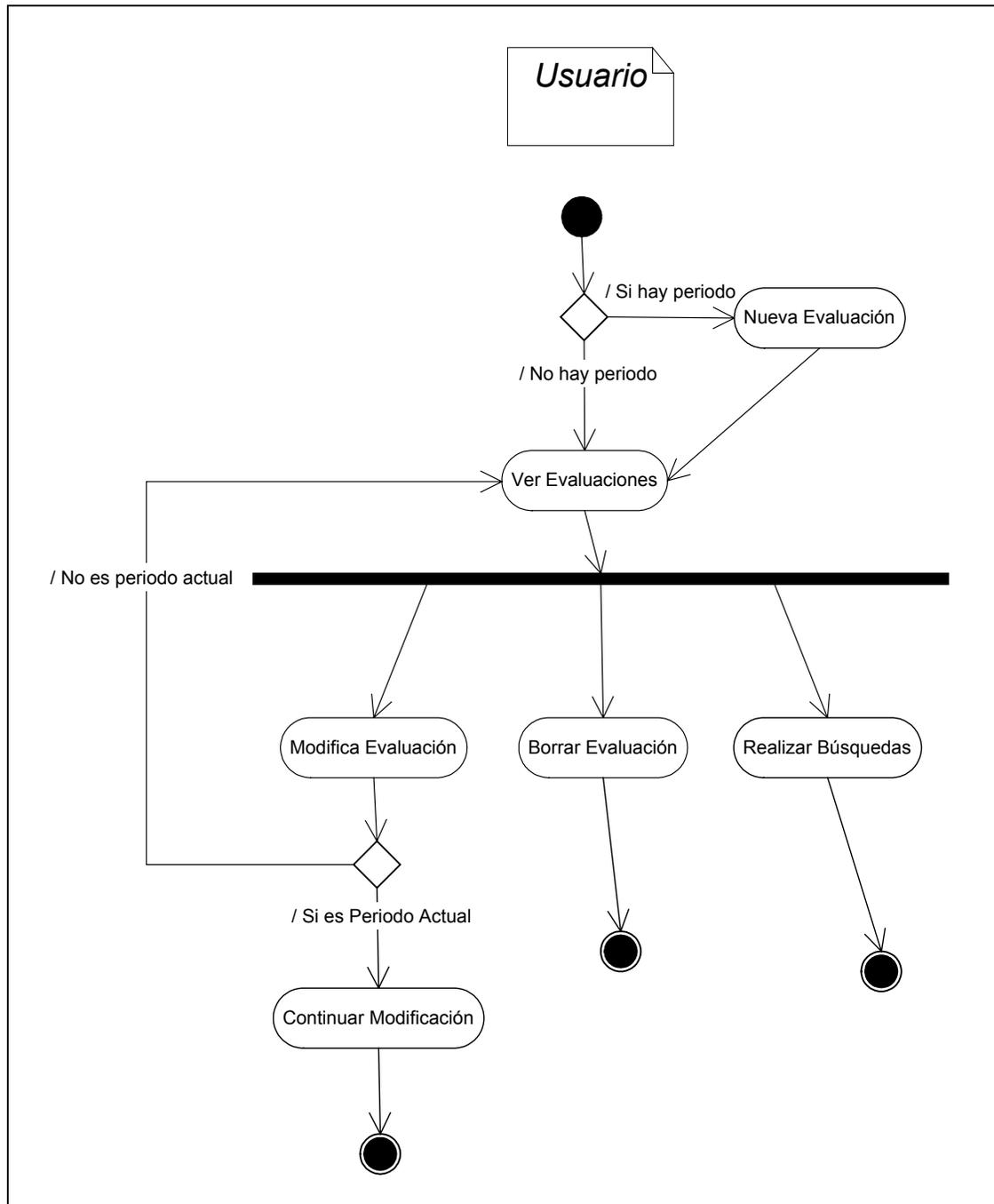


Figura 4.3.1.3.2.1 Diagrama de Actividades del Usuario.

*Pruebas y puesta en marcha  
del SICEB*

---

---

Capítulo

5

## 5.- Pruebas y Puesta en Marcha del SICEB

### 5.1 Las Pruebas de Software

Son los elementos críticos para la garantía de la calidad del software, representan una revisión final de todas las especificaciones, diseño y codificación. Durante los últimos años se ha dado un auge a la creación de pruebas debido a que los costos asociados a un fallo del sistema que se desarrolló son muy altos, en la actualidad muchas empresas emplean entre el 30 y el 40 por ciento del esfuerzo y tiempo total de desarrollo en pruebas.

Durante las etapas de definición y desarrollo del software los desarrolladores construyen un sistema partiendo de conceptos abstractos hasta crear algo tangible que, por lo general entrega un resultado al usuario. Una vez terminadas estas fases, llegan las pruebas, en donde el desarrollador crea una serie de “casos de prueba” que intentan hacer fallar al sistema construido es entonces que la labor del ingeniero se vuelve de constructiva (crear algo) a destructiva (destruir algo).

En las pruebas se requiere saber que fue lo que se construyó o desarrolló para poder plantear situaciones que puedan hacer fallar el sistema. Para esto se deben de imaginar situaciones que por lo general serán improbables que el usuario realice, pero que de hacerlo pueden poner en riesgo la integridad de los datos del sistema e incluso de la organización misma.

Para poder implementar una prueba se deben de tener en cuenta los siguientes puntos:

- Es un proceso que se ejecutará en el sistema con el fin de encontrar un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de encontrar un error no descubierto.
- Una prueba tiene éxito si se descubre un error.

Los datos que se van recabando a medida que se lleva a cabo la prueba nos van mostrando la confiabilidad del software creado y de alguna manera indican la calidad del mismo como un todo. Sin embargo el que se realicen las pruebas no quiere decir que el sistema no tenga ningún fallo solamente nos demuestra que existen defectos en el software.

## ***5.2 Diseño de Casos de Prueba***

Para poder comprobar el software que se desarrolló se tienen dos formas: la primera conociendo la función específica para la que fue diseñado, pudiendo llevar a cabo pruebas que demuestren que cada función que realiza el sistema es completamente operativa al mismo tiempo que se buscan errores en ellas y la segunda es conociendo el funcionamiento del producto, se pueden crear pruebas que aseguren que todas las funciones, módulos, etc. encajen es decir que como un sistema de engranaje todas las piezas que lo conforman funcione a la perfección. El primero de estos enfoques es conocido como prueba de caja negra y el segundo prueba de caja blanca.

## ***5.3 Metodología de las Pruebas***

### ***5.3.1 Pruebas de caja blanca***

Denominada a veces prueba de *caja de cristal* es un método de creación de casos de prueba que usa la estructura de control del diseño procedimental para obtenerlos.

En el desarrollo de la caja blanca, se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos de condiciones y/o *loops* (bucles).

Los casos de prueba garantizan que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo, todas las decisiones lógicas en sus opciones verdadero y falso, todos los bucles hasta sus límites operacionales y las estructuras internas de datos para asegurar su validez.

Las pruebas anteriores se realizan con el fin de probar todos los caminos posibles que una sentencia de evaluación pueda tener, puesto que a menudo se cree que un camino lógico tiene pocas probabilidades de ejecutarse cuando de hecho, lo puede hacer normalmente y también porque cuando escribimos un programa en cualquier lenguaje de programación, no estamos

exentos de tener errores del tipo tipográfico es decir de escritura muchos podrán ser descubiertos por los compiladores de los programas, pero pueden existir algunos que no sean detectados hasta el momento de las pruebas

### *5.3.2 Pruebas de Caja Negra*

Se centra en los requisitos funcionales del software, ésta permite al ingeniero obtener un conjunto de condiciones de entrada que pongan a prueba todos los requerimientos funcionales de un programa

Con esto se buscan errores:

1. De funciones incorrectas o ausentes.
2. De interfaz.
3. En estructuras de datos o acceso a datos externos.
4. De rendimiento.
5. De inicialización y terminación.<sup>1</sup>

La prueba de caja negra, intencionalmente ignora la estructura interna del programa, es decir como fue programado éste centrándose en la funcionalidad y la información que maneja.

## ***5.4 Plan de pruebas del SICEB***

Para diseñarlo se definió que las pruebas requeridas serían básicamente del tipo de caja negra, puesto que se deseaba comprobar que el sistema realizara todo para lo que fue construido. Pruebas de Caja Blanca también fueron realizadas enfocadas a segmentos de código como bucles o condicionales para probar que funcionaban correctamente.

## ***5.5 Puesta en marcha de la Prueba de Caja Negra***

Antes de empezar cualquier prueba de este tipo, se reanalizaron los requerimientos para comprobar que en un principio todo lo mencionado en ellos estaba presente en el sistema con base en esto y en lo que describían que el sistema debía hacer se generó el siguiente plan:

---

<sup>1</sup> Pressman, Roger S. 1998. Ingeniería del Software. México, McGraw-Hill. P.305 195

Primero se comprobó que el sistema respondiera de forma correcta al entrar, esto es comprobando el nombre de usuario y contraseña los cuales en caso de ser erróneos, se deben de volver a pedir evitando así la entrada al sistema de cualquier persona.

Una vez realizado lo anterior se procedió a entrar como administrador esto con la intención de dar de alta un nuevo periodo para evaluar, indicando las fechas de inicio y fin así como los días hábiles del mismo, comprobando que el sistema sólo acepta datos válidos. Además en esta parte se debe de comprobar que antes de que el administrador dé de alta las fechas del periodo los demás usuarios no puedan evaluar a sus becarios.

Luego de esto se procedió a evaluar a los becarios de los diferentes departamentos comprobando que el sistema no permita datos nulos ni incoherentes.

Teniendo evaluaciones de los becarios dentro de la base se comprobaron las funciones de modificar y borrar observando si los resultados eran los esperados; teniendo en cuenta que sólo se modifican aquellas evaluaciones del periodo actual. Para los becarios que tengan una evaluación especial, al ser borrada la general ésta debe ser borrada también para no generar incoherencia de los datos.

## ***5.6 Aplicación del Plan de Pruebas***

### *5.6.1 Pruebas de inicio de sesión*

Se verificó que únicamente se pudiera acceder al sistema mediante un nombre de usuario y contraseña, únicos para cada usuario, como muestran las figuras siguientes:



Figura 5.6.1.1 Iniciando sesión con una contraseña errónea.

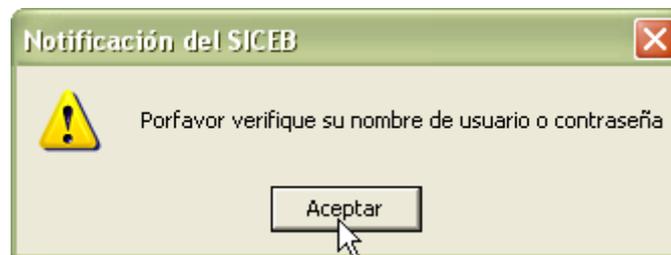


Figura 5.6.1.2 Advertencia puesto que la contraseña que se introdujo no es la correcta para el usuario que esta intentando entrar, al dar clic al botón de aceptar, nos regresara a la pantalla de la figura 5.6.1.1 para volver a introducir la contraseña correcta o bien cambiar de usuario.



Figura 5.6.1.3 Pantalla de bienvenida al sistema, una vez que los datos fueron correctos.

### 5.6.2 Pruebas de llenado del periodo

Una vez verificado que el sistema no permite la entrada a menos que se den el nombre de usuario y contraseña correctos, se procedió a entrar como administrador para poder dar de alta un periodo.

Periodo

Por favor indique las fechas de inicio y término del semestre 2005-2

Inicio

Día: 02

Mes: 02

Año: 2005

Fin

Día:

Mes:

Año:

Por favor indique cuantos días hábiles tuvo este periodo

Días Hábiles

Aceptar

Figura 5.6.2.1 Pantalla para introducir el periodo, se indica el semestre del cuál se solicitan las fechas de inicio y fin, las fechas constan de día, mes y año así como los días hábiles del mismo.

En caso de que se quieran guardar estos datos sin introducir alguno de ellos el sistema mostrará un mensaje de advertencia indicando el campo que falta de ser llenado:

Advertencia SICEB

Indique el día de fin del semestre

Aceptar

Figura 5.6.2.2 Advertencia, en este caso falta especificar el día de fin del semestre

Si los datos fueron llenado completos, pero con errores como muestra la figura 5.6.2.3 se generará una advertencia.

The image shows a web form titled "Periodo" with the instruction "Por favor indique las fechas de inicio y término del semestre 2005-2". It contains two date input sections: "Inicio" (Day: 02, Month: 02, Year: 2005) and "Fin" (Day: 45, Month: 13, Year: 2005). A red arrow points from the "45" in the "Fin" day field to the text "Número mayor al permitido (31)". A modal dialog box is overlaid on the form, titled "SICEB", with a yellow warning icon and the message "El día de fin no debe ser mayor a 31". A red arrow points from this message to the text "Advertencia sobre el número de día de fin del periodo". The dialog box has two "Aceptar" buttons: one with a green checkmark and one without.

Figura 5.6.2.3 Pantalla con datos erróneos o inválidos, mensaje de advertencia generado por tal razón.

Cuando los datos que se introducen son válidos la pantalla antes de aceptar para guardarlos se ve como lo muestra la figura 5.6.2.4:

Periodo

Por favor indique las fechas de inicio y término del semestre 2005-2

Inicio	Fin
Día: 02	Día: 14
Mes: 02	Mes: 08
Año: 2005	Año: 2005

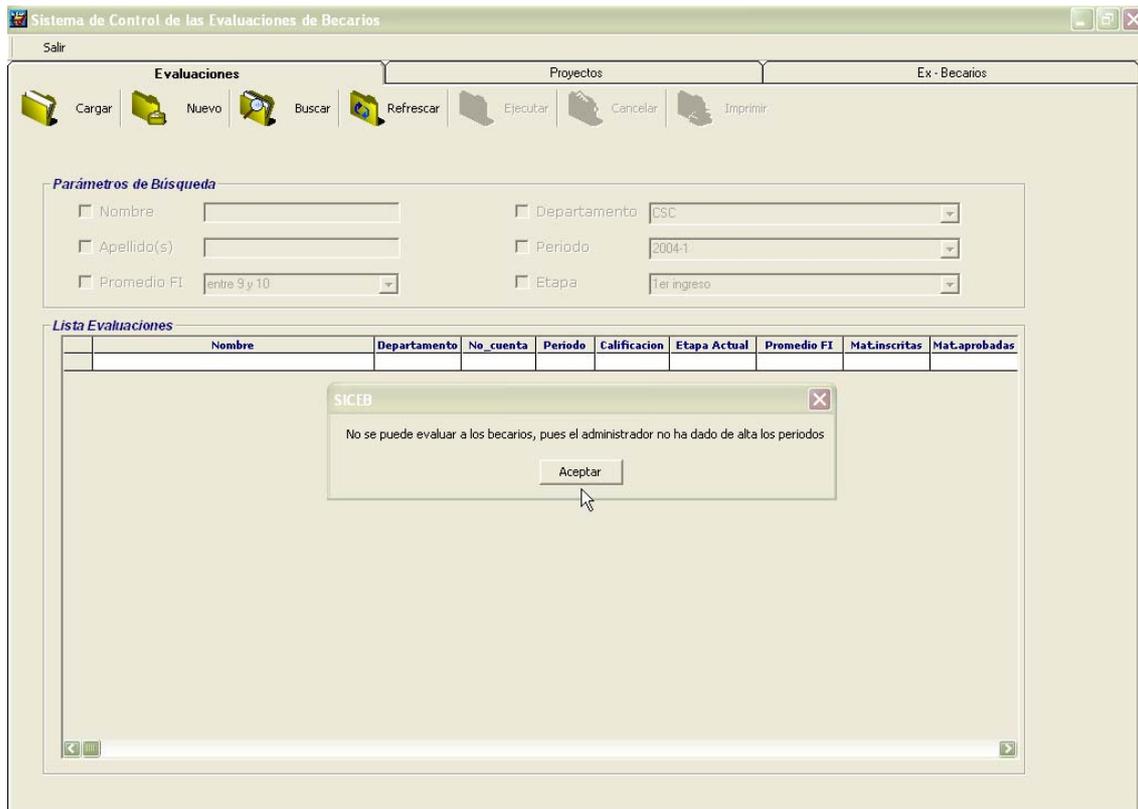
Por favor indique cuantos días hábiles tuvo este periodo

Días Hábiles: 113



Figura 5.6.2.4 Vista de la pantalla del periodo con los datos válidos y coherentes.

Si el administrador del sistema no hubiera dado de alta las fechas del periodo actual los demás usuarios no podrían realizar ninguna nueva evaluación, de intentar hacerlo aparece la advertencia que se muestra en la figura 5.6.2.5



*Figura 5.6.2.5 Al entrar como cualquier otro usuario y no se ha dado de alta el periodo por el administrador, se advierte que no se pueden evaluar a los becarios.*

En caso contrario una vez que se han dado de alta el periodo podemos entrar a la pantalla principal del sistema en este caso se permite ya crear nuevos registros (evaluaciones).

## 5.6.3 Pruebas del cargado de datos

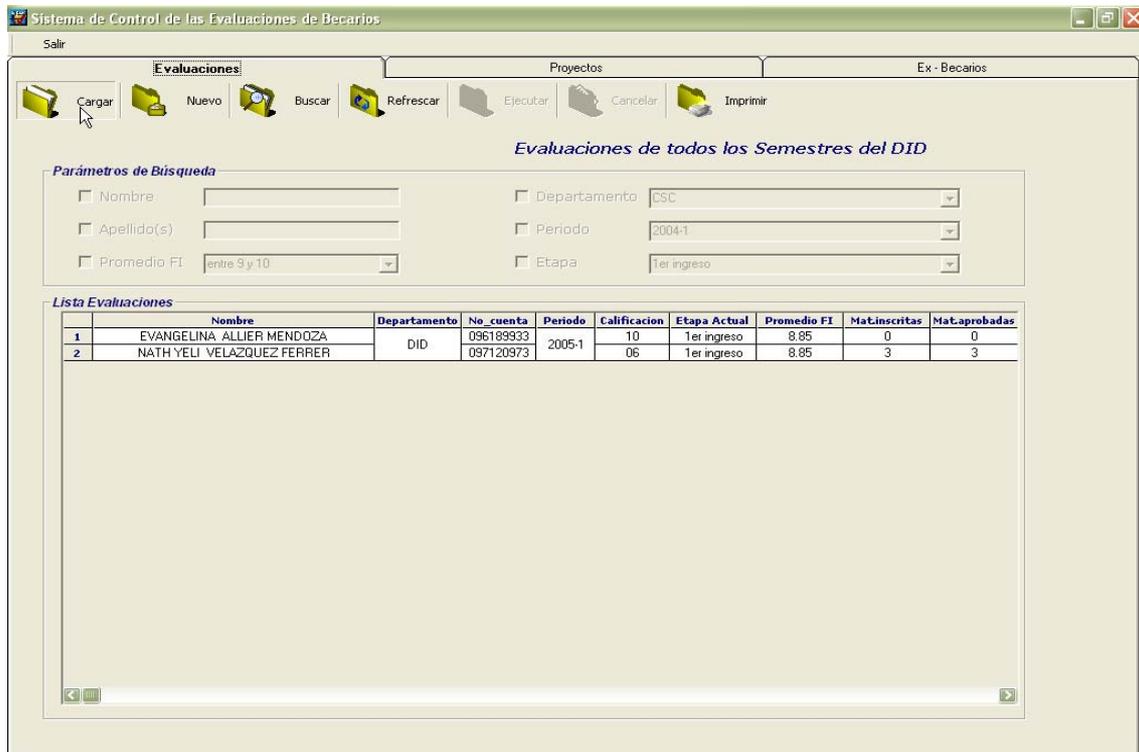


Figura 5.6.3.1 Pantalla principal, luego de solicitar el cargar los datos existentes en la base, para el DID

Como nos muestra la figura anterior sólo se presentan dos registros, eso es porque hasta el momento son los únicos existentes para el usuario que ingresó y aun no ha creado ninguna nueva evaluación (el periodo de pruebas es el 2005-2). Si deseamos conocer los datos completos de la evaluación dando doble clic sobre el nombre del becario nos la mostrará:

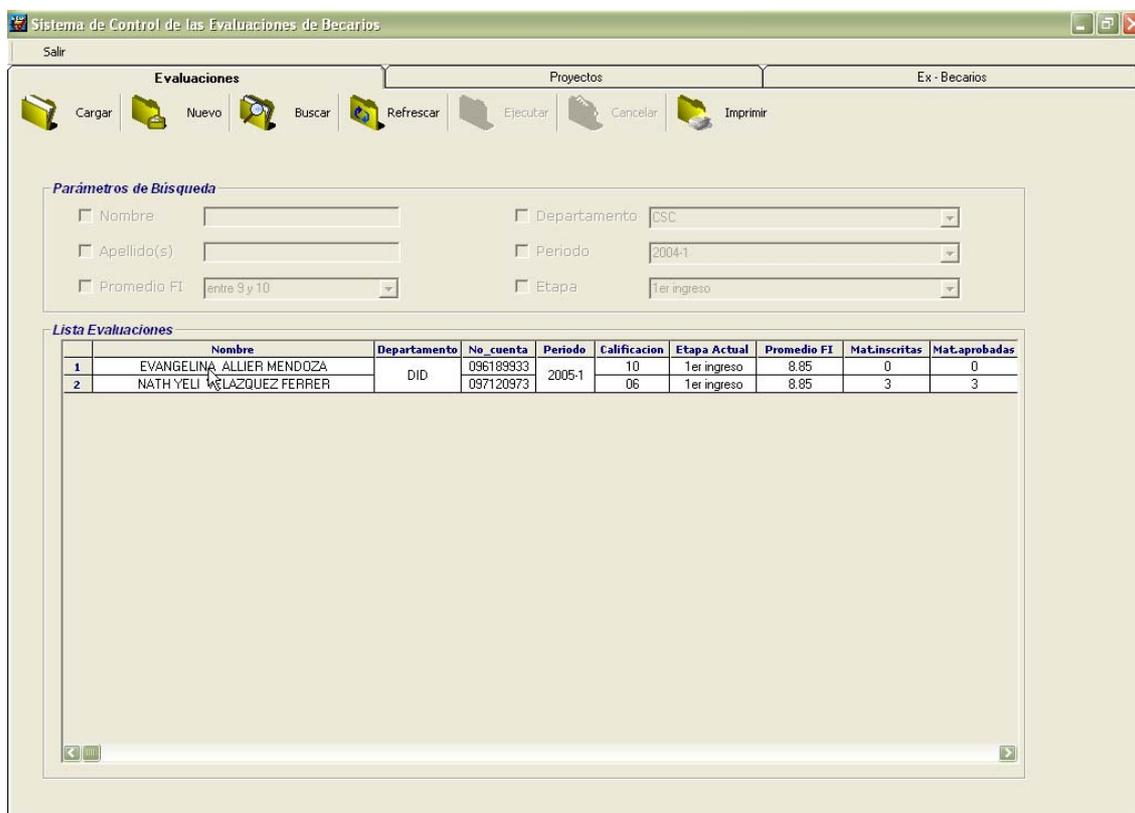


Figura 5.6.3.2 El curso del mouse está posicionado sobre uno de los registros, justo antes de dar doble clic sobre él para que se muestre la evaluación completa.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar | Evaluar | Guardar | Modificar | Refrescar | Borrar | Cancelar | Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8.85

Departamento: DID Semestre que concluyo: 2005-1 Período: 9/8/2004 a 4/2/2004 Evaluación: 10

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001 [Eval.Especial]

**Evaluación**

Rendimiento Académico		Cursos		Proyectos		Asesorías	
Materias inscritas	Termino Creditos	Intersemestrales	1	Lider de Proyectos	0	Asistencia	0
Materias aprobadas	Termino Creditos	Sábados y Domingos	0	Proyectos	0	Retardos	0
Materias reprobadas	Termino Creditos	Internos	0	Oportunidad de Avance	Oportuno	Faltas	0
Estatus:	Inicio Tesis	Especiales	0	Resultados	Oportunos		
		Semestrales	0				
		Cursos Asistidos	0				
		Prebecarios	0				
		Elaboración de Manuales	0				
		Nuevas Propuestas	0				

**Asistencia y Puntualidad**

Asistencia: 0  
Retardos: 0  
Faltas: 113

Nota: los días hábiles para este periodo fueron 113

**Iniciativa**

Disposición para colaborar: Siempre  
Realización de propuestas: Siempre  
Compromiso con UNICA: Siempre

**Calidad en el Trabajo**

Responsabilidad: Siempre  
Confiabilidad: Siempre  
Actitud de Servicio: Siempre

Notificaciones

Comentarios

00

Primero | Anterior | Siguiente | Ultimo

Figura 5.6.3.3 Pantalla de evaluación completa podemos observar que el nombre del becario coincide con el señalado en la figura anterior, se pueden visualizar todos los campos de la evaluación realizada.

Estando en la pantalla que muestra la figura 5.6.3.3 podemos realizar las pruebas para la barra de navegación que se encuentra en la parte inferior esta nos permite movernos entre los registros sin tener que regresar a la pantalla principal si se necesitara visualizar algún otro.

### 5.6.3.1 Pruebas de la barra de navegación

La barra de navegación ubicada en la parte inferior de la pantalla permite desplazarse entre los registros, esto para mayor facilidad y comodidad del usuario de esta forma se pueden recorrer todas las evaluaciones que se hayan cargado en el sistema. Esta barra la podemos observar en la figura 5.6.3.1.1 en la parte inferior de la misma, se ve que el usuario puede desplazarse al primer registro, al último, al anterior y al siguiente.

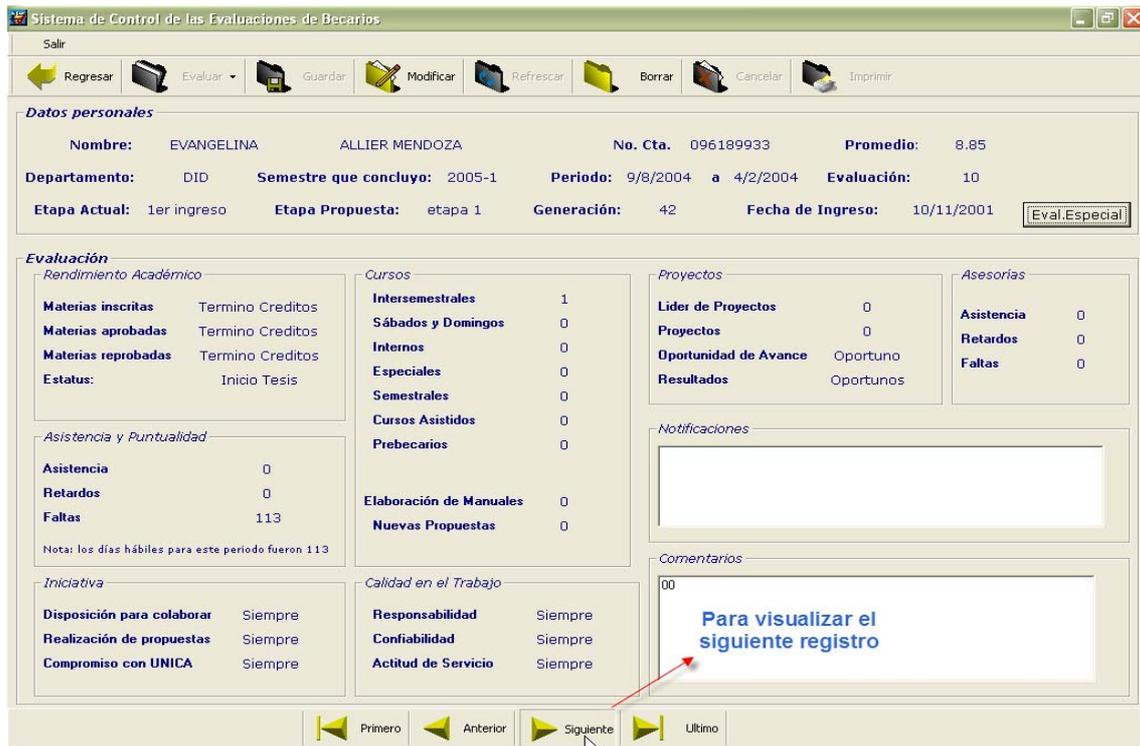


Figura 5.6.3.1.1 Pantalla de evaluación completa, antes de moverse al siguiente registro, el cursor del mouse se encuentra posicionado en este botón.

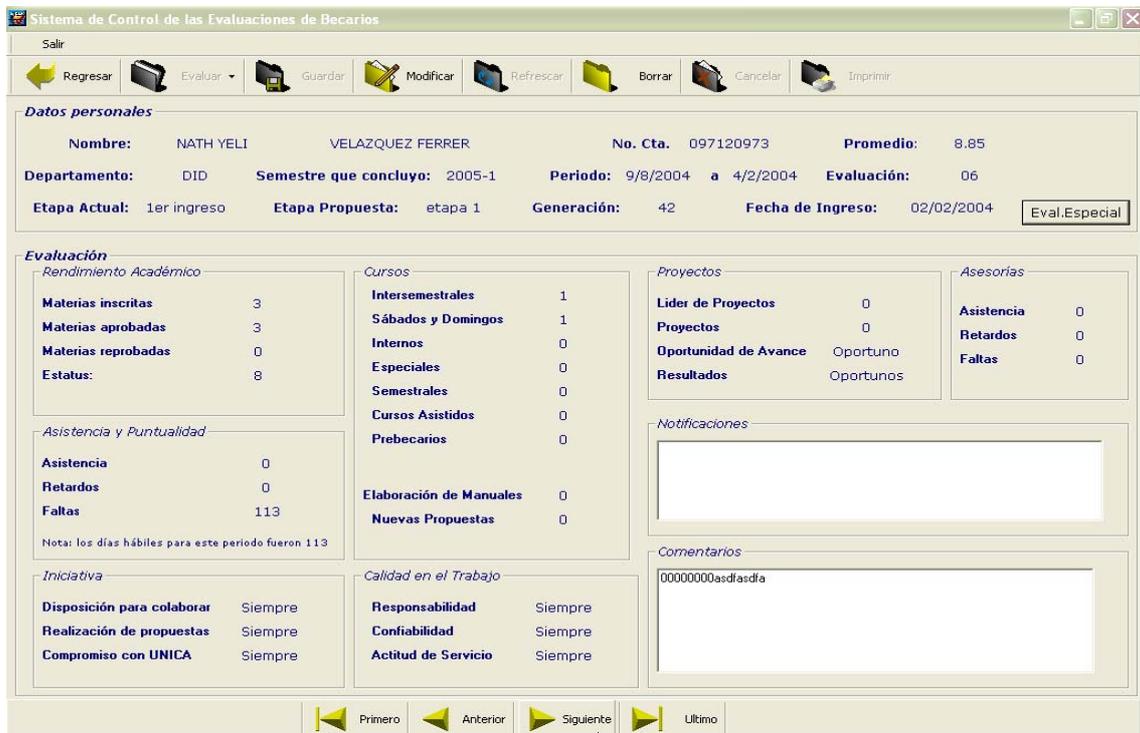


Figura 5.6.3.1.2 Al dar clic al botón de siguiente automáticamente nos trasladamos al registro inmediato, como podemos observar se han cambiado los datos personales del becario, así como los datos de la evaluación, pues se esta visualizando otro registro.

En caso de que se desee seguir visualizando los registros se debe de dar clic al botón de siguiente, sin embargo cuando se llega al final los botones de siguiente y último se deben desactivar mostrando un mensaje de advertencia de que es el último registro. Lo mismo ocurre con los botones de atrás y primero.

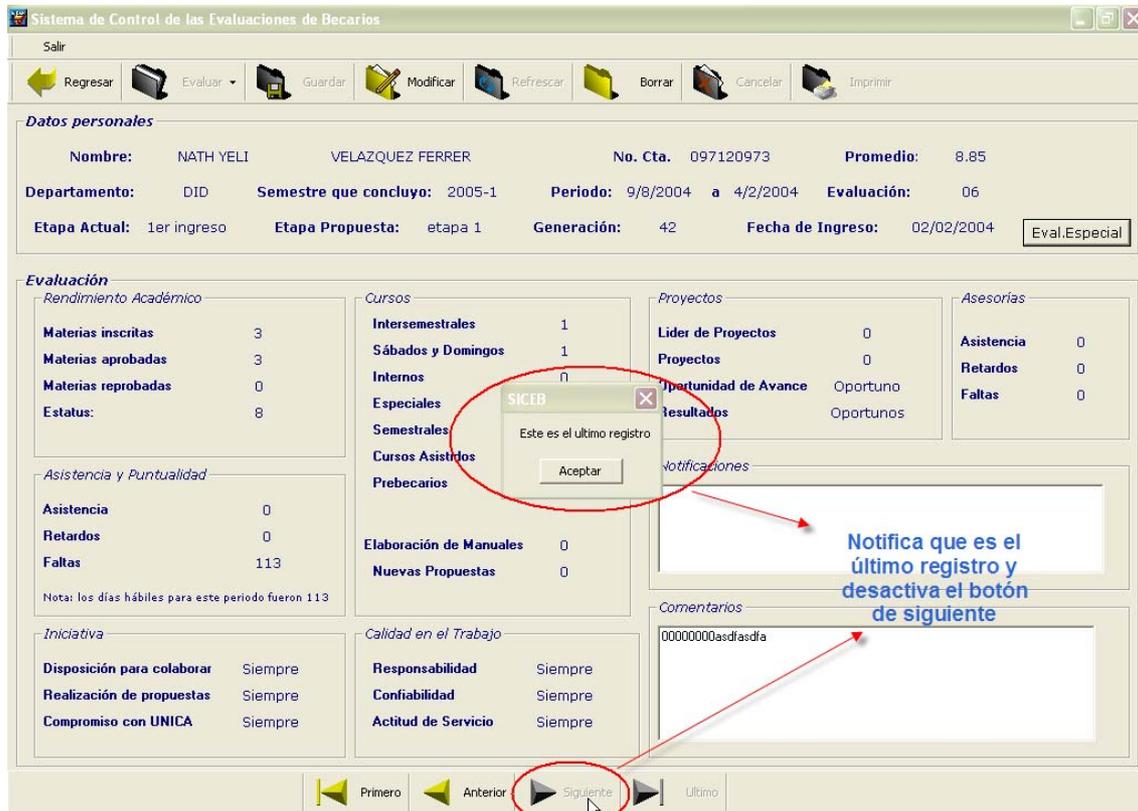


Figura 5.6.3.4.3 Los botones de siguiente y último son desactivados pues se ha llegado al último registro existente en la base de datos.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8.85

Departamento: DID Semestre que concluyo: 2005-1 Periodo: 9/8/2004 a 4/2/2004 Evaluación: 10

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001

**Evaluación**

**Rendimiento Académico**

Materias inscritas Termino Creditos  
Materias aprobadas Termino Creditos  
Materias reprobadas Termino Creditos  
Estatus: Inicio Tesis

**Cursos**

Intersemestrales 1  
Sábados y Domingos 0  
Internos 0  
Especiales 0  
Semestrales 0  
Cursos Asistidos 0  
Prebecarios 0  
Elaboración de Manuales 0  
Nuevas Propuestas 0

**Proyectos**

Lider de Proyectos 0  
Proyectos 0  
Oportunidad de Avance Oportuno  
Resultados Oportunos

**Asesorías**

Asistencia 0  
Retardos 0  
Faltas 0

**Asistencia y Puntualidad**

Asistencia 0  
Retardos 0  
Faltas 113  
Nota: los días hábiles para este periodo fueron 113

**Iniciativa**

Disposición para colaborar Siempre  
Realización de propuestas Siempre  
Compromiso con UNICA Siempre

**Calidad en el Trabajo**

Responsabilidad Siempre  
Confiabilidad Siempre  
Actitud de Servicio Siempre

Comentarios  
00

Primero Anterior Siguiente Ultimo

Este es el primer registro

Aceptar

Notifica que es el primer registro y desactiva el botón de anterior y primero

Figura 5.6.3.4.4 De la misma manera al movernos hacia un registro anterior y llegando al primero, los botones de anterior y siguiente son desactivados, se muestra un mensaje de advertencia.

## 5.6.4 Pruebas para las modificaciones

### 5.6.4.1 Modificación de evaluaciones de periodos anteriores

Cuando se desea modificar un registro se debe de validar que sólo se puedan modificar las evaluaciones del periodo actual no de periodos anteriores, pues de lo contrario se tendrían datos diferentes.

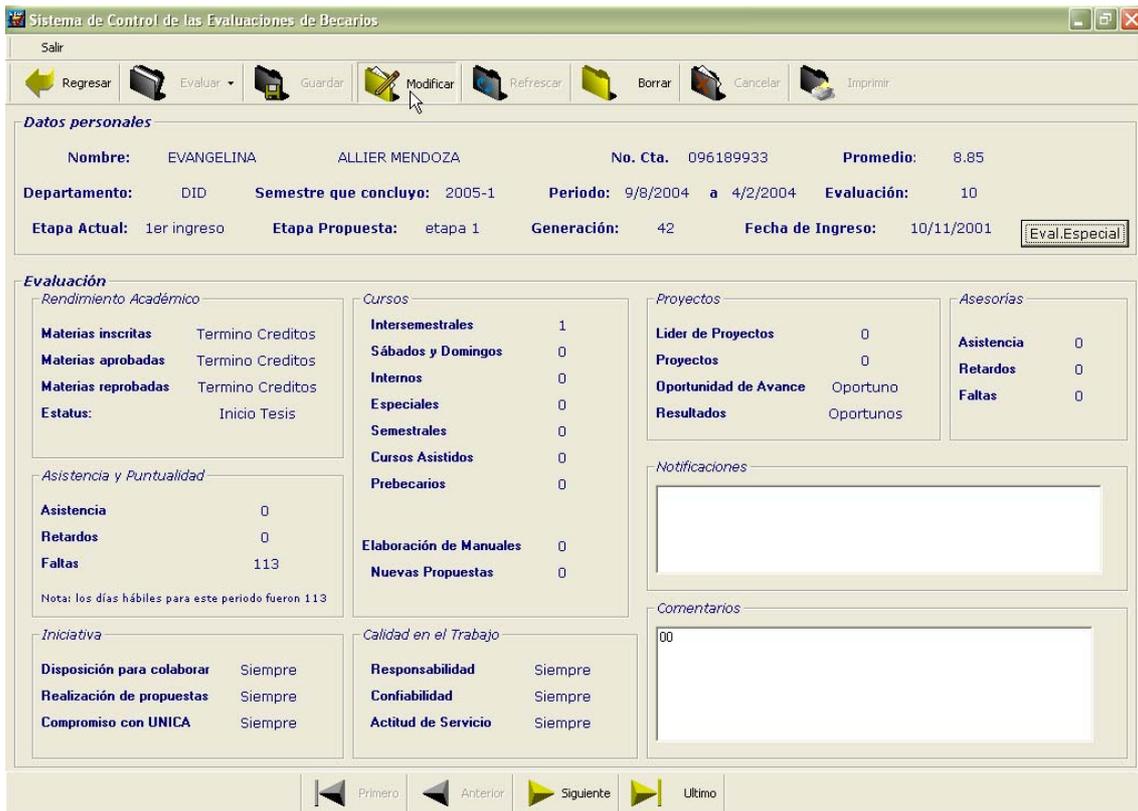


Figura 5.6.4.1.1 Pantalla antes de modificar

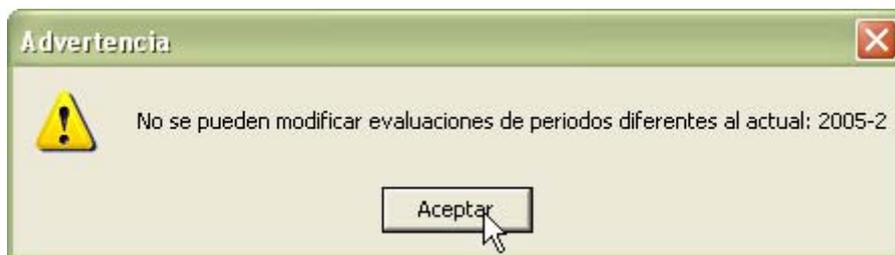


Figura 5.6.4.1.2 Como el periodo mostrado en la figura anterior es 2005-1 y el que se esta evaluando es 2005-2, se advierte que no se pueden modificar evaluaciones pasadas.

### 5.6.4.2 Modificación de evaluaciones del periodo actual.

Para modificar un registro se da clic en la opción de modificar de ésta forma se pasa de la pantalla que muestra la figura 5.6.4.2.1 a la que muestra la figura 5.6.4.2.2

Figura 5.6.4.2.1 Aspecto de la pantalla antes de modificar

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8.55

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 10

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001 Eval.Especial

**Evaluación**

*Rendimiento Académico*

Materias inscritas: Terminó Créditos

Materias aprobadas: Terminó Créditos

Materias reprobadas: Terminó Créditos

Estatus: Inicio Tesis

*Cursos*

Intersemestrales: 1

Sábados y Domingos: 0

Internos: 0

Especiales: 0

Semestrales: 0

Cursos Asistidos: 0

Prebecarios: 1

Elaboración de Manuales: 2

Nuevas Propuestas: 1

*Proyectos*

Lider de Proyectos: 1

Proyectos: 1

Oportunidad de Avance: Oportuno

Resultados: Oportunos

*Asesorías*

Asistencia: 60

Retardos: 1

Faltas: 1

*Asistencia y Puntualidad*

Asistencia: 110

Retardos: 1

Faltas: 3

Nota: los días hábiles para este periodo fueron 113

*Iniciativa*

Disposición para colaborar: Siempre

Realización de propuestas: Siempre

Compromiso con UNICA: Siempre

*Calidad en el Trabajo*

Responsabilidad: Siempre

Confiabilidad: Siempre

Actitud de Servicio: Siempre

*Notificaciones*

El becaria alcanzó la etapa máxima

*Comentarios*

Terminando la tesis, próxima a titularse

Primero Anterior Siguiente Ultimo

Figura 5.6.4.2.2 Al modificar, se habilitan las cajas de texto las cuales muestran los datos que se visualizaban esto con el fin de modificar sólo aquellos que se requieran.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8.55

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 10

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001 Eval.Especial

**Evaluación**

*Rendimiento Académico*

Materias inscritas: Terminó Créditos

Materias aprobadas: Terminó Créditos

Materias reprobadas: 0

Estatus: Inicio Tesis

*Cursos*

Intersemestrales: 1

Sábados y Domingos: 0

Internos: 0

Especiales: 0

Semestrales: 0

Cursos Asistidos: 0

Prebecarios: 1

Elaboración de Manuales: 2

Nuevas Propuestas: 1

*Proyectos*

Lider de Proyectos: 1

Proyectos: 1

Oportunidad de Avance: Oportuno

Resultados: Oportunos

*Asesorías*

Asistencia: 60

Retardos: 1

Faltas: 2

*Asistencia y Puntualidad*

Asistencia: 110

Retardos: 1

Faltas: 3

Nota: los días hábiles para este periodo fueron 113

*Iniciativa*

Disposición para colaborar: Algunas veces

Realización de propuestas: Algunas veces

Compromiso con UNICA: Muchas veces

*Calidad en el Trabajo*

Responsabilidad: Algunas veces

Confiabilidad: Siempre

Actitud de Servicio: Pocas veces

*Notificaciones*

Se modificó

*Comentarios*

Terminando la tesis, próxima a titularse

Primero Anterior Siguiente Ultimo

Figura 5.6.4.2.3 Se modificaron los datos de la evaluación, como lo son la iniciativa y las faltas a asesoría

Al guardar una evaluación modificada aparece un aviso de que se debe refrescar la consulta para ver los cambios realizados:

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar | Evaluar | Guardar | Modificar | Refrescar | Borrar | Cancelar | Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8,55

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 10

Etapa Actual: Ter ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001 Eval.Especial

**Evaluación**

**Rendimiento Académico**

Materias inscritas: Terminó Créditos

Materias aprobadas: Terminó Créditos

Materias reprobadas: 0

Estatus: Inicio Tesis

**Asistencia y Puntualidad**

Asistencia: 110

Retardos: 1

Faltas: 3

Nota: los días hábiles para este periodo fueron 113

**Iniciativa**

Disposición para colaborar: Algunas veces

Realización de propuestas: Algunas veces

Compromiso con UNICA: Muchas veces

**Cursos**

Intersemestrales: 1

Sábados y Domingos: 0

Internos: 0

Especiales: 0

Semestrales: 0

Cursos Asistencia: 0

Prebecarios: 0

Elaboración de Manuales: 2

Nuevas Propuestas: 1

**Proyectos**

Lider de Proyectos: 1

Proyectos: 1

Modalidad de Avance: Oportuno

Retardos: 1

Faltas: 2

**Asesorías**

Asistencia: 60

Retardos: 1

Faltas: 2

**Comentarios**

Terminando la tesis, próxima a titularse

Primero | Anterior | Siguiete | Ultimo

Figura 5.6.4.2.4 Se solicita refrescar la consulta para ver los cambios

Al refrescar visualizamos el registro con los cambios realizados:

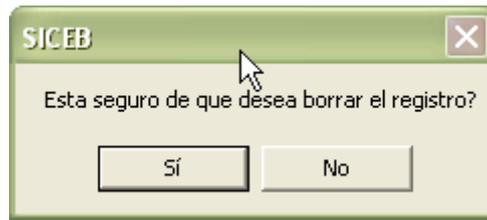
Figura 5.6.4.2.5 Los cambios realizados al a evaluación han quedado realizados.

En caso de que no se desee realizar una modificación, estando en la pantalla del a figura 5.6.4.2.3, si cancelamos regresamos a la pantalla original de visualización; figura 5.6.4.2.1.

### 5.6.5 Pruebas para el borrado de registros

#### 5.6.5.1 Borrado de un solo registro

Para borrar un registro basta dar clic al botón de ese nombre, al realizarlo se deberá confirmar la decisión esto por precaución ya que muchas veces de forma accidental se llegan a presionar botones que no se deseaba si esto ocurriera y no existe una confirmación se borraría un registro que quizá no se debía eliminar, el mensaje es el que muestra la figura siguiente:



*Figura 5.6.5.1.1 Confirmación del borrado de un registro.*

En caso de que se elegía la opción de “No”, regresaremos a la pantalla la figura 5.6.4.2.5, si se elige “Si” el registro es borrado de la base de datos; si la evaluación cuenta con una evaluación especial también será borrada automáticamente y nos mostrara el siguiente aviso:



*Figura 5.6.5.1.2 Mensaje de notificación de borrado de registro, para ver los cambios se debe de Refrescar la consulta.*

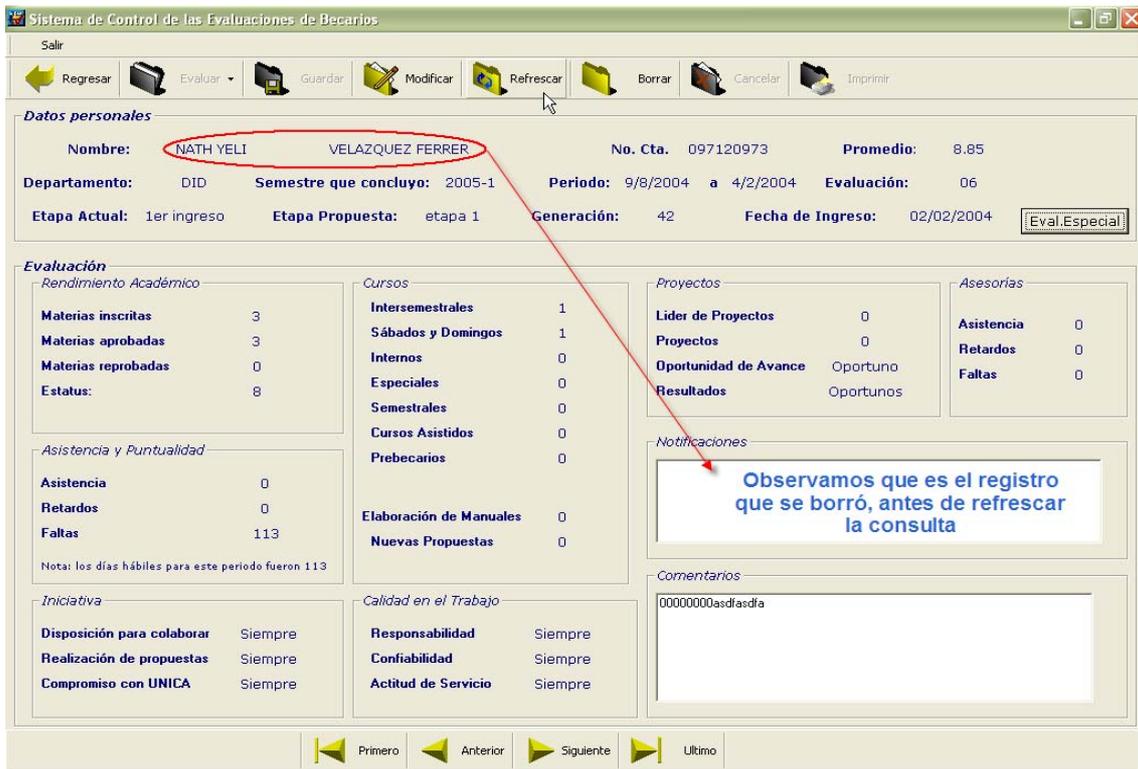


Figura 5.6.5.1.3 Se observa que ya se borró el registro pero sigue apareciendo en pantalla, para poder ver los cambios se debe de Refrescar la consulta.

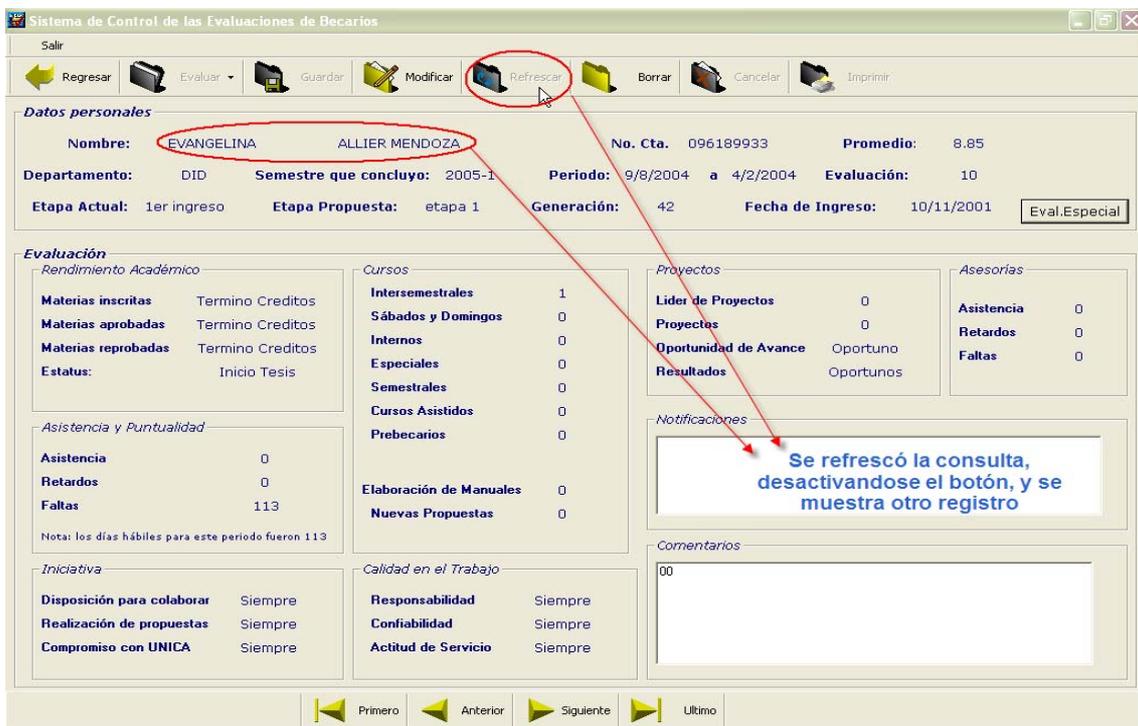


Figura 5.6.5.1.4 Luego de que se refresca la consulta, el registro borrado ya no aparece, en su lugar queda uno de los que existen.

Estando aquí podemos regresar a la pantalla principal con el botón de ese nombre:

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: EVANGELINA ALLIER MENDOZA No. Cta. 096189933 Promedio: 8.85

Departamento: DID Semestre que concluyo: 2005-1 Período: 9/8/2004 a 4/2/2004 Evaluación: 10

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 42 Fecha de Ingreso: 10/11/2001

**Evaluación**

**Rendimiento Académico**

Materias inscritas	Termino	Creditos
Materias aprobadas	Termino	Creditos
Materias reprobadas	Termino	Creditos
Estatus:	Inicio Tesis	

**Cursos**

Intersemestrales	1
Sábados y Domingos	0
Internos	0
Especiales	0
Semestrales	0
Cursos Asistidos	0
Prebecarios	0
Elaboración de Manuales	0
Nuevas Propuestas	0

**Proyectos**

Lider de Proyectos	0
Proyectos	0
Oportunidad de Avance	Oportuno
Resultados	Oportunos

**Asesorías**

Asistencia	0
Retardos	0
Faltas	0

**Asistencia y Puntualidad**

Asistencia	0
Retardos	0
Faltas	113

Nota: los días hábiles para este periodo fueron 113

**Iniciativa**

Disposición para colaborar	Siempre
Realización de propuestas	Siempre
Compromiso con UNICA	Siempre

**Calidad en el Trabajo**

Responsabilidad	Siempre
Confiability	Siempre
Actitud de Servicio	Siempre

**Notificaciones**

**Comentarios**

00

Primero Anterior Siguiete Ultimo

Figura 5.6.5.1.5 Estando en la pantalla de la evaluación completo, en cualquier momento podemos regresar a la pantalla principal de sistema.

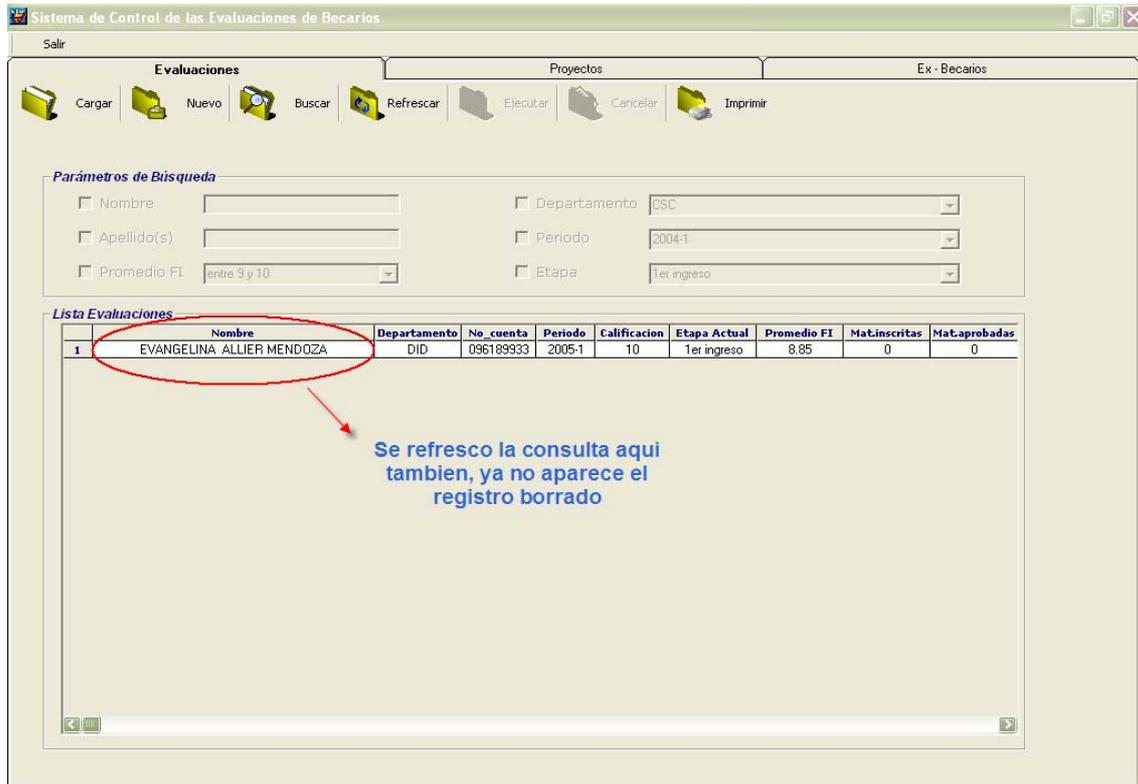


Figura 5.6.5.1.6 Se regreso a la pantalla principal, se puede observar que aquí también se refrescó la consulta mostrando ahora un solo registro pues el otro fue eliminado.

### 5.6.5.2 Borrado de todas las evaluaciones

Para borrar todos los registros existentes se pidió en los requerimientos que se borrara uno por uno, como se vio en las pruebas de borrado se solicita una confirmación y si esta es afirmativa el registro es borrado debiéndose de refrescar la consulta si se llegan a borrar todos marcará al refrescar que no existen registros en la base de datos; al aceptar este mensaje nos regresará a la pantalla principal del sistema.

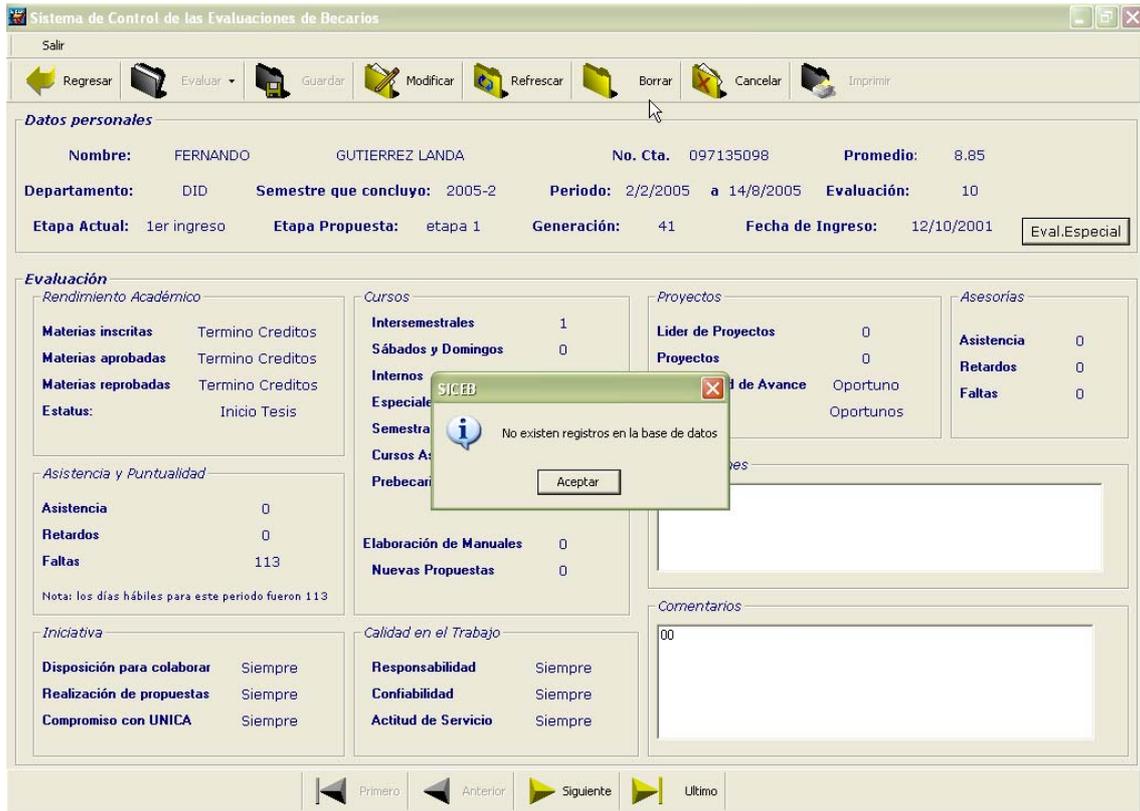
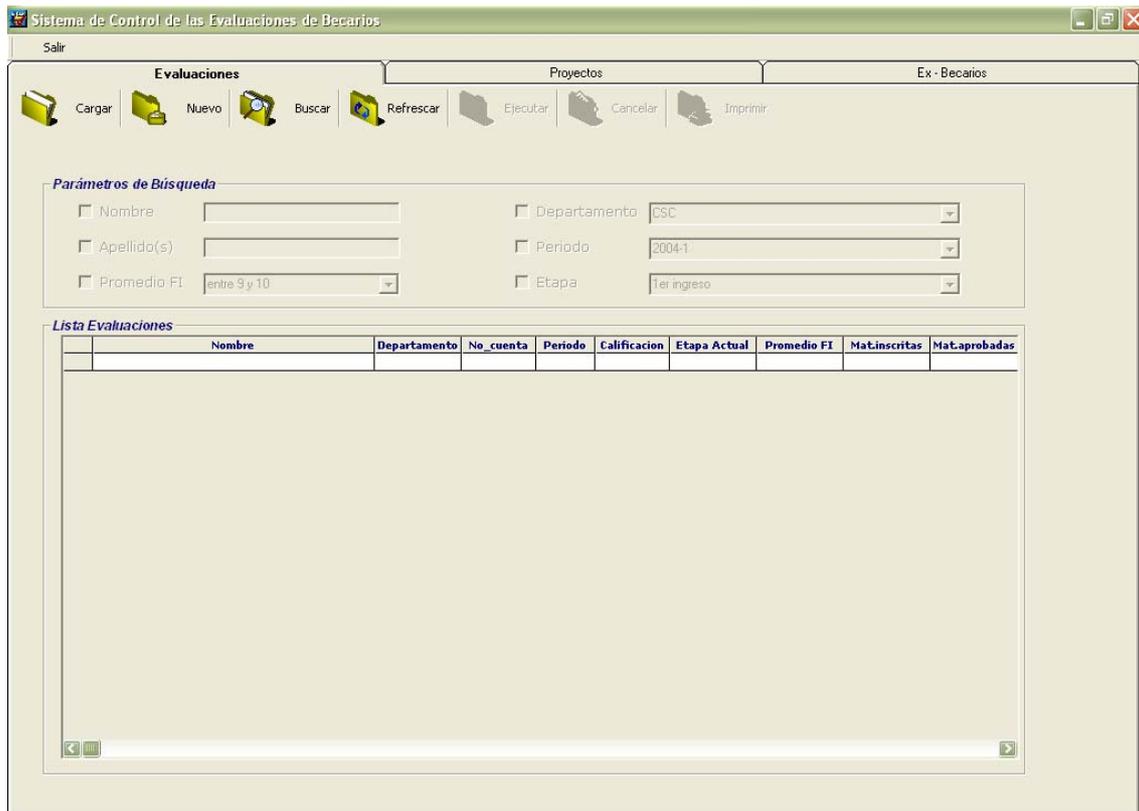


Figura 5.6.5.2.1 No existen registros en este caso porque todos fueron eliminados



*Figura 5.6.5.2.2 Como no existen registros el sistema nos regresa a la pantalla principal.*

### *5.6.6 Pruebas para la creación de una nueva evaluación*

Para crear una nueva evaluación estando en la pantalla principal del sistema (figura 5.6.5.2.2), le damos clic a nuevo y se presentarán todos los becarios que falten de ser evaluados para el departamento que haya ingresado al sistema.

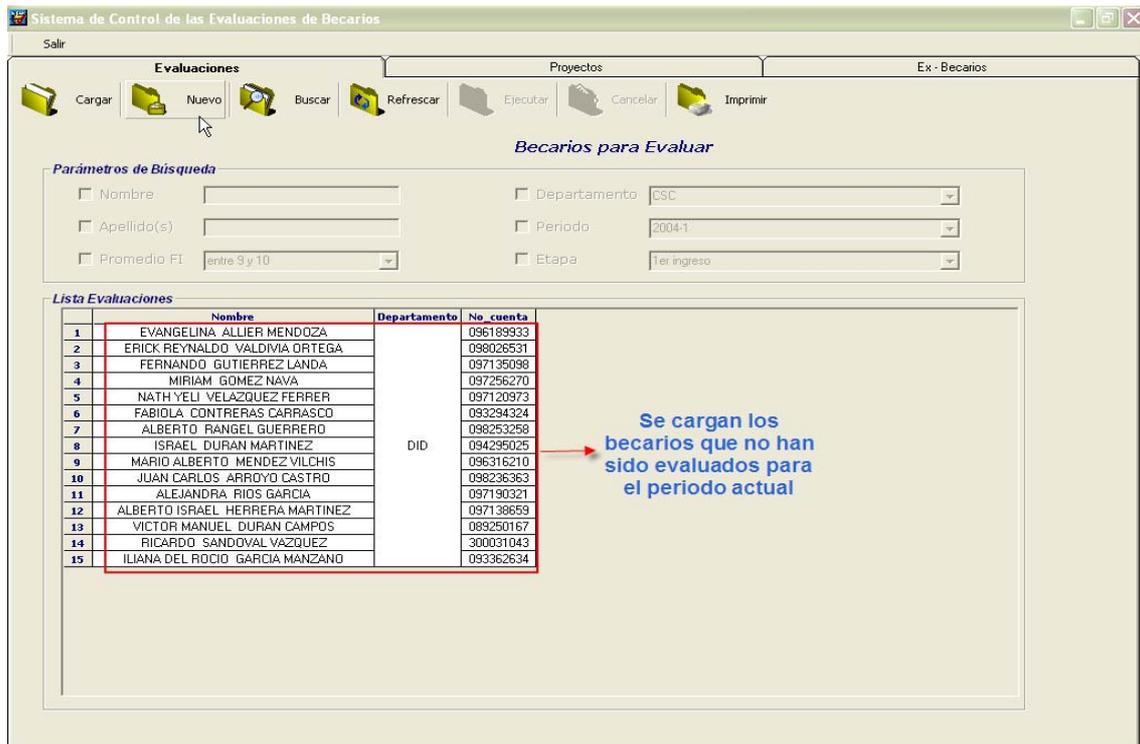


Figura 5.6.6.1 Se muestran en pantalla los becarios que no han sido evaluados pudiendo elegir alguno de ellos para evaluarlo o bien todos. Para poder evaluar basta con colocarse en su nombre y dar doble clic..

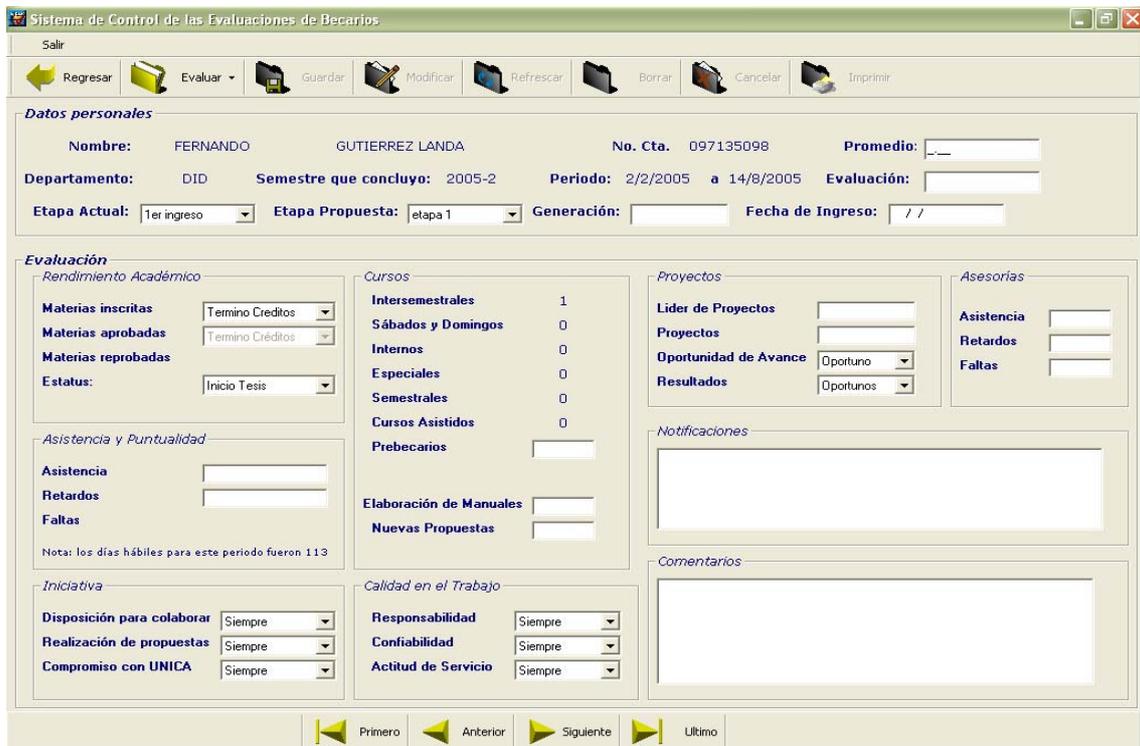


Figura 5.6.6.2 Se seleccionó evaluar a un becario automáticamente el sistema nos envía a la pantalla de evaluación completa pero con la diferencia de que se ven cajas de texto en los campos que deben ser llenados por el usuario.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: FERNANDO GUTIERREZ LANDA No. Cta. 097135098 Promedio: 7.32

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 60

Etapa Actual: etapa 3 Etapa Propuesta: etapa 3 Generación: 41 Fecha de Ingreso: 2001/10/12

**Evaluación**

*Rendimiento Académico*

Materias inscritas: 6  
Materias aprobadas: 2  
Materias reprobadas: 10  
Semestre: 10

*Cursos*

Intersemestrales: 1  
Sábados y Domingos: 0  
Internos: 0  
Especiales: 0  
Semestrales: 0  
Cursos Asistidos: 0  
Prebecarios: 1  
Elaboración de Manuales: 2  
Nuevas Propuestas: 0

*Proyectos*

Lider de Proyectos: 1  
Proyectos: 2  
Oportunidad de Avance: Regular  
Resultados: Mínimo

*Asesorías*

Asistencia: 60  
Retardos: 2  
Faltas: 3

*Asistencia y Puntualidad*

Asistencia: 100  
Retardos: 2  
Faltas: 2

Nota: los días hábiles para este periodo fueron 113

*Iniciativa*

Disposición para colaborar: Algunas veces  
Realización de propuestas: Algunas veces  
Compromiso con UNICA: Pocas veces

*Calidad en el Trabajo*

Responsabilidad: Muchas veces  
Confiabilidad: Siempre  
Actitud de Servicio: Pocas veces

Notificaciones

Comentarios

Primero Anterior Siguiente Ultimo

Figura 5.6.6.3 Una vez que se llenan los datos, este es el aspecto de la pantalla de evaluación, el campo de comentarios es opcional, el de notificaciones será llenado en caso de que el becario haya incurrido en alguna falta.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: FERNANDO GUTIERREZ LANDA No. Cta. 097135098 Promedio: 7.32

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 10

Etapa Actual: etapa 3 Etapa Propuesta: etapa 3 Generación: 41 Fecha de Ingreso: 2001/10/12

**Evaluación**

*Rendimiento Académico*

Materias inscritas: 6  
Materias aprobadas: 2  
Materias reprobadas: 4  
Semestre: 10

*Cursos*

Intersemestrales: 1  
Sábados y Domingos: 0  
Internos: 0  
Especiales: 0  
Semestrales: 0  
Cursos Asistidos: 0  
Prebecarios: 1  
Elaboración de Manuales: 2  
Nuevas Propuestas: 0

*Proyectos*

Lider de Proyectos: 1  
Proyectos: 2  
Oportunidad de Avance: Regular  
Resultados: Mínimo

*Asesorías*

Asistencia: 60  
Retardos: 2  
Faltas: 3

*Asistencia y Puntualidad*

Asistencia: 100  
Retardos: 2  
Faltas: 13

Nota: los días hábiles para este periodo fueron 113

*Iniciativa*

Disposición para colaborar: Algunas veces  
Realización de propuestas: Algunas veces  
Compromiso con UNICA: Pocas veces

*Calidad en el Trabajo*

Responsabilidad: Muchas veces  
Confiabilidad: Siempre  
Actitud de Servicio: Pocas veces

**Notificaciones**

El becario esta a consideración por el número de materias reprobadas. El becario alcanzó la etapa máxima. El becario esta a consideración por el número de faltas en asesoría. El promedio del becario esta por debajo de lo requerido

**Comentarios**

Se generan notificaciones pues el becario faltó a ciertos puntos

Primero Anterior Siguiente Ultimo

Figura 5.6.6.4 Luego de evaluar, se generan las notificaciones y ya se puede guardar la evaluación

Las notificaciones por faltas del becario son:

- En caso de que su promedio sea menor a 8.
- Haya faltado 3 o más veces a asesoría.
- No se haya realizado cambio de etapa del becario.
- Si reprueba más de una materia.

Si por alguna razón en el momento de evaluar faltara algún dato de llenar se genera un aviso indicando que campo falta, evitando así campos nulos; como se ve en la figura el botón de guardar permanece inactivo hasta que se llenen todos los campos.

Figura 5.6.6.5 Datos incompletos, se indica en este caso que falta indicar el número de cursos a Prebecarios que el becario impartió.

Cuando todos los datos han sido llenados y son correctos se puede proceder a guardar la evaluación; el sistema notificará mediante un aviso que los datos fueron guardados o bien si no lo fueron.

The screenshot shows the 'Sistema de Control de las Evaluaciones de Becarios' interface. A confirmation dialog box titled 'SICEB' is centered on the screen, displaying the message 'Se guardaron los datos' and an 'Aceptar' button. The background form is partially obscured but shows the following data:

**Datos personales:**  
 Nombre: FERNANDO GUTIERREZ LANDA, No. Cta. 097135098, Promedio: 8.45  
 Departamento: DID, Semestre que concluyo: 2005-2, Periodo: 2/2/2005 a 14/8/2005, Evaluación: 10  
 Etapa Actual: etapa 3, Etapa Propuesta: contratado, Generación: 41, Fecha de Ingreso: 2001/10/12

**Evaluación:**

- Rendimiento Académico:** Materias inscritas: 6, Materias aprobadas: 5, Materias reprobadas: 1, Semestre: 10
- Asistencia y Puntualidad:** Asistencia: 100, Retardos: 2, Faltas: 13. Nota: los días hábiles para este periodo fueron 113
- Iniciativa:** Disposición para colaborar: Algunas veces, Realización de propuestas: Algunas veces, Compromiso con UNICA: Pocas veces
- Cursos:** Intersemestrales: 1, Sábados y Domingos: 0, Internos: 0, Semestrales: 0, Cursos Asistidos: 0, Prebecarios: 0
- Elaboración de Manuales:** 2, Nuevas Propuestas: 0
- Calidad en el Trabajo:** Responsabilidad: Muchas veces, Confiabilidad: Siempre, Actitud de Servicio: Pocas veces
- Proyectos:** Lider de Proyectos: 1, Proyectos: 2, Oportunidad de Avance: Regular, Resultados: Mínimo
- Asesorías:** Asistencia: 60, Retardos: 2, Faltas: 2

Navigation buttons at the bottom: Primero, Anterior, Siguiete, Ultimo.

Figura 5.6.6.6 Confirmación de que los datos fueron guardados exitosamente.

De la misma manera que se puede navegar entre los registros para visualizarlos se puede hacer para evaluarlos:

Figura 5.6.6.7 Con la barra de navegación nos movemos entre registros para evaluarlos, las cajas de texto se limpian para hacer una nueva evaluación

#### 5.6.6.1 Pruebas de los campos especiales de la evaluación

Dentro de la evaluación existen dos campos que son generación y fecha de ingreso los cuales sólo son introducidos la primera vez que se evalúa a un becario, como lo muestra la figura siguiente:

Figura 5.6.6.1.1 Los datos de generación y fecha de ingreso son requeridos pues el becario Miriam nunca ha sido evaluado, por lo tanto no se encuentran dentro de la base.

Si el becario ha sido evaluado al menos 1 vez estos datos ya no serán pedidos pues de otra manera sería poco funcional y molesto para el usuario tener que estar recordando cada vez que evalué a un becario la generación a la que pertenece y la fecha en que ingresó a la unidad.

Figura 5.6.6..1.2 Como podemos observar, el becario Nath Yeli ya había sido evaluado al menos una vez, por lo que los datos de generación y periodo ya no son requeridos.

### 5.6.6.2 Pruebas de las evaluaciones especiales

Dentro de los departamentos de Seguridad y Redes se pidió tener una evaluación especial, esto es como complemento a los puntos que se toman en cuenta en la general para esta nueva evaluación se toman en cuenta las mismas funciones que tiene la general: sólo se pueden modificar las evaluaciones especiales del periodo actual y cuando se borra la general se borra la especial. Así también no se puede guardar una evaluación general si no se ha guardado antes la especial esto para seguridad de que la segunda sea realizada y guardada.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: DULCE CAMPOS DEL RAZO No. Cta. 098270002 Promedio: 8.10

Departamento: DROS Semestre que concluyo: 2005-2 Periodo: 7/2/2005 a 12/8/2005 Evaluación: 06

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 2005/03/12

**Evaluación**

*Rendimiento Académico*

Materias inscritas: 4  
Materias aprobadas: 4  
Materias reprobadas: 0  
Semestre: 8

*Asistencia y Puntualidad*

Asistencia: 110  
Retardos: 2  
Faltas: 4  
Nota: los días hábiles para este periodo fueron 114

*Iniciativa*

Disposición para colaborar: Siempre  
Realización de propuestas: Siempre  
Compromiso con UNICA: Siempre

*Cursos*

Intersemestrales: 0  
Sábados y Domingos: 1  
Internos: 0  
Especial: SICEB  
Semestre: [ ]  
Curso: [ ]  
Prebe: [ ]

Elaboración de Manuales: 1  
Nuevas Propuestas: 0

*Proyectos*

Lider de Proyectos: 1  
Proyectos: 1  
Oportunidad de Avance: Oportuno

*Asesorías*

Asistencia: 100  
Retardos: 2  
Faltas: 1

*Calidad en el Trabajo*

Responsabilidad: Siempre  
Confiabilidad: Siempre  
Actitud de Servicio: Siempre

Comentarios: Realizó instalaciones de servidores, y administración en linux

Primero Anterior Siguiente Ultimo

Proyecto1 - Microsoft Visual Basic [ejecutar]

*Figura 5.6.6.2.1 Se ha llenado la evaluación general para un becario del DROS sin embargo en el momento de querer guardarla indica que antes debe realizar la evaluación especial.*

La pantalla de evaluación especial es la que nos muestra la figura 5.6.6.2.2:

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos

Número de Atendidos Parcialmente

Realización de Respaldos de Servidores

Número de respaldos

Cantidad de información en MB respaldada

Número de sistemas para la administración al que le da mantenimiento

Cantidad

Actualización del sistema

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos

Número de horas invertidas

*Figura 5.6.6.2.2 Vista de la pantalla de la evaluación especial del DROS la barra de herramientas es muy parecida a la que se tiene para la evaluación general así como las funciones*

De la misma manera que en la evaluación especial cuando se intente guardar esta evaluación y alguno de los datos falte el usuario será notificado del faltante.

Evaluación Especial del DROS

Regresar Modificar Guardar Cancelar Refrescar

Evaluación DROS

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos  
Número de Atendidos Parcialmente

Realización de Respaldos de Servidores

Número de respaldos  
Cantidad de información en

Número de sistemas para la administración

Cantidad  
Actualización del sistema

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos  
Número de horas invertidas

SICEB

Indique el numero de eventos atendidos totalmente

Aceptar

Figura 5.6.6.2.3 Notificación de dato faltante, esto para prevenir el que se generen datos nulos dentro de la evaluación.

Una vez completados todos los campos se puede guardar este botón se desactiva una vez que son guardados los datos.

Evaluación Especial del DROS

Regresar    Modificar    Guardar    Cancelar    Refrescar

Evaluación DROS

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos	9
Número de Atendidos Parcialmente	5

Realización de Respaldos de Servidores

Número de respaldos	1
Cantidad de información en MB respaldada	2

Número de sistemas para la administración al que le da mantenimiento

Cantidad	3
Actualización del sistema	1

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad	3
----------	---

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos	3
Número de horas invertidas	3

Figura 5.6.6.2.4 Guardado de los datos de la evaluación especial, el botón se desactiva y los datos ya quedaron almacenados en la base de datos.

Una vez guardada la evaluación especial se regresa a la pantalla de la evaluación general y ya se permite guardarla.

Si se regresa a la pantalla principal y se cargan los registro aparecerá la evaluación recién realizada.

Para visualizar la evaluación especial en la pantalla de evaluación general existe un botón con ese nombre, al dar clic sobre él nos lleva a la pantalla de la figura 2.6.6.2.4, sólo que en modo de lectura es decir, sin las cajas de texto activadas.

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos	
Número de Atendidos	9
Número de Atendidos Parcialmente	5

Realización de Respaldos de Servidores	
Número de respaldos	1
Cantidad de información en MB respaldada	2

Número de sistemas para la administración al que le da mantenimiento	
Cantidad	3
Actualización del sistema	1

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo	
Cantidad	3

Número de atención a incidentes de operación de servidores y redes fuera del horario	
Cantidad de Eventos	3
Número de horas invertidas	3

*Figura 5.6.6.2.5 Visualización de la evaluación especial*

Para modificar esta evaluación de la misma manera que en la general el botón modificar nos habilita las cajas de texto, como lo vemos en la figura 5.6.6.2.4

Luego de realizar una modificación se debe de refrescar la consulta para que ésta tenga efecto y sea visualizada en tiempo real; vamos a modificar en el campo de realización de respaldos de servidores el valor del número de respaldos que actualmente tiene el valor de 1 y lo modificaremos por 5.

Evaluación Especial del DROS

Regresar    Modificar    Guardar    Cancelar    Refrescar

Evaluación DROS

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos	<input type="text" value="9"/>
Número de Atendidos Parcialmente	<input type="text" value="5"/>

Realización de Respaldos de Servidores

Número de respaldos	<input type="text" value="1"/>
Cantidad de información en MB respaldada	<input type="text" value="2"/>

Número de sistemas para la administración al que le da mantenimiento

Cantidad	<input type="text" value="3"/>
Actualización del sistema	<input type="text" value="1"/>

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad	<input type="text" value="3"/>
----------	--------------------------------

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos	<input type="text" value="3"/>
Número de horas invertidas	<input type="text" value="3"/>

Figura 5.6.6.2.6 Vista de la pantalla de evaluación especial, antes de modificar el campo

**Evaluación Especial del DROS**

Regresar   Modificar   Guardar   Cancelar   Refrescar

**Evaluación DROS**

**Número de atención de solicitud de eventos de soporte técnico atendidos**

Número de Atendidos	9
Número de Atendidos Parcialmente	5

**Realización de Respaldos de Servidores**

Número de respaldos	5
Cantidad de información en MB respaldada	2

**Número de sistemas para la administración al que le da mantenimiento**

Cantidad	3
Actualización del sistema	1

**Número de Aportaciones al sistema de conocimientos de Administración en Cómputo**

Cantidad	3
----------	---

**Número de atención a incidentes de operación de servidores y redes fuera del horario**

Cantidad de Eventos	3
Número de horas invertidas	3

*Figura 5.6.6.2.7 Vista de la pantalla una vez modificado, antes de guardar*

Una vez que le damos guardar, refrescamos y el aspecto de la pantalla es el siguiente:

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos	
Número de Atendidos	9
Número de Atendidos Parcialmente	5

Realización de Respaldos de Servidores	
Número de respaldos	5
Cantidad de información en MB respaldada	2

Número de sistemas para la administración al que le da mantenimiento	
Cantidad	3
Actualización del sistema	1

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo	
Cantidad	3

Número de atención a incidentes de operación de servidores y redes fuera del horario	
Cantidad de Eventos	3
Número de horas invertidas	3

*Figura 5.6.6.2.7 Modificación realizada en la evaluación especial.*

### 5.6.6.3 Evaluación de un becario que ya contaba con una evaluación

Como se había realizado la evaluación de un becario para comprobar que el sistema funcionara y esta ya se había guardado, al regresar a la pantalla principal observamos que aparecen dos becarios evaluados uno de ellos es la nueva evaluación de Fernando Gutiérrez, si observamos los periodos veremos que una pertenece al semestre 2005-1 y la otra a la del 2005-2 (nueva evaluación).

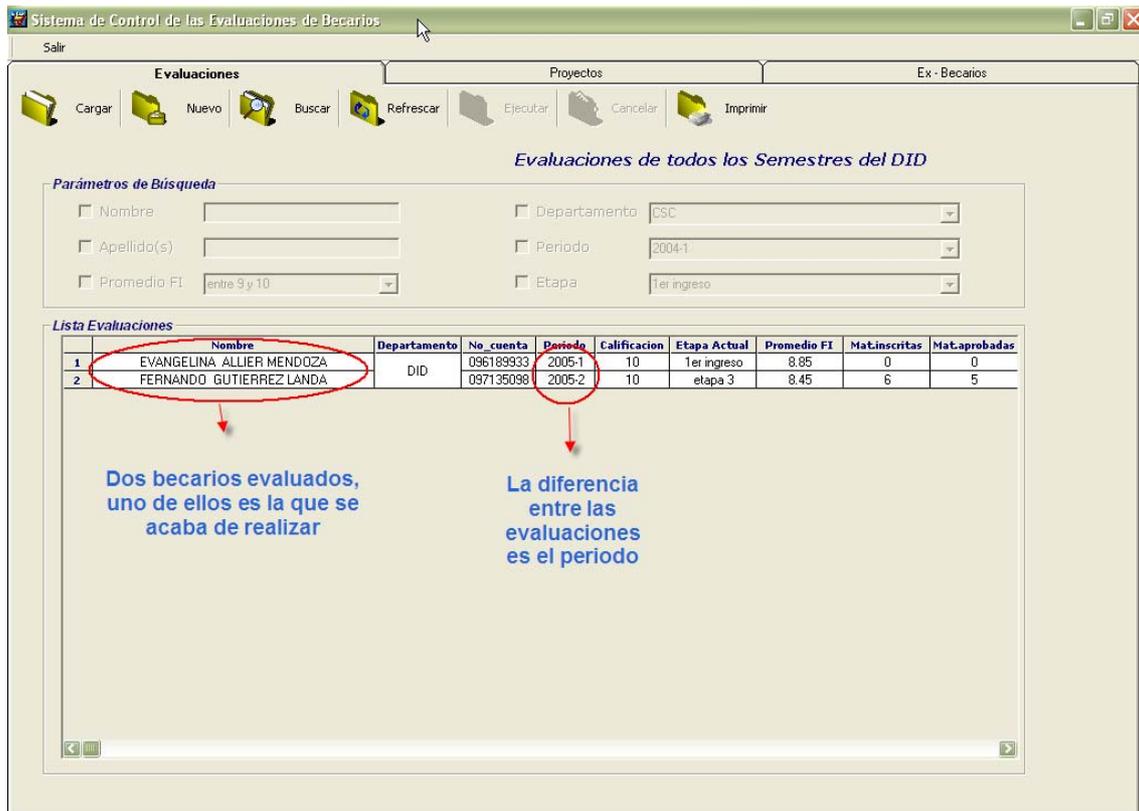


Figura 5.6.6.3.1 Se observa que se tienen dos becarios, uno de ellos es la evaluación que se realizó para verificar que lo requerido para una nueva evaluación funcionara correctamente.

En este caso se evaluó a un becario diferente, pero es lógico que los becarios tendrán más de una evaluación dentro del sistema ya que por especificaciones se va generando un historial para tal caso cuando se tiene un nuevo periodo el o los becarios aparecerán en la lista de nuevas evaluaciones, esta se llevará a cabo de acuerdo a la metodología antes citada y se tendrá que guardar esta nueva evaluación.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: **EVANGELINA ALLIER MENDOZA** No. Cta. 096189933 Promedio: 8.55

Departamento: DID Semestre que concluyo: 2005-2 Periodo: 2/2/2005 a 14/8/2005 Evaluación: 10

Etapa Actual: etapa 3 Etapa Propuesta: etapa 3

**Se evaluó a un becario que tenía una evaluación en el sistema**

**Evaluación**

*Rendimiento Académico*

Materias inscritas Terminó Créditos

Materias aprobadas Terminó Créditos

Materias reprobadas 0

Estatus: No inicio Tesis

*Cursos*

Intersemestrales 1

Sábados y Domingos 0

Internos 0

Especiales 0

Semestrales 0

Cursos Asistidos 0

Prebecarios 1

Elaboración de Manuales 2

Nuevas Propuestas 1

*Proyectos*

Lider de Proyectos 1

Proyectos 1

Oportunidad de Avance Oportuno

Resultados Oportunos

*Asesorías*

Asistencia 60

Retardos 1

Faltas 1

*Asistencia y Puntualidad*

Asistencia 110

Retardos 1

Faltas 3

Nota: los días hábiles para este periodo fueron 113

*Iniciativa*

Disposición para colaborar Muchas veces

Realización de propuestas Siempre

Compromiso con UNICA Algunas veces

*Calidad en el Trabajo*

Responsabilidad Muchas veces

Confiability Muchas veces

Actitud de Servicio Siempre

*Notificaciones*

El becario alcanzó la etapa máxima

*Comentarios*

Terminando la tesis, próxima a titularse

Primero Anterior Siguiete Ultimo

Figura 5.6.6.3.2 Se muestra que se está evaluando al becario Evangelina para el cuál ya existía un registro previo lo cuál no afecta para nada la siguiente evaluación.

**Lista Evaluaciones**

	Nombre	Departamento	No_cuenta	Periodo	Calificación	Etapa Actual	Promedio FI	Mat.Inscritas	Mat.aprobadas
1	EVANGELINA ALLIER MENDOZA	DID	096189933	2005-1	10	1er ingreso	8.85	0	0
2	FERNANDO GUTIERREZ LANDA		097135098	2005-2	10	etapa 3	8.45	6	5

Se tienen dos evaluaciones para un mismo becario

Cambian en los periodos

Figura 5.6.6.3.3 Al evaluar a un mismo becario y cargar los datos observamos que se cargó el registro de manera correcta, ahora Evangelina tiene dos evaluaciones asociadas a ella, una del periodo 2005-1 y la otra del 2005-2

Al igual que se puede navegar entre las evaluaciones de diferentes becarios lo podemos hacer para uno mismo:

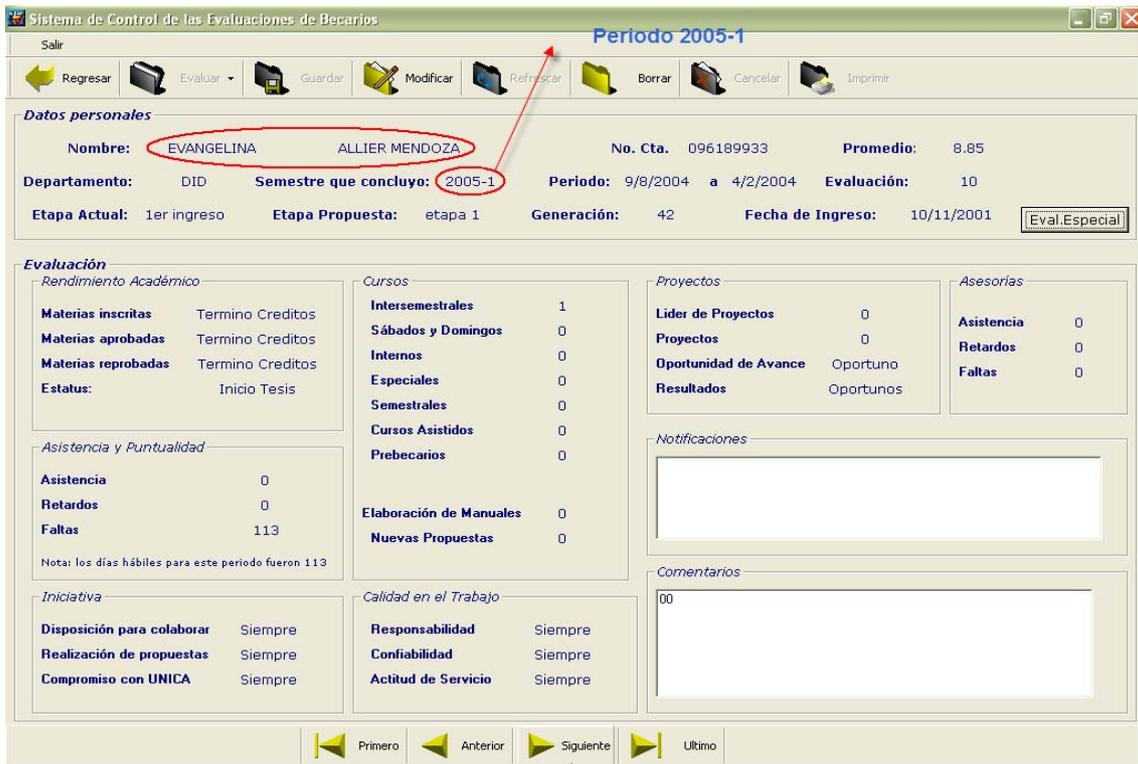


Figura 5.6.6.3.4 Se está visualizando la evaluación del periodo 2005-1 de Evangelina

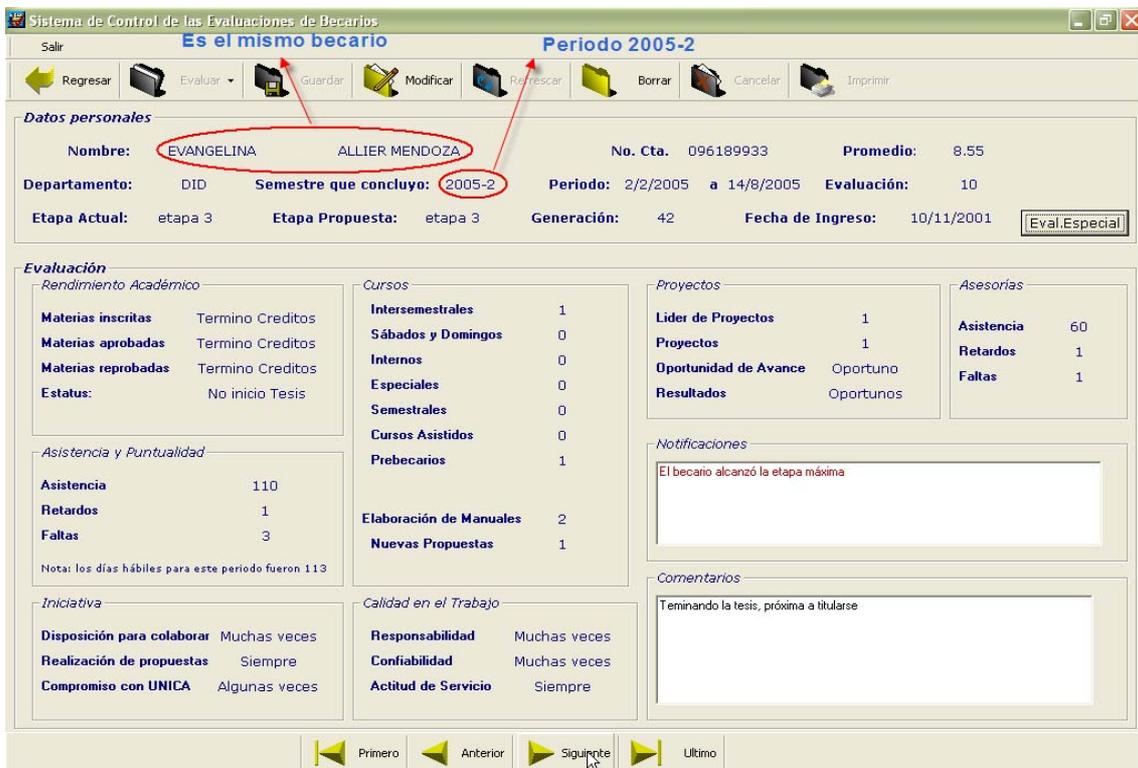


Figura 5.6.6.3.5 Al dar siguiente, se ve que es el mismo becario, pero los datos y el semestre corresponden a otra evaluación

### 5.6.7 Pruebas para la realización de búsquedas

Se solicitó dentro de los requerimientos que existiera un área especial que fuera activada cuando se deseara realizar una búsqueda, que son realizadas por parámetros esto quiere decir que uno puede escoger de varias opciones la o las que necesite para restringir las búsquedas.

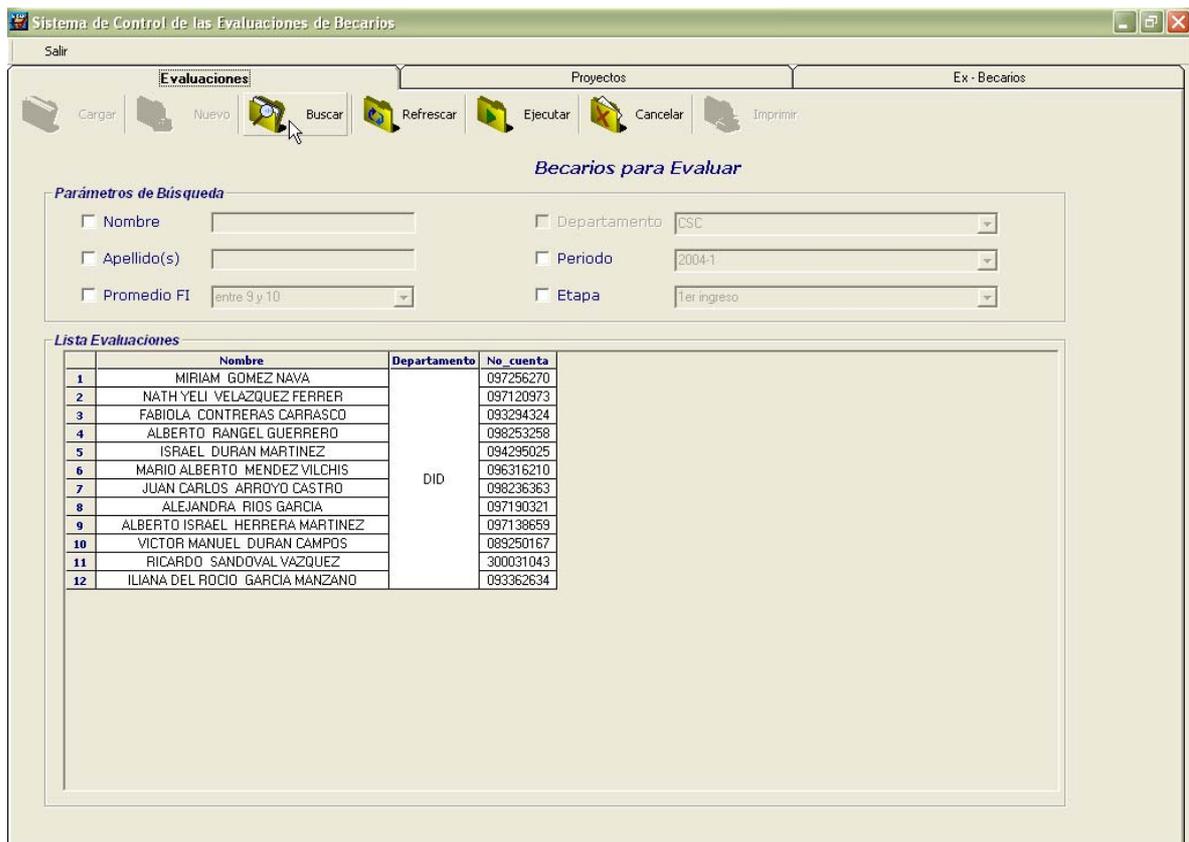


Figura 5.6.7.1 Activación de la búsqueda por parámetros

Las búsquedas se realizan por diferentes parámetros:

- Nombre
- Apellidos
- Promedio de la facultad
- Periodo
- Etapa

El administrador es el único que puede hacer búsquedas por departamento.

Cuando se elige una casilla de alguno de los parámetros la caja de texto asociada a ella se habilita para permitir escribir o bien seleccionar de una caja de opción múltiple la opción de la búsqueda. Esta se puede realizar combinando diferentes parámetros. Por ejemplo la figura 5.6.7.2.1 muestra la búsqueda realizada por nombre:

**Parámetros de Búsqueda**

Nombre   Departamento

Apellido(s)   Periodo

Promedio FI   Etapa

**Lista Evaluaciones**

	Nombre	Departamento	No_cuenta	Periodo	Calificacion	Etapa Actual	Promedio FI	Mat.inscritas	Mat.aprobadas
1	EVANGELINA ALLIER MENDOZA	DID	096189933	2005-1	08	etapa 3	8.52	0	0
2	ERICK REYNALDO VALDIVIA ORTEGA		098026531		05	1er ingreso	9.15	3	3
3	FERNANDO GUTIERREZ LANDA		097135098		08	etapa 2	9.54	4	1

Figura 5.6.7.2.1 Activación de la casilla Nombre, para escribir el nombre del becario del cuál se quiere ver la evaluación.

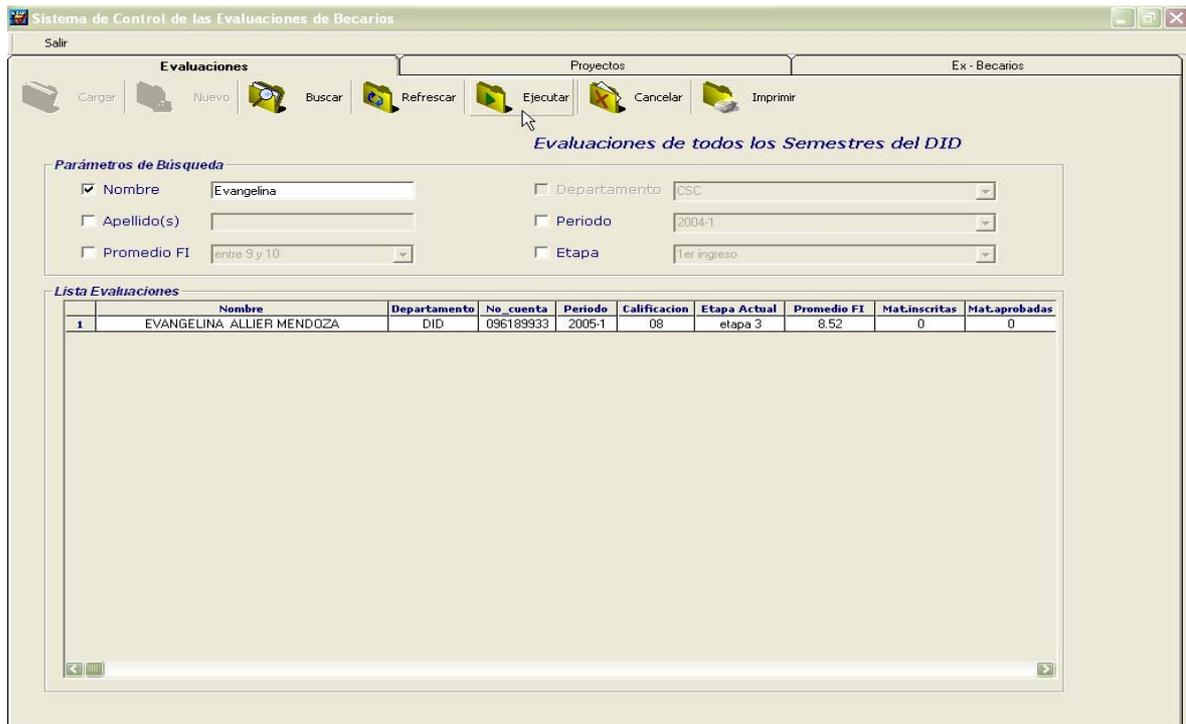


Figura 5.6.7.2.2 Al ejecutar la búsqueda se muestran aquellos registros que coincidan con el parámetro especificado.

De la misma manera si se eligen más parámetros de búsqueda estas son ejecutadas, en la sección de nombre y apellidos se deben escribir tal cual por ejemplo si el becario tiene dos nombre se deben escribir los dos.

#### 5.6.8 Pruebas para la impresión de Reportes.

Dentro de la pantalla principal del SICEB si existen registros dentro del sistema se permitirá realizar la impresión del reporte; actualmente esta opción sólo está habilitada para el administrador del sistema.

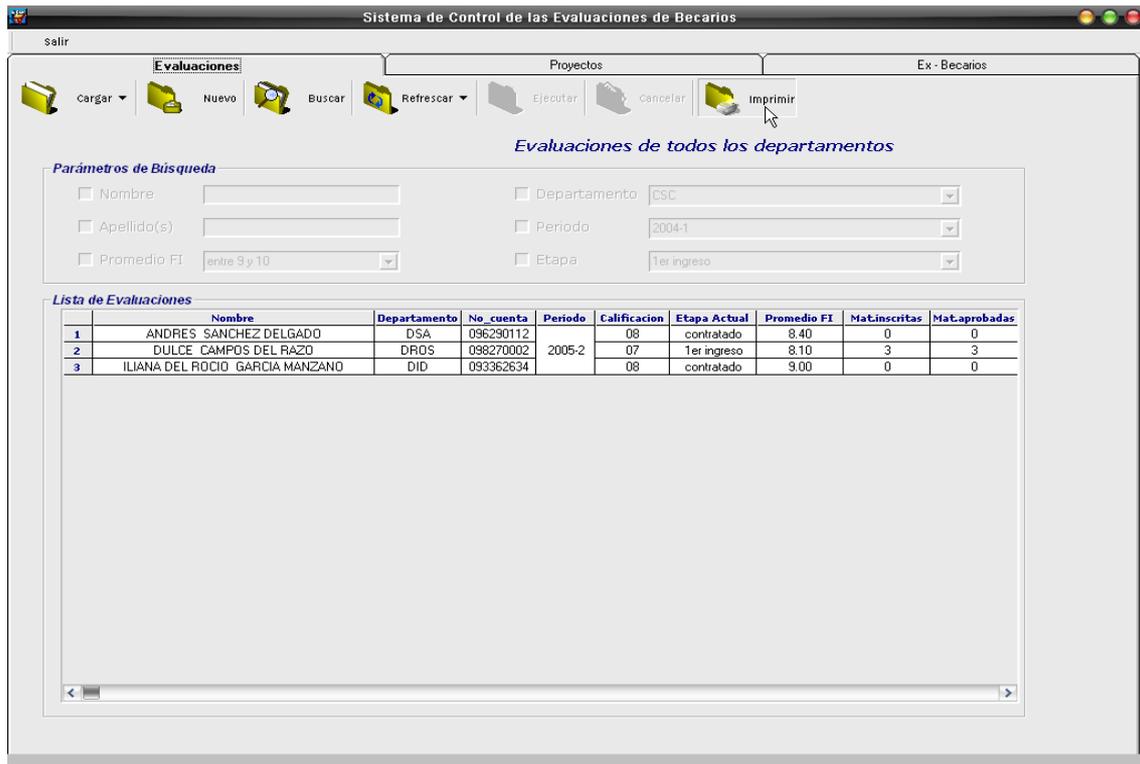


Figura 5.6.7.2.3 Si existen registros en la base de datos, el botón de imprimir se habilita y permite mostrar la pantalla de impresión

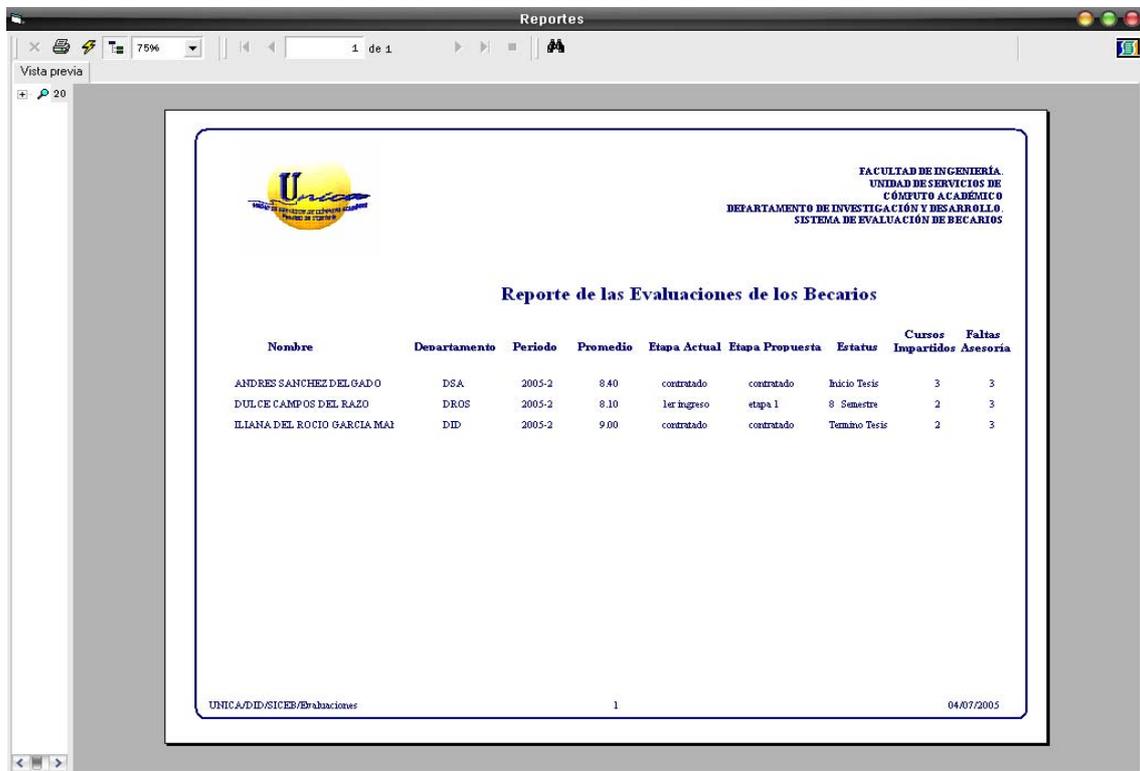


Figura 5.6.7.2.4 Pantalla de impresión del reporte general.



FACULTAD DE INGENIERÍA.  
UNIDAD DE SERVICIOS DE  
CÓMPUTO ACADÉMICO  
DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO.  
SISTEMA DE EVALUACIÓN DE BECARIOS

### Reporte de las Evaluaciones de los Becarios

Nombre	Departamento	Periodo	Promedio	Etapas Actual	Etapas Propuestas	Estatus	Cursos Impartidos	Faltas Asesoría
ANDRES SANCHEZ DEL GADO	DSA	2005-2	8.40	contratado	contratado	Inicio Tesis	3	3
DULCE CAMPOS DEL RAZO	DROS	2005-2	8.10	1er ingreso	etapa 1	8 Semestre	2	3
ILIANA DEL ROCIO GARCIA MAI	DID	2005-2	9.00	contratado	contratado	Termino Tesis	2	3

UNICA/DID/SICEB/Evaluaciones 1 04/07/2005

*Figura 5.6.7.2.5 Vista del reporte general una vez impreso.*

Evaluaciones Individuales

1 de 1+

Vista previa



Facultad de Ingeniería  
Secretaría General  
Unidad de Servicios de Cómputo Académico

### Evaluación de Becarios

Nombre del Becario: ANDRES SANCHEZ DELGADO		No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005-2	Estatus: Inicio Tesis	
Etapas actual: contratado	Etapas propuesta: contratado	Evaluación: 08	

Característica	Desarrollo	Cantidad
Rendimiento Académico	Materias Inscritas	0
	Materias Aprobadas	0
	Materias Reprobadas	0
Asistencia y Puntualidad en UNICA	Asistencias	100
	Retardos	1
	Faltas	14
Iniciativa	Disposición para colaborar	Muchas veces
	Realización de propuestas	Algunas veces
	Compromiso con UNICA	Algunas veces
Cursos	Intersemestrales	1
	Internos	0
	Especiales	0
	Prebecarios	1
	Sábados y Domingos	1
	Semestrales	0

*Figura 5.6.7.2.6 Pantalla de impresión del reporte individual.*

<b>Evaluación de Becarios</b>		
Nombre del Becario: ANDRÉS SANCHEZ DELGADO	No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005-2	Estatus: Inicio Teses
Equipo actual: continúo	Equipo propuesta: continúo	Evaluación: 08
Categorización	Detalle	Cantidad
Rendimiento Académico	Materias Inscritas	0
	Materias Aprobadas	0
	Materias Reprobadas	0
Asistencia y Puntualidad en UNICA	Asistencias	100
	Retardos	1
	Faltas	14
Iniciativa	Disposición para colaborar	Muchas veces
	Realización de propuestas	Algunas veces
	Compromiso con UNICA	Algunas veces
Cursos	Intercursuales	1
	Interno	0
	Especiales	0
	Prebexarios	1
	Sábados y Domingos	1
	Seminarios	0
	Elaboración de Minutaes	2
	Cursos Asociados	0
	Nuevas propuestas	1
Calidad en el trabajo	Responsabilidad	Algunas veces
	Confiable	Muchas veces
	Actitud de Servicio	Algunas veces
Proyectos	Lider de Proyectos	0
	Participación en Proyectos	0
	Oportunidad de Avance	Mínimo
	Resultados	Regular
Asesorías	Asistencias	100
	Retardos	2
	Faltas	3
Comentarios		
Notas	El becarío está a consideración por el número de faltas en asesoría	

UNICA/DID/SICEB 12/08/2001

Figura 5.6.7.2.7 Vista del reporte ya impreso

De esta forma se realizó el plan de pruebas para el SICEB obteniendo como resultado del mismo algunos errores durante su realización, lo cual era el objetivo de dicho plan.

Esos errores fueron subsanados a la par que se realizaban las pruebas, para al final terminar con un sistema limpio y que cumpla su función, cabe aclarar que si en un futuro se modifica o añade más código al sistema este debe de ser probado antes de actualizar la versión del usuario.

*Conclusiones*

---

---

Capítulo

6

## 6. Conclusiones

### *6.1.- Contribución de la Tesis*

Con la creación de este sistema, se puso en práctica la manipulación de la metodología de software OMT y el lenguaje de modelado UML, para su desarrollo. Con base en ellas fue que se analizó el problema de la generación de las evaluaciones semestrales de los becarios de UNICA, dicho problema fue resuelto con las metodologías, siguiendo las bases y preceptos que ellas han sentado para el desarrollo de software, dando como resultado la creación del Sistema de Control de las Evaluaciones de los Becarios de la Unidad de Servicios de Cómputo Académico (SICEB).

Para el modelado del sistema, el contar con el lenguaje UML fue fundamental, ya que gracias a él el tiempo de desarrollo fue bien empleado desde un principio debido a que nos brindó una vista general del sistema y poco a poco se fue haciendo más específica. Gracias a esto los requerimientos del cliente pudieron ser llevados al sistema.

Se logró desarrollar un sistema robusto, estable y funcional, fue muy gratificante el trabajar en este ambiente de programación además de que resultaron sencillas las implementaciones que se realizaron para conseguir que el sistema trabajara como se esperaba; de manera rápida se podían subsanar las carencias que tuviera el lenguaje Visual Basic, en cuanto a la programación orientada a objetos.

A continuación se listan los resultados obtenidos en la realización del Sistema de Control de las Evaluaciones de los Becarios (SICEB) para la Unidad de Servicios de Cómputo Académico (UNICA):

- Se desarrolló un sistema capaz de realizar las evaluaciones de los becarios de la Unidad, con el propósito de agilizar este proceso que se lleva a cabo cada semestre.
- Gracias al sistema se genera un historial en base de datos, el cual puede ser consultado las veces que sean necesarias para generar los reportes y estadísticas.
- Se tiene una interacción dinámica con el: Sistema de Inscripción y Control de Cursos (SICC), ya que en él se encuentran almacenados los

datos personales de los becarios los cuales son mostrados en el momento de realizar la evaluación.

- El proceso de evaluación se encuentra automatizado y es controlado a través del Sistema.
- Se optimizó el tiempo que se invierte en llevar a cabo las evaluaciones.
- Se realizan las evaluaciones de todos los becarios de un mismo departamento.
- Se cuenta con una interfáz segura que sólo permite el acceso a usuarios autorizados.
- A cada jefe de departamento se le está permitido ver y evaluar a sus becarios respectivos.
- Solamente se permite modificar las evaluaciones del periodo actual, haciendo al sistema confiable, pues los datos posteriores que se conservan no pueden ser alterados.
- Con el desarrollo de este sistema se utilizó una metodología de software que ya estaba en uso dentro de la Unidad y que ha dado excelentes resultados.
- El SICEB se ha convertido en un sistema que conjunta la información de otros ya existentes como lo son el SICC (Sistema de Inscripción y Control de Cursos) y el CORA (Sistema de Control y Reporte de Asesorías). Siendo una forma resumida y práctica de visualizar la información más importante de estos dos sistemas en uno.

## ***6.2.- Perspectivas a Futuro***

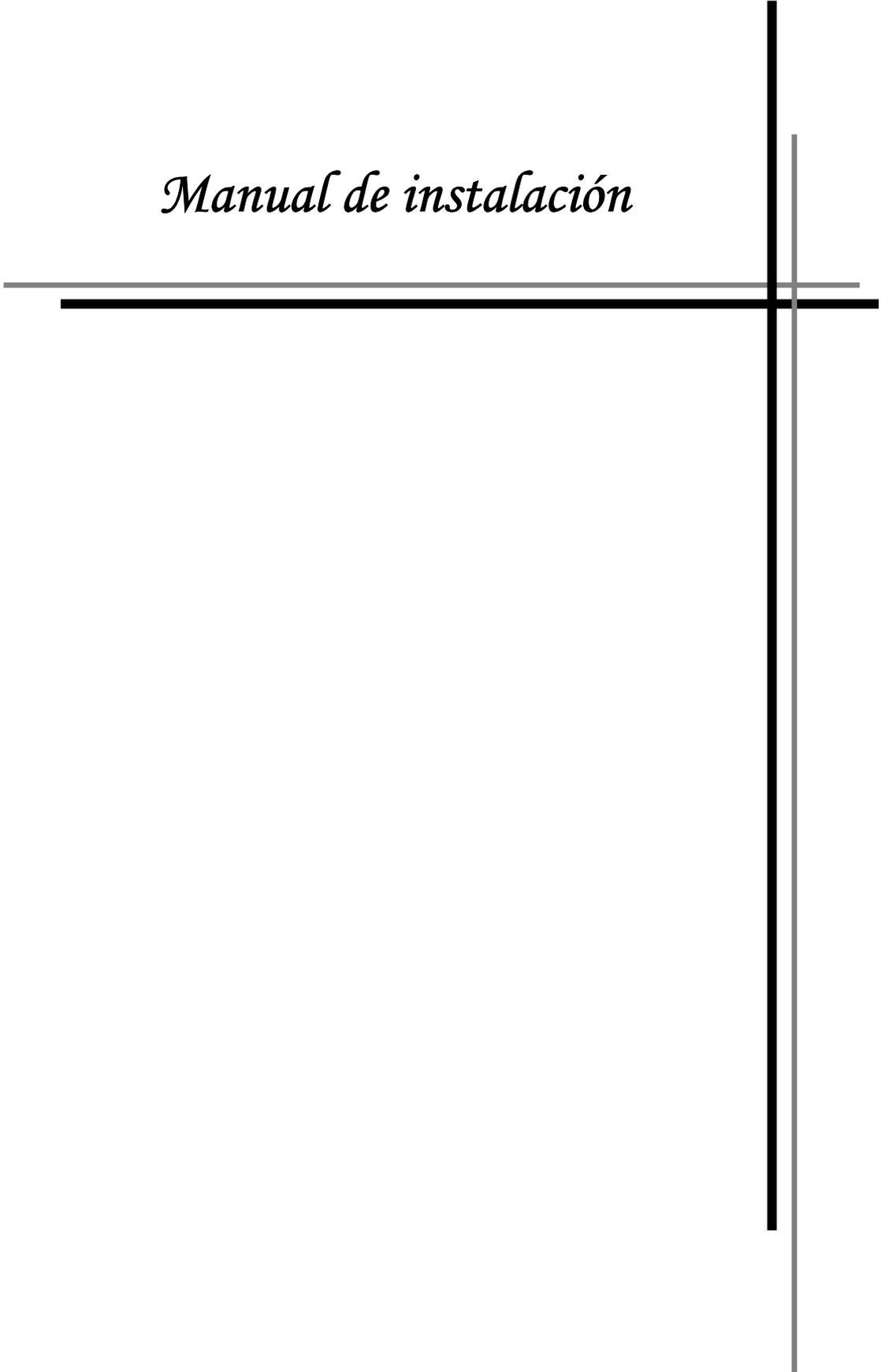
Dadas las características del SICEB, es muy fácil ampliarlo para implementar aplicaciones para las cuales en sus inicios no estaba pensado como lo son los módulos de proyectos, ex - becarios y reporte semestral, que aunque son mencionados dentro de este trabajo aun no están concluidos pues se definió que el presente sólo abarcaría las evaluaciones, sin embargo estos módulos se encuentran ya en desarrollo.

Con base en lo anterior, la existencia del SICEB cobra una importancia vital, debido a que este sistema se convierte una conjunción de varios puntos que dentro de la Unidad se toman en cuenta para la permanencia de sus becarios.

Mientras que el SICC controla toda la información personal de cada becarios y el CORA lo referente a las asesorías que da, el SICEB muestra esta información condensada por semestre, lo cual facilita su visualización y su búsqueda inmediata de manera que el usuario no necesita estar cambiando de sistema para visualizar diferentes datos; el SICEB lo muestra de forma clara y fácil.

Actualmente el módulo referente a proyectos está en fase de desarrollo, para de esta manera tener toda la información referente a ellos en conjunción con la evaluación. Mientras tanto los módulos restantes se encuentran en la fase de modelado, para que cuando se termine se inicie su desarrollo.

# *Manual de instalación*



## 1. Manual de Instalación

El Sistema de Evaluación de Becarios (SICEB) fue realizado en el entorno de desarrollo Visual Basic 6.0, interactuando con Crystal Reports para la generación de los reportes.

Una vez que se ha completado su desarrollo, el siguiente paso es instalarlo en la máquina del cliente, para tal efecto se deben de seguir varios pasos para que el usuario final pueda hacer uso de el.

### 1.1 Instalación del controlador: PsqIODBC

El archivo ejecutable PsqIODBC nos provee del controlador de orígenes de datos para poder realizar las consultas a la base (que esta alojada en un servidor que maneja Postgres) desde el sistema. Por lo general este archivo se puede descargar gratuitamente desde la página:

<http://www.postgreSQL.org>



*Figura 1.1.1 Presentación del icono del controlador PsqIODBC*

Una vez que lo tengamos, se debe de instalar en la máquina, esta es sencilla sin mayores complicaciones:

Se empieza por dar doble clic sobre el icono, esto nos mostrará una pantalla como la siguiente:



Figura 1.1.2 Ventana de instalación del PsqlODBC

Luego de dar clic al botón de “Next”, aparecerá una pantalla donde debemos de aceptar la licencia que el controlador tiene.

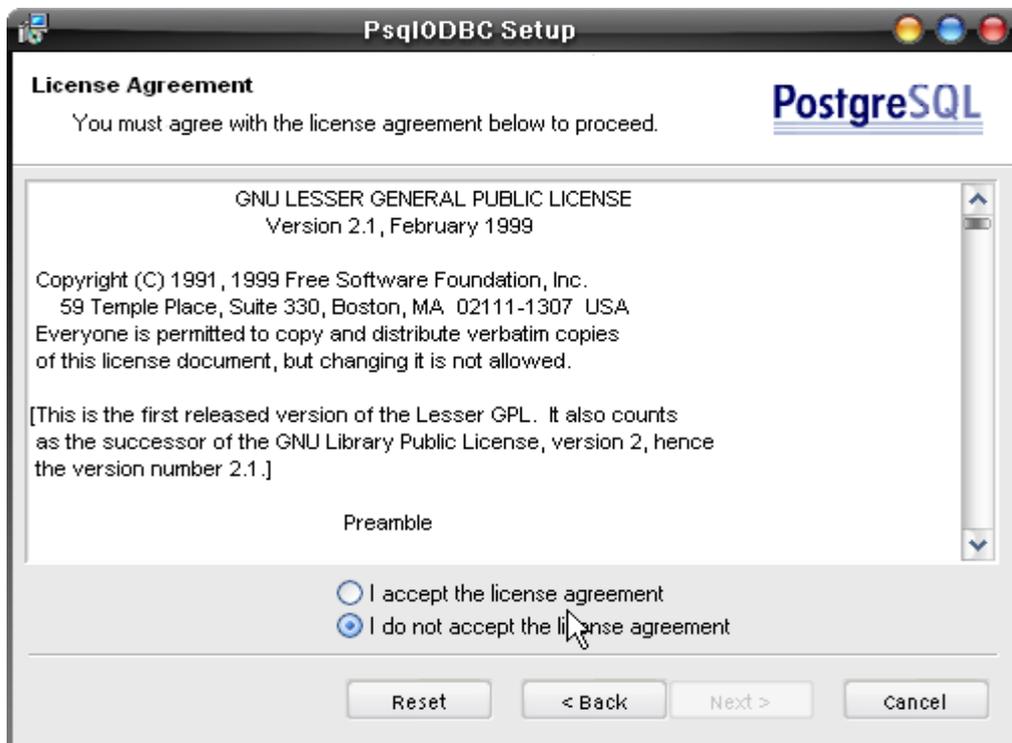
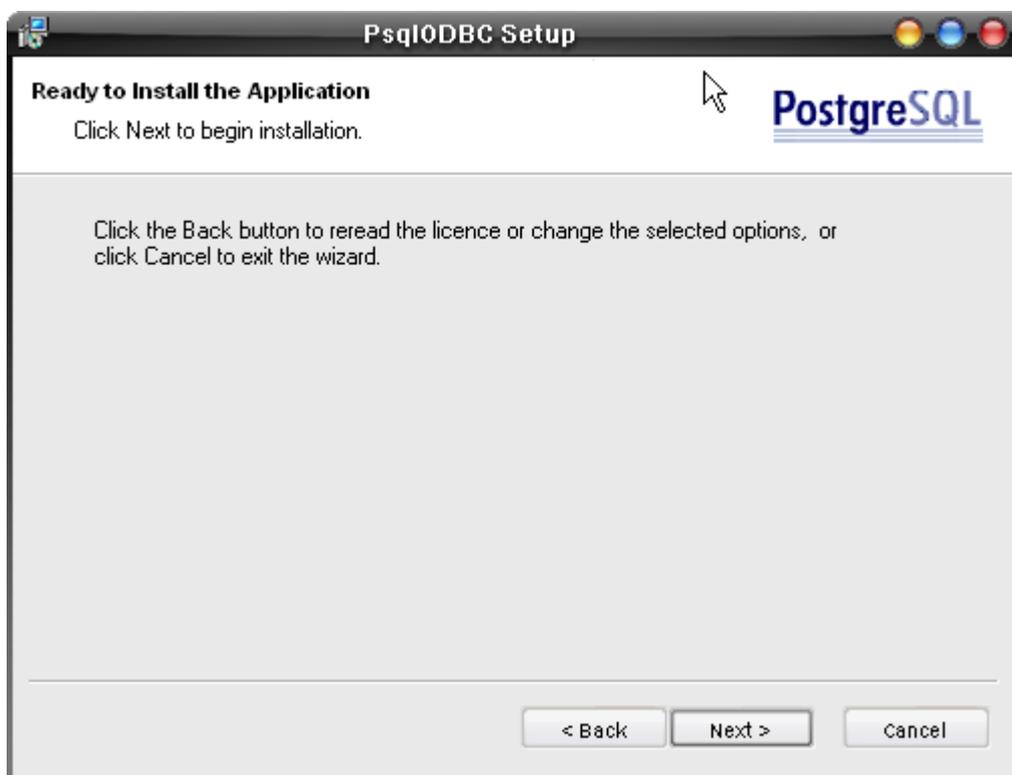


Figura 1.1.3 Ventana de aceptación de las políticas de PostgreSQL

Al aceptar la licencia es decir, habiendo seleccionado la primer opción que aparece debajo, el botón de “*Next*” se habilitara permitiéndonos continuar con la instalación al oprimirlo se muestra la siguiente pantalla:



*Figura 1.1.4 Ventana que indica que esta listo para instalar el controlador.*

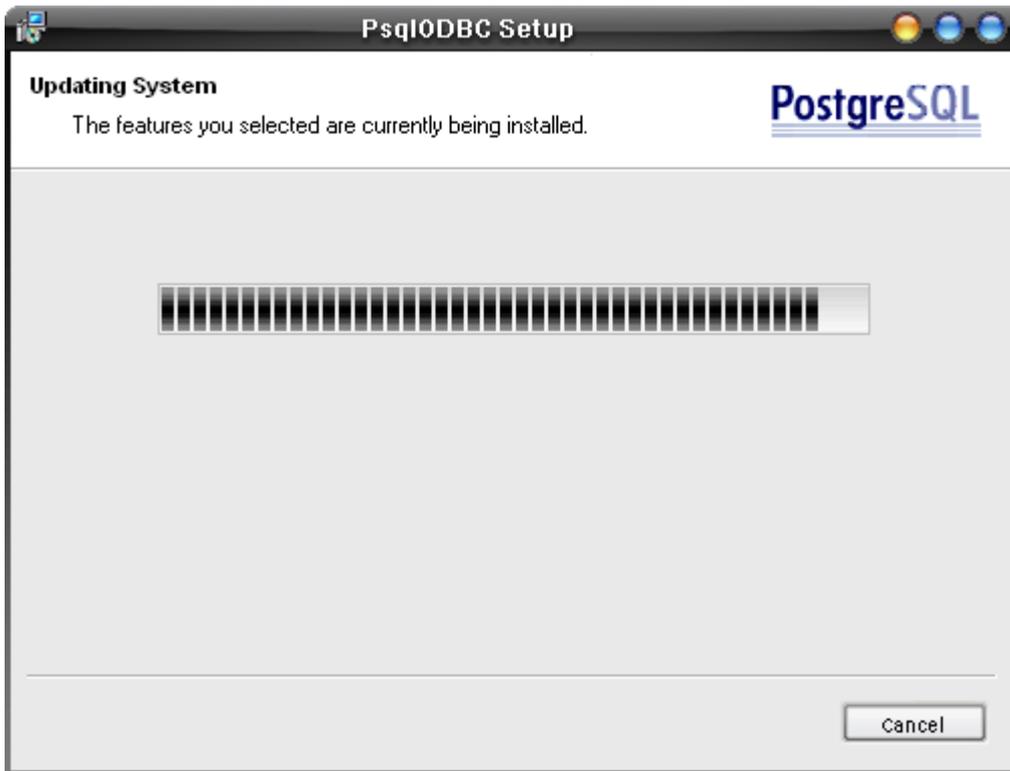


Figura 1.1.5 Instalación del controlador en proceso.

Una vez terminada la instalación nos muestra la pantalla de finalizar, se debe de dar clic al botón de "Finish", terminando de esta manera con la instalación.

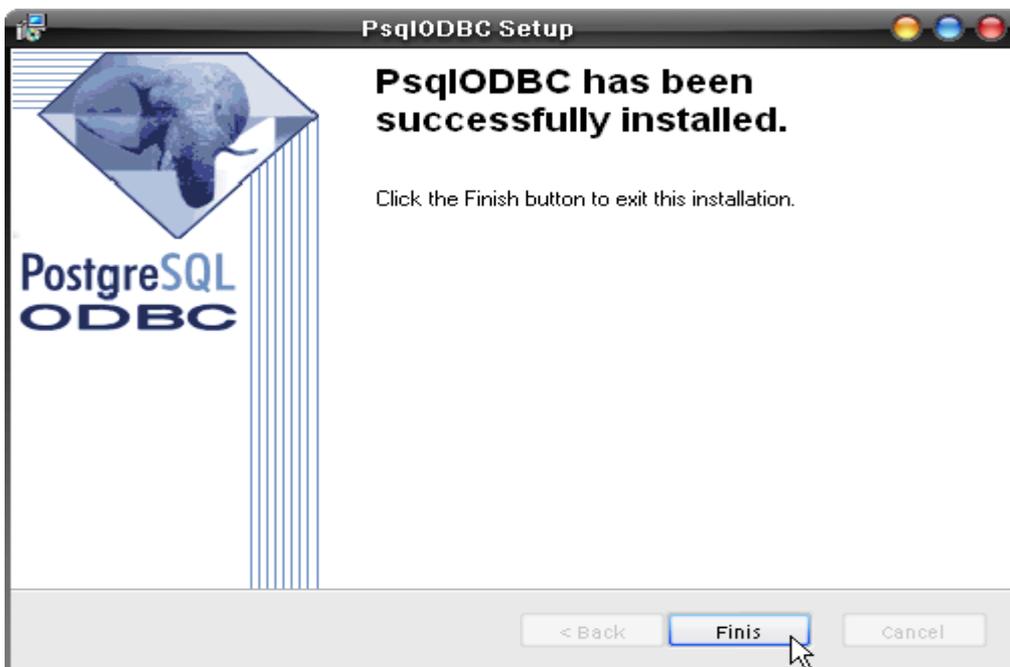


Figura 1.1.6 Pantalla de finalización de la instalación

## 1.2 Dando de alta las Bases de Datos.

Una vez terminado lo anterior, se deben dar de alta en el sistema las dos bases de datos de las que hace uso el SICEB. Esto se lleva a cabo desde el panel de control en la sección de “Herramientas administrativas”:



*Figura 1.2.1 Icono de Herramientas administrativas dentro del panel de control.*

Cuando se le da doble clic se abre una ventana con diferentes opciones, el icono que nos interesa es el llamado: Orígenes de Datos.



*Figura 1.2.2 Icono de orígenes de datos*

Al dar doble clic se abre una ventana donde aparecen varias opciones agrupadas por tabuladores las que nos interesan son aquellas marcadas con los nombres de: DNS del sistema y DNS de usuario.

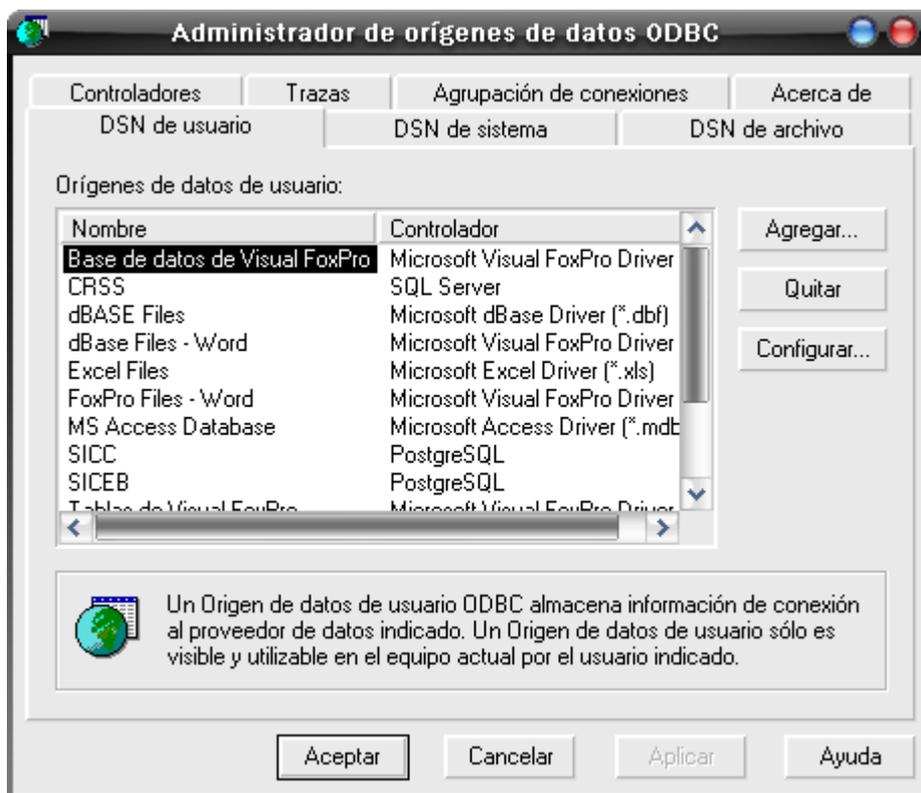


Figura 1.2.3 Ventana de opciones de ODBC

Aquí en ambas ventanas, tanto DNS de usuario como de sistema, debemos de dar de alta las dos bases de datos que utiliza el SICEB, esto se hace haciendo clic al botón de “Agregar”, el cual nos lleva a la siguiente pantalla:



Figura 1.2.4 Elección del controlador

Como podemos observar, en esta ventana se debe elegir controlador del origen de datos de entre todos los que se listan se debe buscar el que se llama PostgreSQL y se da clic a “Finalizar”.

Una vez realizado esto se van a dar de alta los datos de las base en la siguiente pantalla:

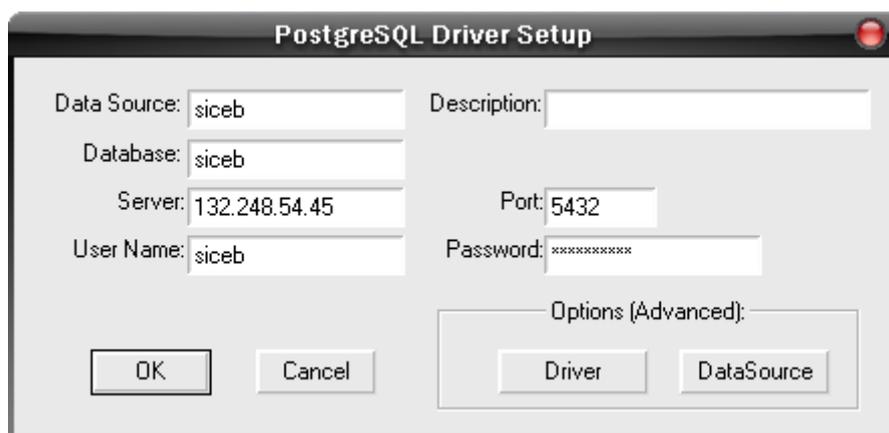


Figura 1.2.5 Pantalla para introducir los datos de la base.

Los datos que se requieren son:

- *Data Source*: Se coloca el nombre con el que se conocerá la base, aquí no necesariamente debe ser el nombre dentro del servidor.
- *Description*: Esta opción nos permite dar una pequeña explicación de la base.
- *Database*: Aquí si se debe de colocar el nombre de la base, tal y como es conocida dentro del servido
- *Server*: Se coloca ya sea la dirección IP del servidor o bien su dirección URL.
- *Port*: Esta opción por default indica el puerto de conexión 5432, a no ser que sea otro, se debe de dejar este.
- *User name*: Nombre de usuario que tiene permisos para acceder a la base.
- *Password*: Contraseña del usuario.

Una vez llenados estos campos se da clic al botón “OK” , con esto regresamos a la pantalla de la figura 1.2.3, con en el caso del SICEB, son dos bases de datos las que se utilizan se debe dar de alta la otra, siguiendo el mismo procedimiento. Al terminar, se debe de volver a hacer lo mismo para las dos base en el tabulador de: DNS del sistema.

Todo lo anterior nos sirve para que la máquina donde se va a instalar el SICEB pueda comunicarse con las bases de datos.

## 1.2 Instalación del MDAC

MDAC es una acrónimo para “*Microsoft Data Access*”, este es un controlador para que *Windows* pueda hacer uso de las opciones de conectividad a bases de datos, ODBC, ADO y OLE DB.

Muchas veces aunque tengamos el controlador ODBC instalado, el sistema no permite realizar ninguna modificación en la base es decir sólo permite lectura de datos más no escritura.

Por lo general el MDAC esta disponible en la página de Microsoft:

<http://www.microsoft.com>

Sin embargo, para poder descargar actualizaciones se debe de contar con una licencia original pues de no tenerla aunque se puede descargar el archivo este no funcionará.

Una vez instalado ya se puede proceder a la instalación del SICEB.

## 1.3 Creación del archivo de instalación del SICEB.

Una vez realizado todo lo anterior, se procede a crear el archivo de instalación del SICEB, para esto y antes que nada se debe de crear un archivo ejecutable, esto se realiza dentro de la ventana Visual Basic 6.0 en la opción de Archivo -> Generar .EXE, al escoger esta opción la aplicación comenzará a compilar el proyecto y creará el archivo ejecutable. Si bien este ya puede ser utilizado, se le debe por lo general de “empaquetar” ¿porqué?, bien por la razón de que es muy probable que la máquina donde se vaya a instalar no cuente con algunos archivos de sistema o librerías (.DLL) que el sistema utiliza, al empaquetarlo aseguramos que el archivo de instalación contendrá todo lo necesario para que el sistema funcione.

### 1.3.1 Empaquetado del SICEB

Visual Basic, viene a parte de con la aplicación para desarrollo de proyectos, con varias utilidades más (que se instalan al momento de instalar Visual 6.0) entre ellas el “Asistente de empaquetado y distribución” para poder utilizarlo debemos de ir al Menú de Inicio -> Programas -> Microsoft Visual Studio 6.0 -> Herramientas de Microsoft Visual Studio 6.0 -> Asistente para empaquetado y distribución.

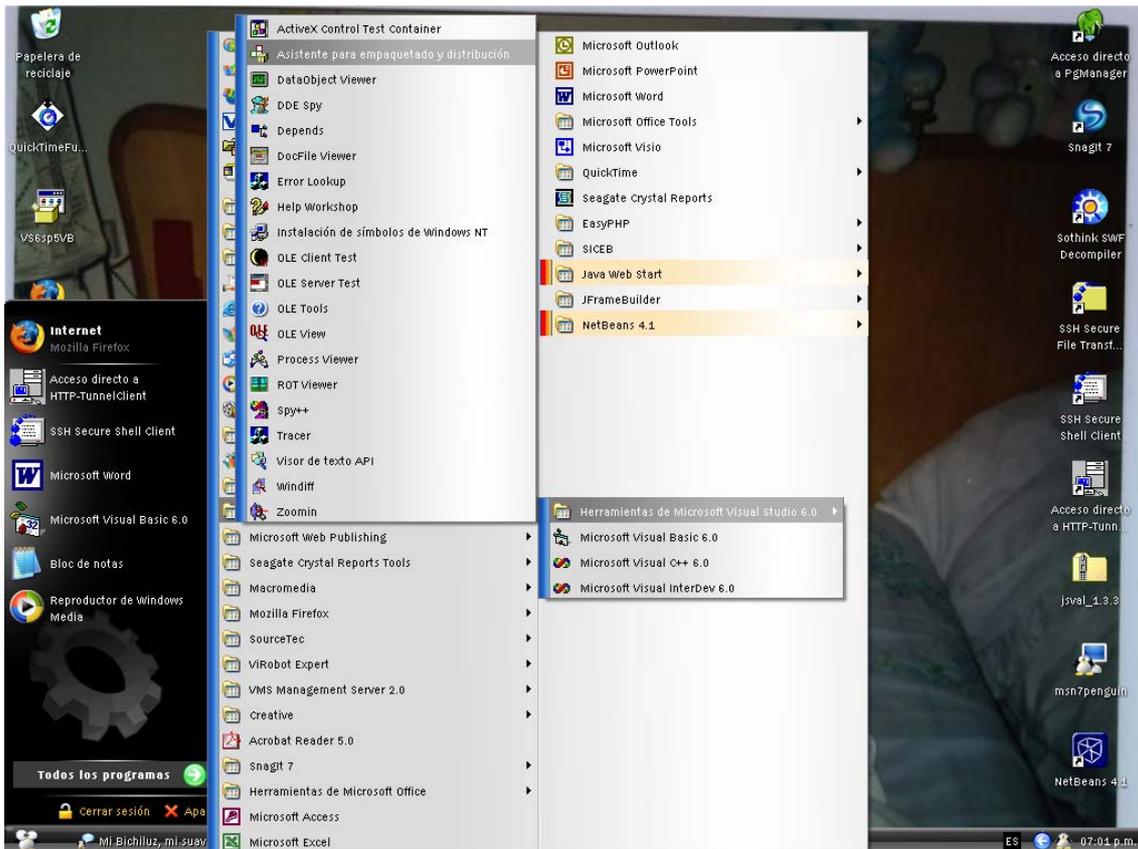


Figura 1.3.1.1 Selección de la herramienta de empaquetado.

Una vez seleccionado nos muestra una pantalla con diferentes opciones, en la parte superior se encuentra una caja de opción múltiple, de la cuál se va a escoger el proyecto que se va a empaquetar, en caso de que no se encuentre en la lista se debe de dar clic al botón de examinar para buscar la ruta de donde se encuentra nuestro ejecutable.

Ya que se dio la ruta del proyecto a empaquetar se procede a dar clic al botón de ese nombre, que es el primero que aparece en la lista como lo muestra la figura:

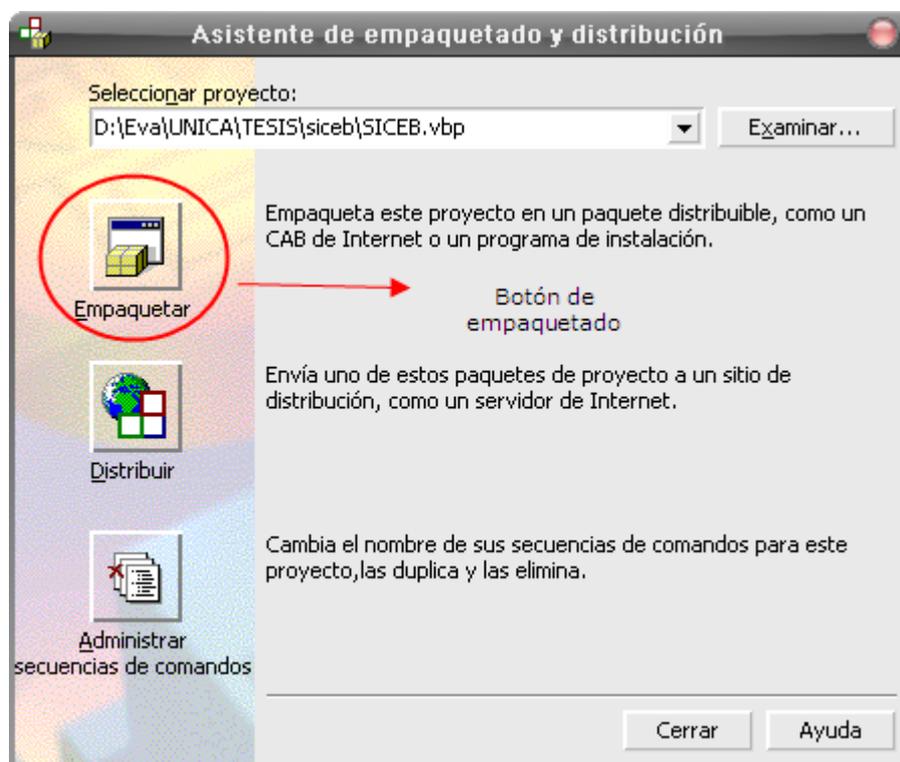


Figura 1.3.1.2 Ventana para el empaquetado

Al elegir empaquetar, nos muestra la ventana de que el asistente se está preparando:

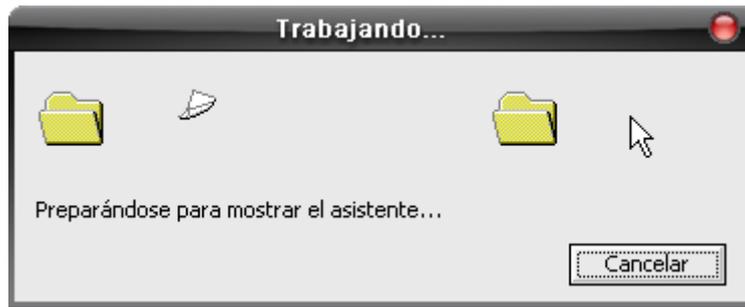


Figura 1.3.1.3 Mostrando el asistente.

Al terminar de cargarse el asistente, muestra la pantalla de secuencia de comandos, esta opción indica de que manera se irá instalando el sistema.

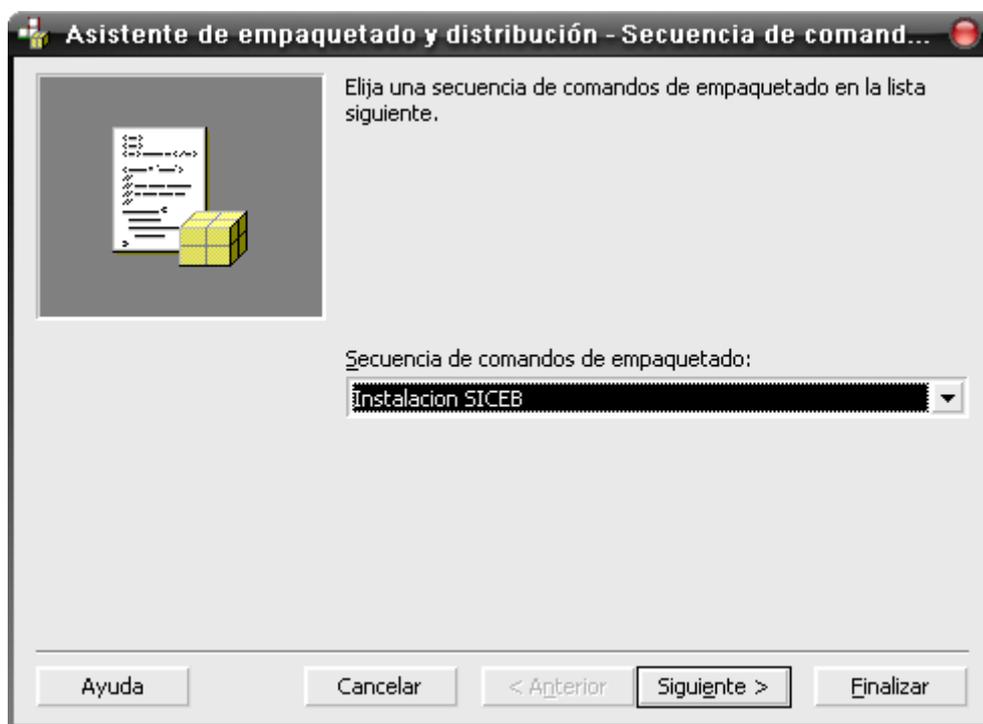


Figura 1.3.1.4 Elegir una secuencia de comandos.

Luego de la secuencia se debe de elegir el tipo de paquete, dentro de dos categorías: Paquete de instalación estándar, donde se creará un paquete de instalación ejecutable (recomendado) y el Archivo de dependencia que se usa para crear un archivo que almacene información sobre componentes de tiempo de ejecución que requiere su aplicación.



Figura 1.3.1.5 Selecciónar el tipo de empaquetado

Se elige la ruta y carpeta donde se guardará el paquete de instalación, si la carpeta no existiera la va a crear, se le puede cambiar el nombre por *default* le da el nombre de Paquete; de lo contrario si ya existe, preguntará si desea sobrescribirla.

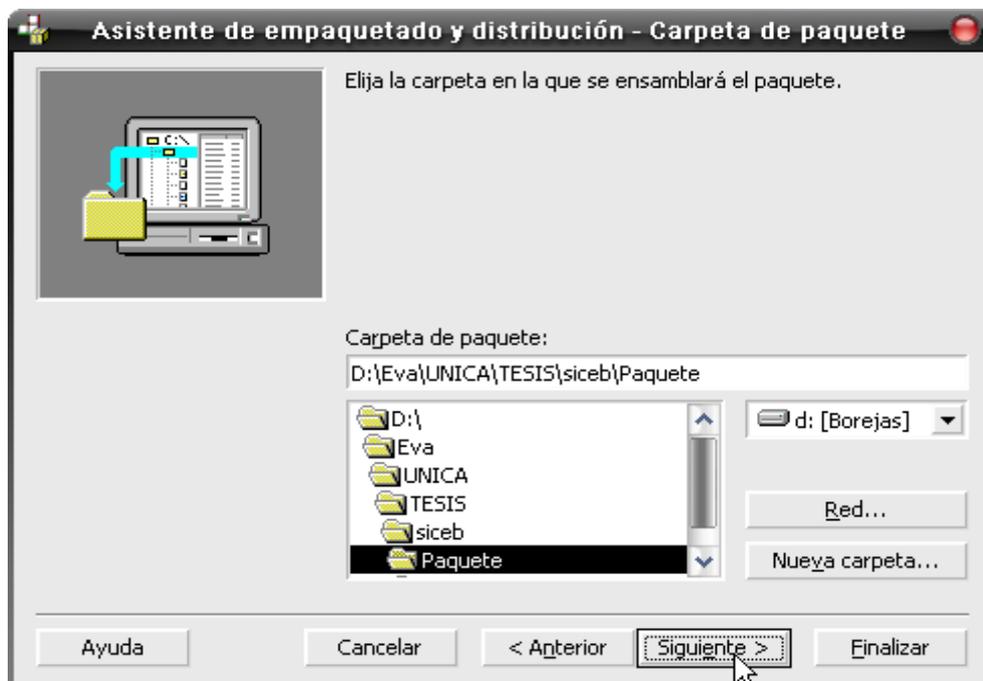


Figura 1.3.1.6 Eligiendo la ruta y carpeta.

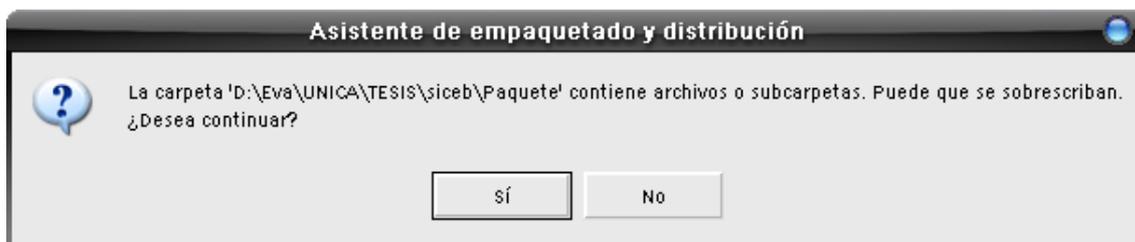


Figura 1.3.1.7 En caso de que la carpeta ya exista, pide confirmar si desea uno sobrescribirla.

En este caso como ya existía un paquete de instalación preguntó si se deseaba sobrescribirlo y se eligió la opción de “SÍ”, de no existir se pasaría directamente de la figura 1.3.1.6 a la 1.3.1.8.

A continuación muestra los archivos que se incluirán (librerías) para que el sistema pueda funcionar, se pueden agregar archivos adicionales o bien quitar los que no se deseen de la lista que se muestra.

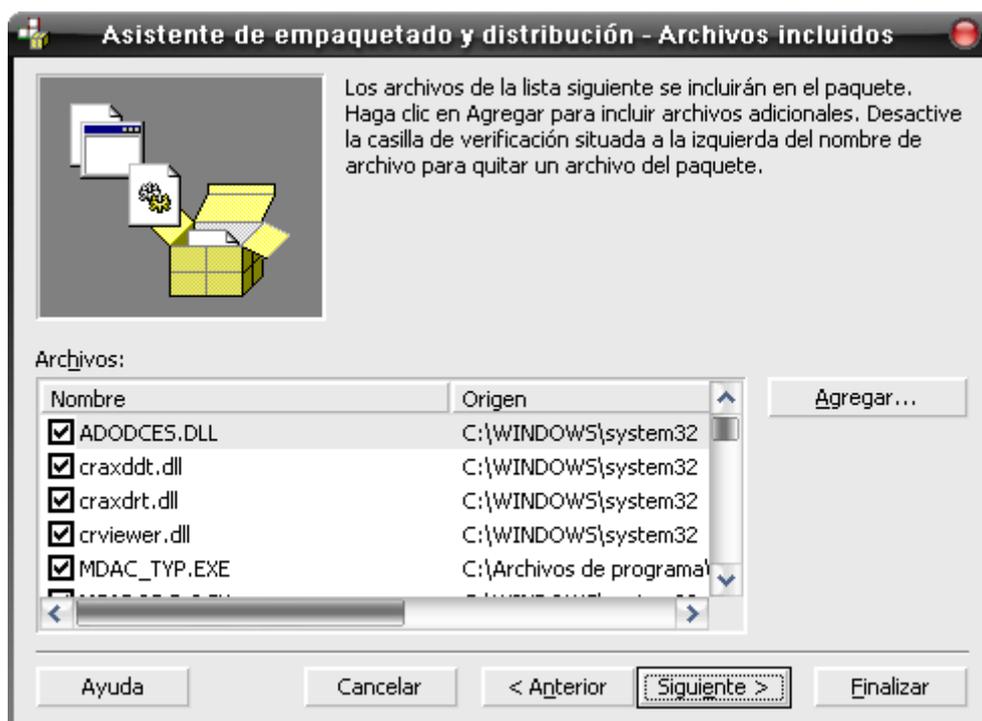
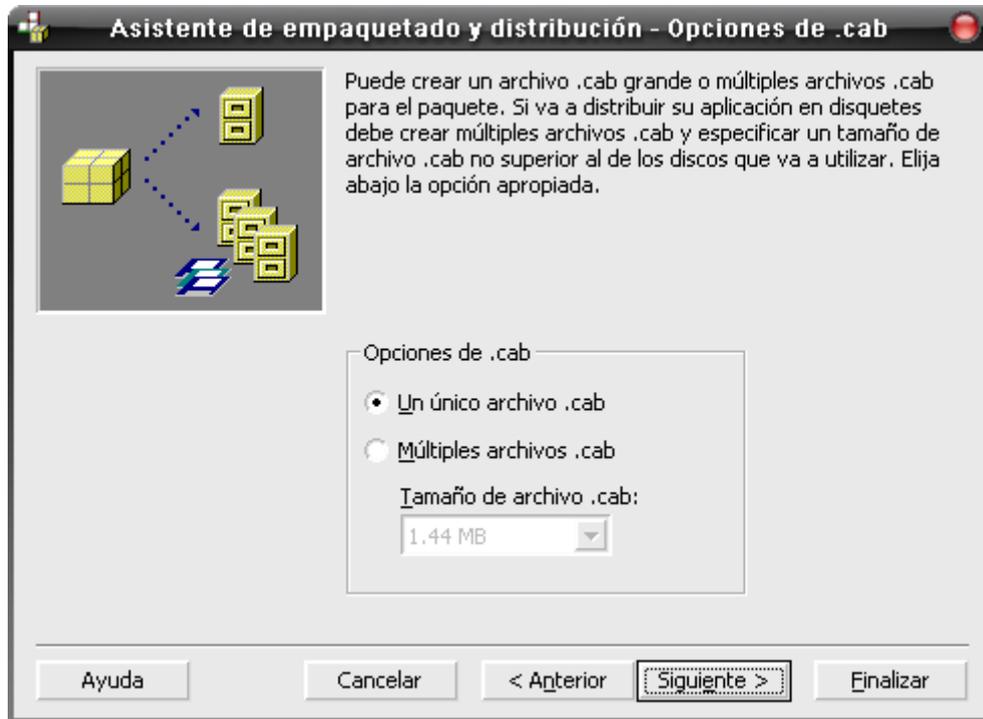


Figura 1.3.1.8 Librerías que se incluirán

Por lo general este asistente busca las librerías y las incluye en la lista de la figura anterior, sin embargo pueden existir algunas que no encuentre en la ruta en la que normalmente están, si se llegara a dar el caso se debe de buscar la librería en la máquina e indicarle al asistente la ruta donde se encuentra para poder incluirla.

Ya hecho lo anterior, se debe elegir como se desea que se cree el archivo de empaquetado ya sea un solo archivo o bien si se va a distribuir en discos se debe de crear múltiples archivos de instalación.



*Figura 1.3.1.9 Tipo de instalación.*

Luego se indica el nombre que se le dará al paquete de instalación cuando este se ejecute.



*Figura 1.3.1.10 Nombre de la instalación*

A continuación se especifica de que manera se instalará el sistema dentro del menú de Windows, es decir la ruta que seguirá se puede dejar la que el asistente marca por default, o bien elegir un nuevo grupo.



*Figura 1.3.1.11 Grupos del menú inicio.*

Como se incluyen librerías dentro de nuestro empaquetado, pudiera ser que al instalarlas, se desee que queden en alguna otra ruta dentro de la maquina del usuario que la predeterminada por Windows, en la siguiente ventana se puede modificar lo anterior.

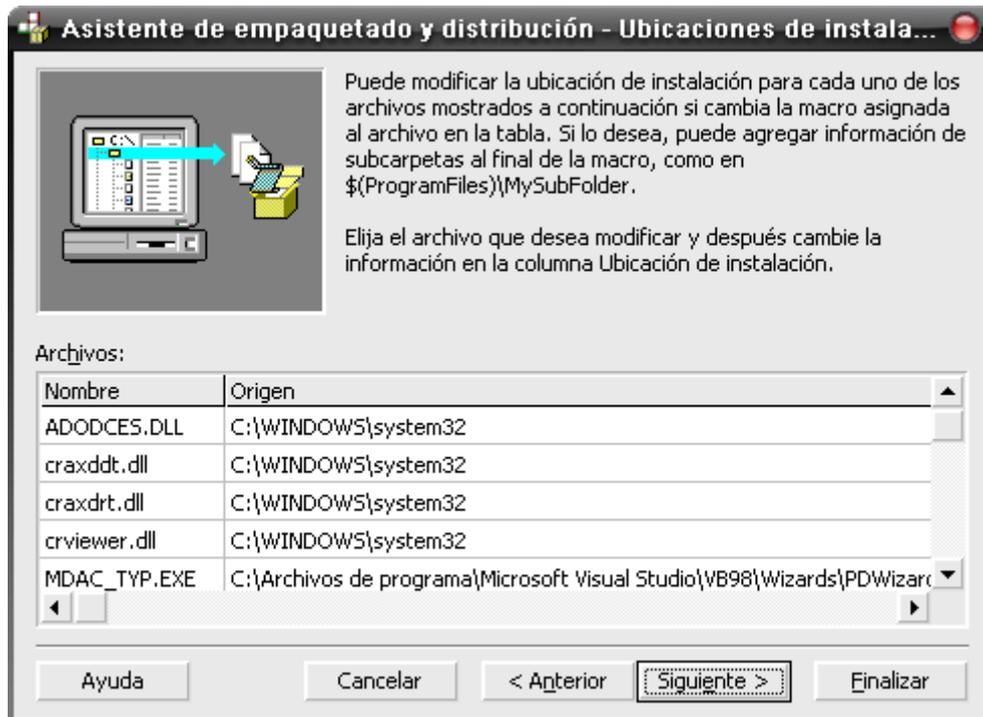


Figura 1.3.1.12 Modificación de las rutas de las librerías.

Se debe de indicar, si se desea que algunos de los archivos que se van a instalar sean compartidos, es decir que más de un programa los puede usar.

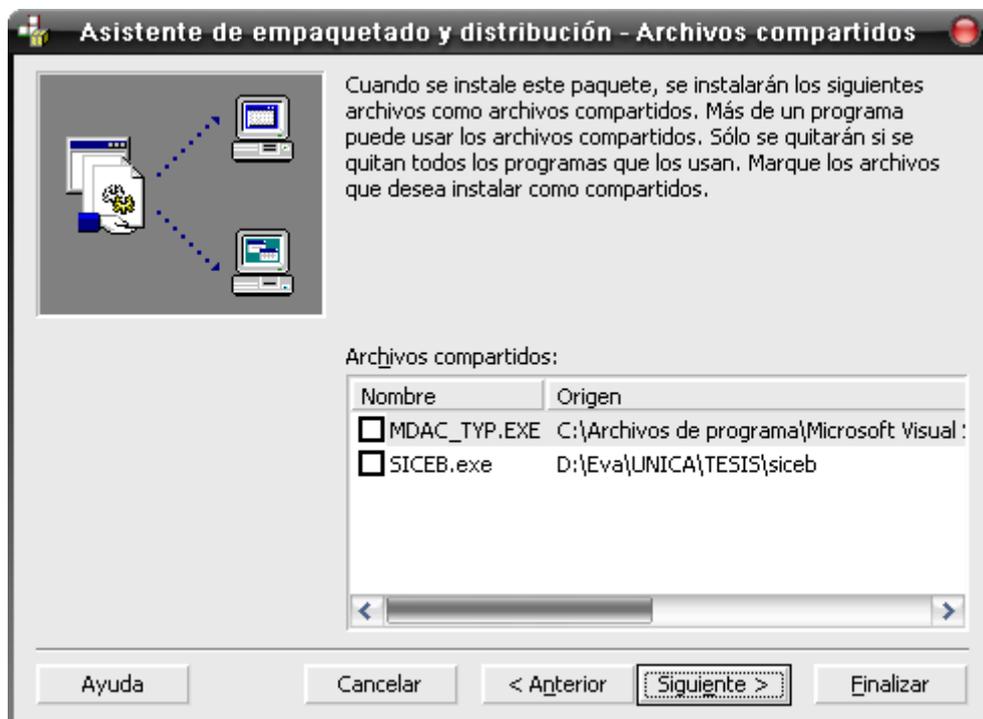


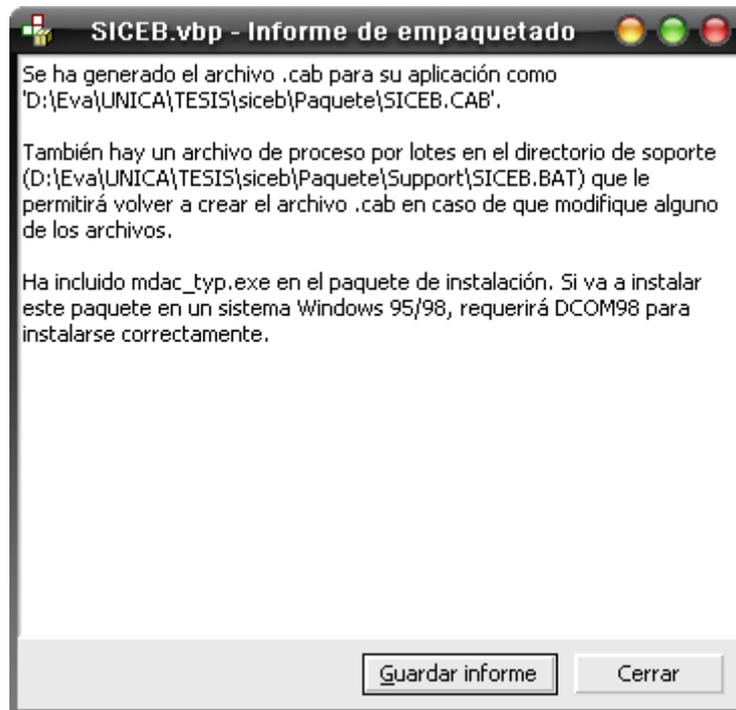
Figura 1.3.1.13 Archivos Compartidos.

En este punto estamos a punto de concluir el empaquetado se confirma el nombre de la secuencia de comandos y se debe de dar clic a Finalizar para que el asistente pueda llevar a cabo el empaquetado.



*Figura 1.3.1.14 Finalizando el empaquetado.*

Por último se genera un reporte, este puede ser guardado para futuras referencias.



*Figura 1.3.1.14 Generación de reporte.*

#### 1.4 Instalación del SICEB.

Ya que se tiene un archivo de instalación este puede ser transportado a la máquina del usuario para instalarle el sistema recordando que antes se instaló el psqLODBC y el MDAC.



*Figura 1.4.1 Icono generado por el asistente de empaquetado para el SICEB.*

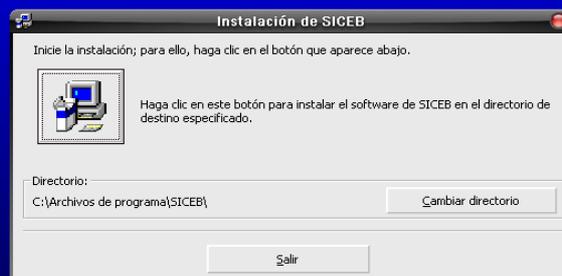
Al dar doble clic en este icono, se comienza e instalar el SICEB.

## ***Instalación de SICEB***



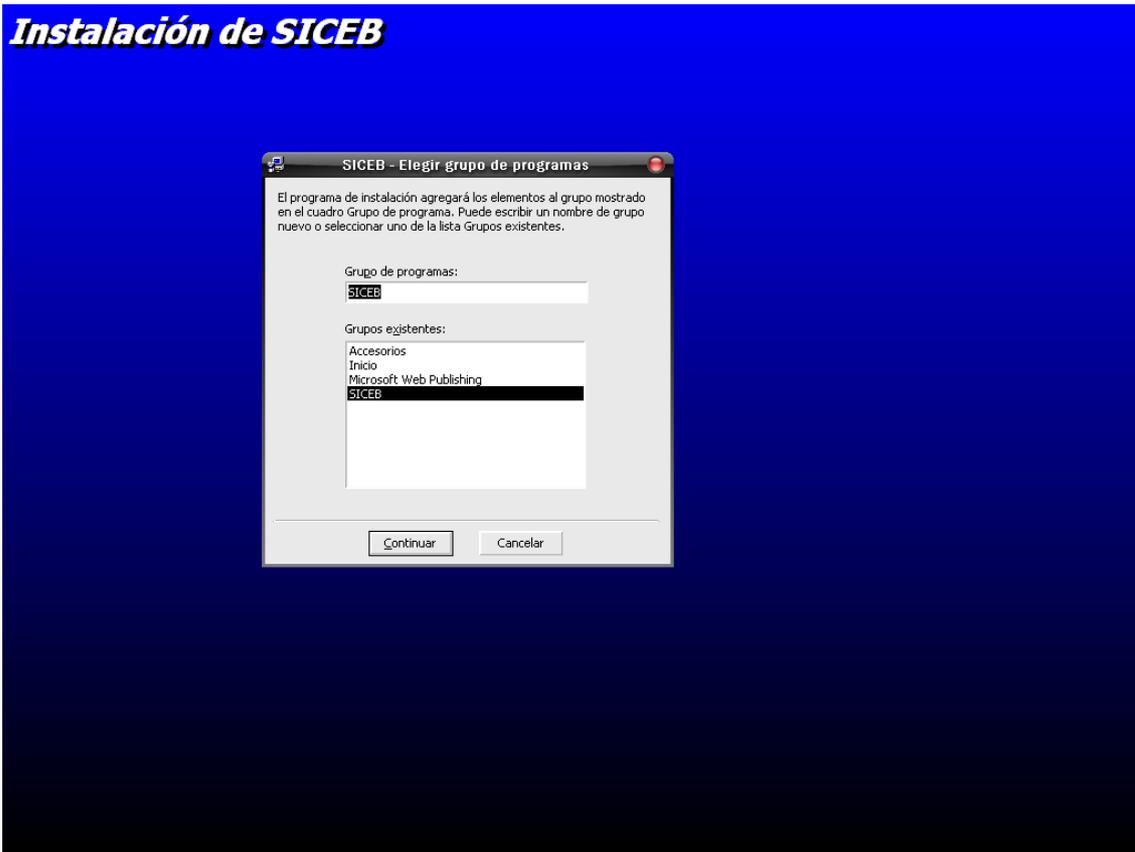
*Figura 1.4.2 Pantalla de instalación del SICEB.*

## ***Instalación de SICEB***



*Figura 1.4.3 Petición para iniciar la instalación*

## ***Instalación de SICEB***



*Figura 1.4.4 Petición para la ruta de instalación del Sistema.*

Luego de esta última pantalla, la instalación será automática y el SICEB quedará listo para utilizarse.

La ruta para poder acceder al SICEB quedará de la siguiente manera:

Inicio  
Programas  
Siceb  
SICEB

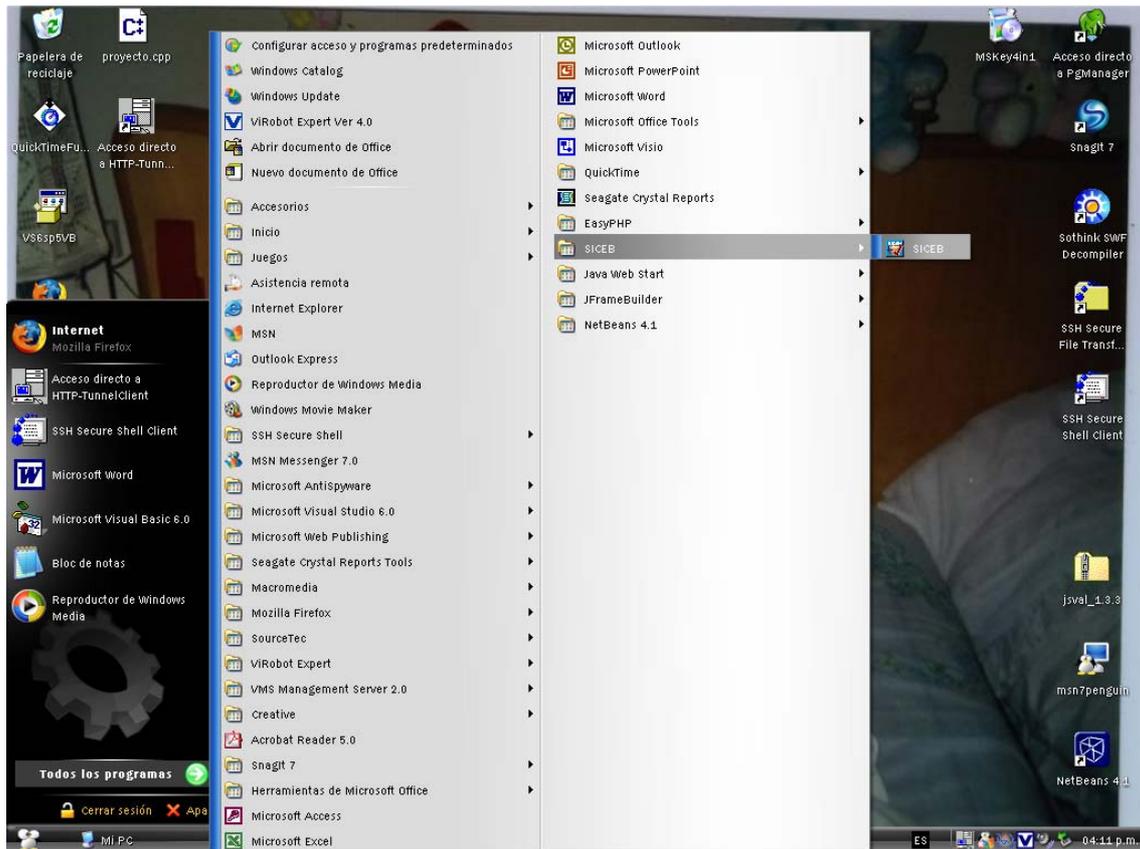
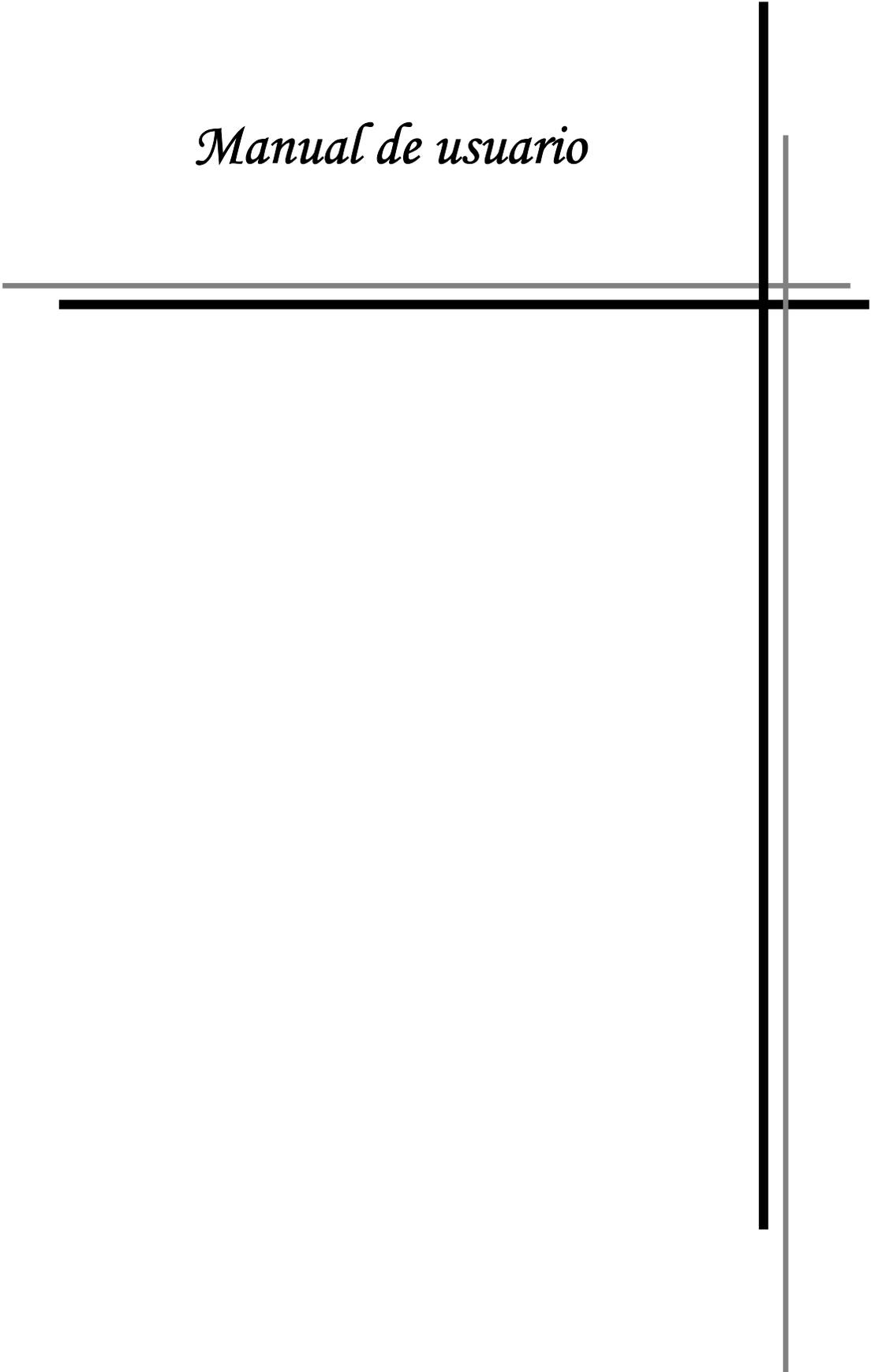


Figura 1.4.5 Ruta para poder acceder al SICEB

# *Manual de usuario*



## 1. Iniciando sesión en el Sistema y la Pantalla Principal

Al ejecutar la aplicación el sistema mostrará una pantalla como la que se indica en la figura 1.1, en la cual se deberá de seleccionar el nombre de usuario, que puede ser cualquiera de los departamentos (Did, Dros, Dsa, Dsc y CSC), siendo el DSA el súper usuario.



*Figura 1.1 Pantalla inicial*

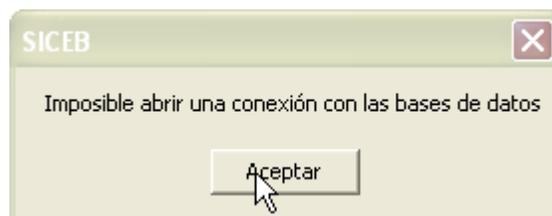
Dentro del combo “Usuario”, tenemos las opciones de escoger entre los cinco departamentos, cada uno de ellos cuenta con una contraseña única, algunos usuarios tienen evaluaciones especiales las cuales serán cargadas al conectarse al sistema.

Un ejemplo de conexión se encuentra en la figura 1.2:



*Figura 1.2 Iniciando sesión en el sistema.*

En caso de que por algún motivo la base de datos del sistema estuviera fuera de servicio al tratar de realizar la conexión se mostrará la advertencia de la figura 1.3.



*Figura 1.3 Error en la conexión.*

Si por el contrario no existiera ningún problema de conexión, pero el usuario introdujo su contraseña de forma incorrecta se visualizaría el siguiente aviso:



*Figura 1.4 Verificación de datos*

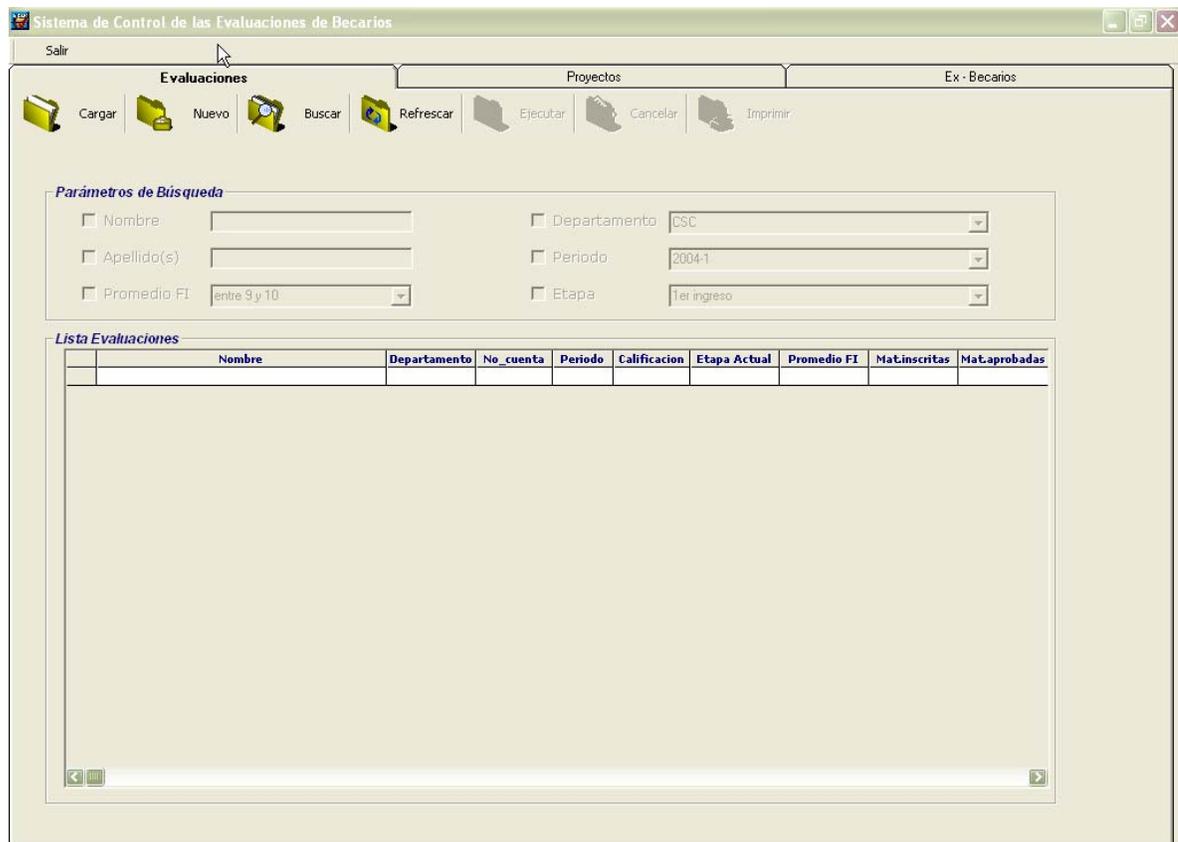
Una vez que se le da clic al botón de Aceptar de la pantalla de la figura 1.4, se regresa a la mostrada en la figura 1.2 donde se deberá volver a introducir la contraseña.

Si no existen errores se mostrará una pantalla de bienvenida, que indicara que el Sistema está siendo cargado.



*Figura 1.5 Carga del Sistema*

En el momento en que finaliza la carga del Sistema, se nos presenta la pantalla principal del SICEB.



*Figura 1.6 Pantalla Principal*

### ***1.1.- Elementos de la Pantalla Principal***

Esta pantalla es la que nos sirve para la manipulación de registros de manera general, así como para realización de búsquedas por parámetros y obtención de nuevos registros para evaluar. Las áreas en las que se divide son:

**Botón de Salida:** Ubicado en la parte superior izquierda, hace salir por completo de la aplicación.



*Figura 1.1.1 Salir del Sistema*

**Barra de herramientas:** Sirve para la manipulación de los registros dentro del sistema:



Figura 1.1.2 Barra de Herramientas

Dentro de esta barra podemos apreciar diferentes botones cada uno de ellos con una función específica:

**Cargar:** Carga todos los becarios que se encuentran ya evaluados.

**Nuevo:** Muestra los becarios que faltan de ser evaluados para el periodo actual.

**Buscar:** Habilita la parte de búsqueda por parámetros.

**Refrescar:** Actualiza la consulta luego de una modificación

**Ejecutar:** El botón se habilita luego de dar clic a búsqueda, sirve para ejecutarla y que se muestren los resultados en pantalla.

**Cancelar:** Anular el realizar una búsqueda.

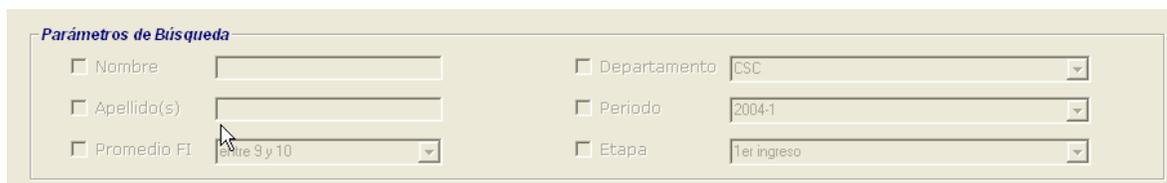
**Imprimir:** Impresión de los reportes de las evaluaciones.

**Tabs para los módulos:** Permite moverse entre las diferentes funcionalidades del sistema estas son: Evaluaciones, Proyectos y Ex Becarios.



Figura 1.1.4 Tabs

**Área de búsqueda:** Esta parte se habilita en el momento de dar clic al botón de búsqueda.

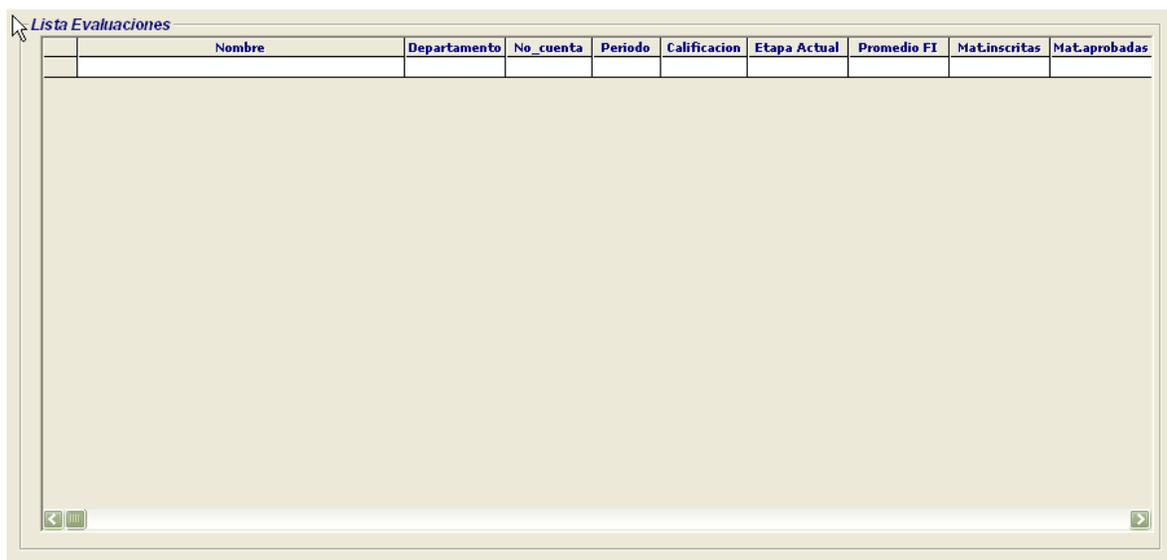


The screenshot shows a search interface titled "Parámetros de Búsqueda". It contains six search criteria, each with a checkbox and a corresponding input field or dropdown menu:

- Nombre: Text input field.
- Apellido(s): Text input field.
- Promedio FI: Dropdown menu with "Entre 9 y 10" selected.
- Departamento: Dropdown menu with "CSC" selected.
- Período: Dropdown menu with "2004-1" selected.
- Etapa: Dropdown menu with "1er ingreso" selected.

Figura 1.1.5 Búsqueda por parámetros

**Área de datos:** En esta área se mostraran los resultados de cargar datos, búsquedas o bien nueva evaluación.



The screenshot shows a data area titled "Lista Evaluaciones". It contains a table with the following columns:

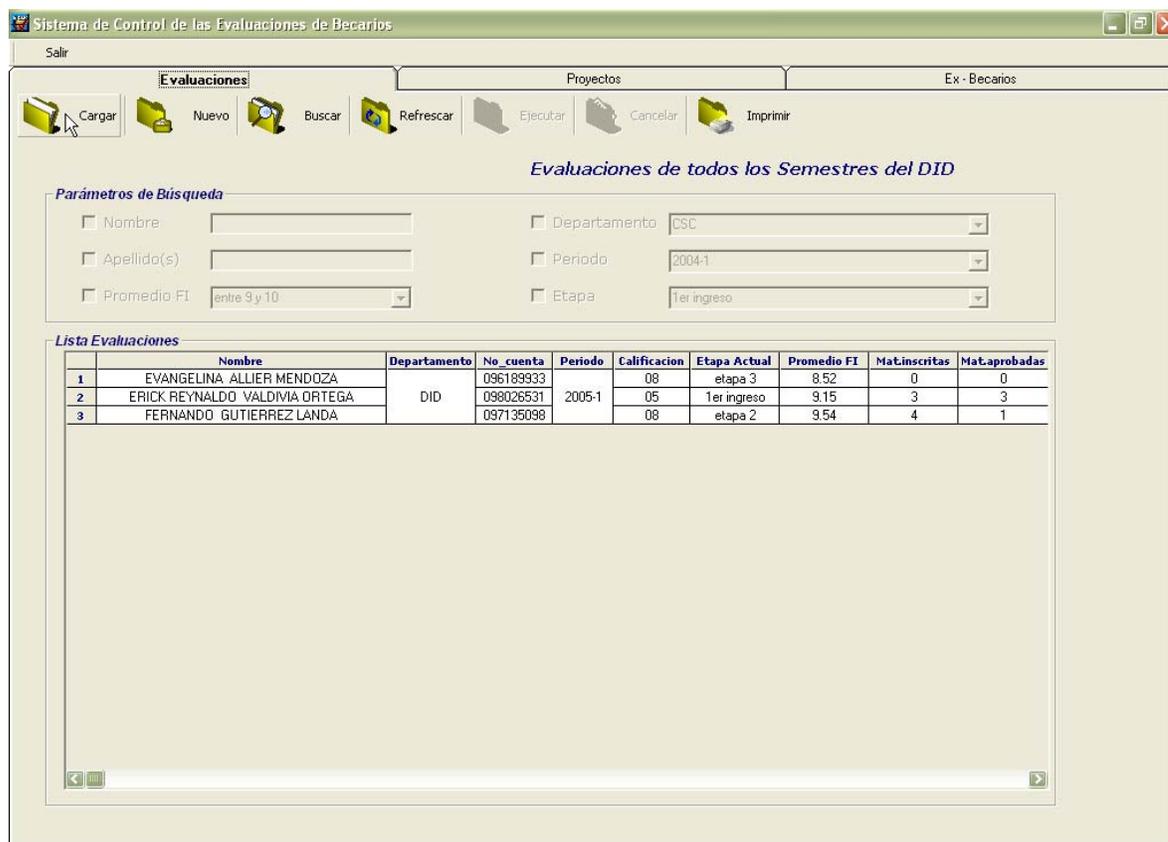
	Nombre	Departamento	No_cuenta	Periodo	Calificacion	Etapas Actual	Promedio FI	Mat.Inscritas	Mat.aprobadas
[Empty table body]									

Figura 1.1.6 Área de datos

## 2.- Utilizando la Barra de Herramientas

Como ya se mencionó esta barra nos va a servir para realizar la manipulación de los datos del sistema. Veamos cada botón:

*2.1.- Cargar:* Al dar clic nos aparecerán en el área de datos aquellos becarios que cuenten con al menos una evaluación dentro de la base, esto con el propósito de observar su desempeño a lo largo de los semestres de manera rápida.



Como podemos observar, en este ejemplo sólo existen para el DID tres becarios evaluados para el periodo 2005-1 (4 columna de izquierda a derecha).

2.2.- *Nuevo registro*: Al dar clic a este botón, nos aparecerán aquellos becarios que faltan de ser evaluados para el periodo actual.

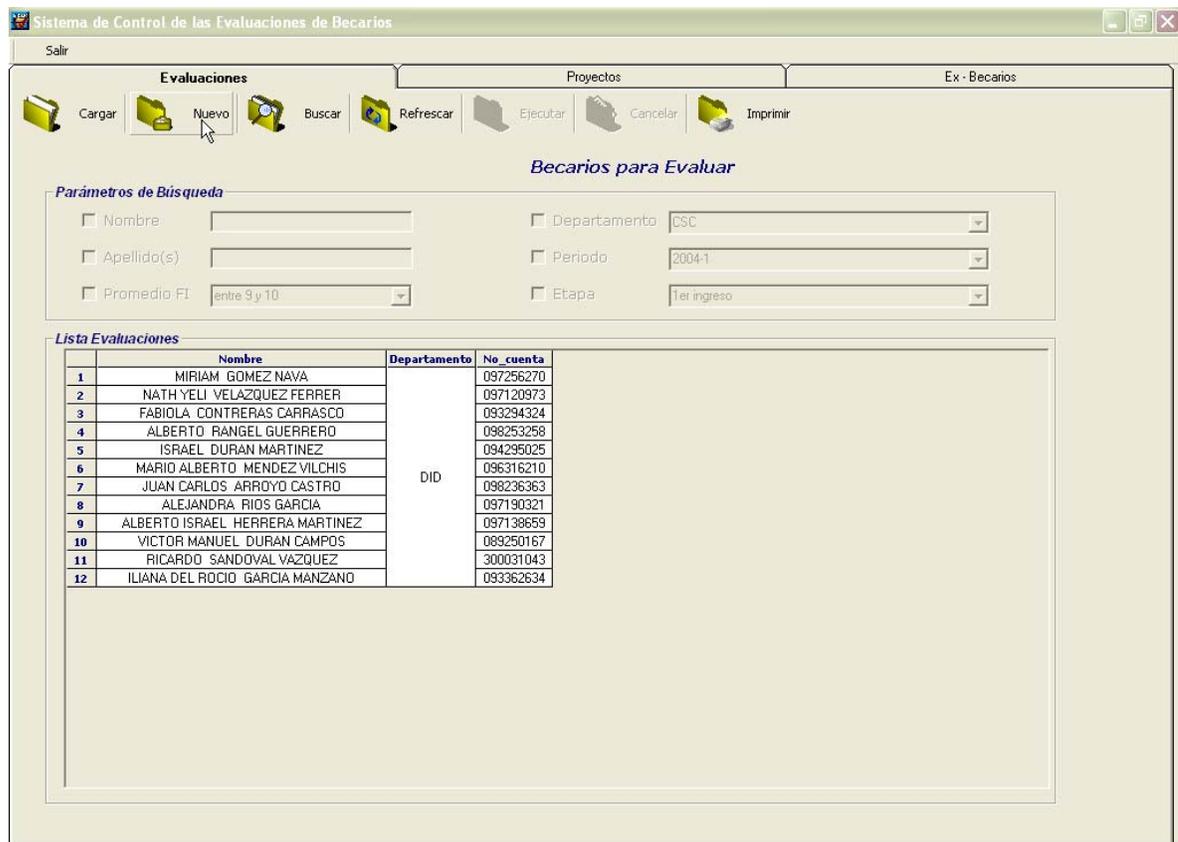


Figura 2.2.1 Becarios que faltan de evaluar

Podemos observar que en este caso se tienen 12 becarios del DID.

2.3-. *Búsqueda*: En el momento que damos clic a buscar se habilita la sección de este nombre, así como las cajas de texto, *combos* y *check boxes* que la conforman para permitirnos realizar una búsqueda por parámetros, así también se habilita el botón de Ejecutar.

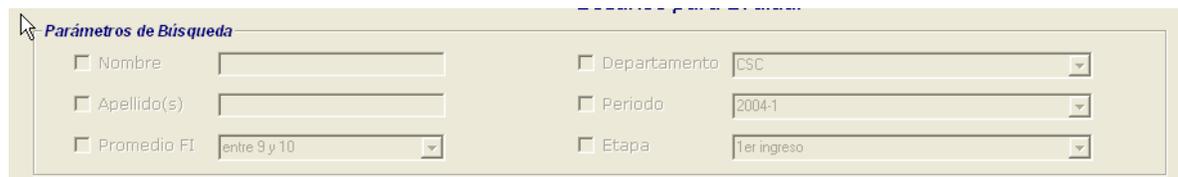


Figura 2.3.1 Sección de búsqueda antes de dar clic al botón de Buscar

**Parámetros de Búsqueda**

<input type="checkbox"/> Nombre	<input type="text"/>	<input type="checkbox"/> Departamento	CSC
<input type="checkbox"/> Apellido(s)	<input type="text"/>	<input type="checkbox"/> Periodo	2004-1
<input type="checkbox"/> Promedio FI	entre 9 y 10	<input type="checkbox"/> Etapa	1er ingreso

Figura 2.3.2 Sección de búsqueda luego de dar clic al botón de Buscar.

En las figuras anteriores podemos apreciar el antes y el después de dar clic al botón de Búsqueda.

Sistema de Control de las Evaluaciones de Becarios

Salir

Evaluaciones | Proyectos | Ex - Becarios

Cargar | Nuevo | **Buscar** | Refrescar | Ejecutar | Cancelar | Imprimir

**Becarios para Evaluar**

**Parámetros de Búsqueda**

<input type="checkbox"/> Nombre	<input type="text"/>	<input type="checkbox"/> Departamento	CSC
<input type="checkbox"/> Apellido(s)	<input type="text"/>	<input type="checkbox"/> Periodo	2004-1
<input type="checkbox"/> Promedio FI	entre 9 y 10	<input type="checkbox"/> Etapa	1er ingreso

**Lista Evaluaciones**

	Nombre	Departamento	No_cuenta
1	MIRIAM GOMEZ NAVA	DID	097256270
2	NATH YELI VELAZQUEZ FERRER		097120973
3	FABIOLA CONTRERAS CARRASCO		093294324
4	ALBERTO RANGEL GUERRERO		098253258
5	ISRAEL DURAN MARTINEZ		094295025
6	MARIO ALBERTO MENDEZ VILCHIS		096316210
7	JUAN CARLOS ARROYO CASTRO		098236363
8	ALEJANDRA RIOS GARCIA		097190321
9	ALBERTO ISRAEL HERRERA MARTINEZ		097138659
10	VICTOR MANUEL DURAN CAMPOS		089250167
11	RICARDO SANDOVAL VAZQUEZ		300031043
12	ILIANA DEL ROCIO GARCIA MANZANO		093362634

Figura 2.3.3 Aspecto de la pantalla luego de dar clic a Buscar.

En la figura 2.3.3 podemos apreciar que el botón de Ejecutar que previamente había aparecido deshabilitada, ahora se encuentra habilitado para poder como su nombre lo indica ejecutar la búsqueda.

A continuación se debe escoger el (los) parámetro(s) de búsqueda, estos pueden ser:

- *Por el nombre del becario:* Este debe ser escrito de manera idéntica a como el becario fue registrado, si cuenta con dos nombres se deben escribir. No importa si son mayúsculas o minúsculas.
- *Por los apellidos del becarios:* Se deben escribir los apellidos completos del becario.
- *Por promedio de la Facultad:* Se pueden escoger tres intervalos: entre 7 y 8, 8 y 9, 9 y 10.
- *Por departamento:* Esta opción sólo se encuentra habilitada al administrador del sistema debido a que es el único que puede visualizar las evaluaciones de otros departamentos.
- *Por periodo:* Por el periodo (2005-1 por ejemplo ) en el cuál se haya realizado las evaluaciones.
- *Por etapa:* Se podrán buscar a todos los becarios por la etapa tenían al momento de hacer la evaluación.

*2.4.- Refrescar:* Este botón se utiliza para actualizar la consulta en el momento dado de que se realice algún cambio dentro de la base de datos, al refrescarla se cargaran nuevamente los becarios que se encuentren ya evaluados, deshabilitándose el botón de Ejecutar, y la sección de búsqueda por parámetros.

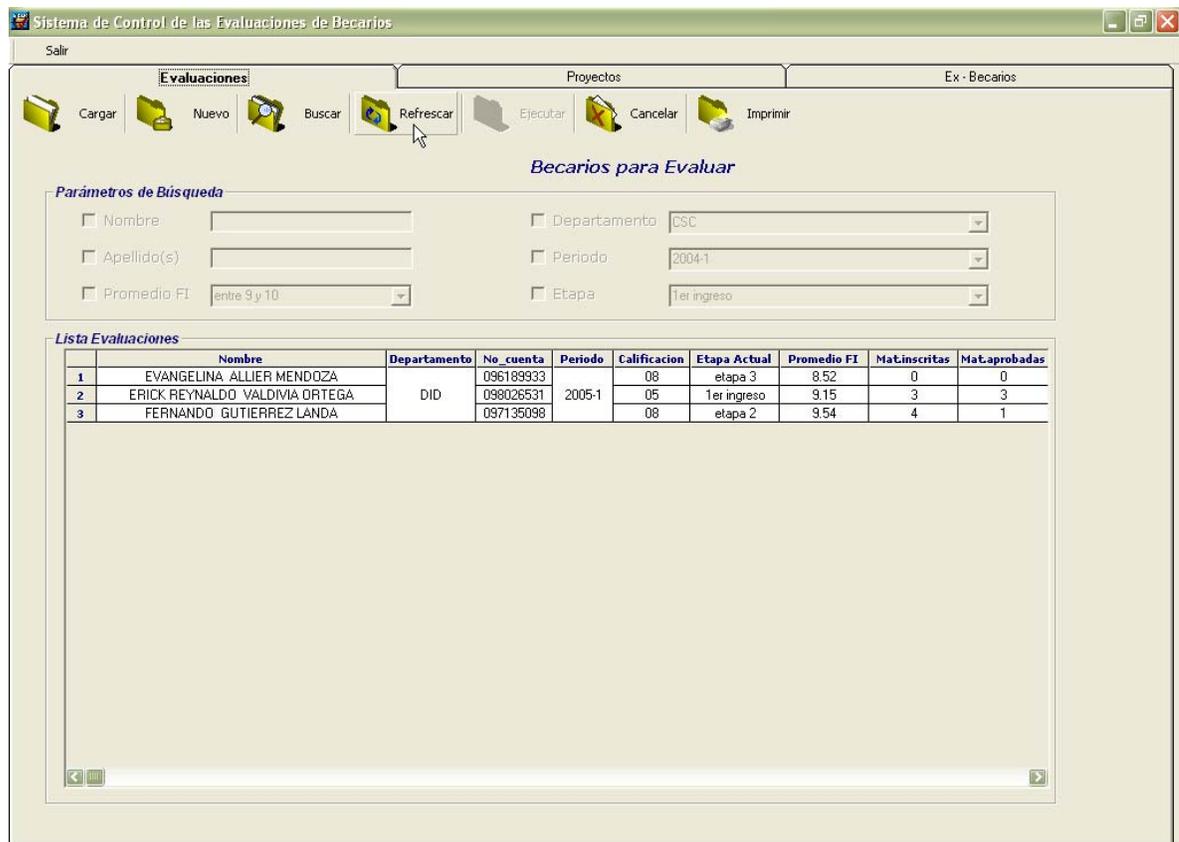


Figura 2.4.1 Refrescando la consulta

## 2.5.- Ejecutar: Funciona para realizar la búsqueda por parámetros.

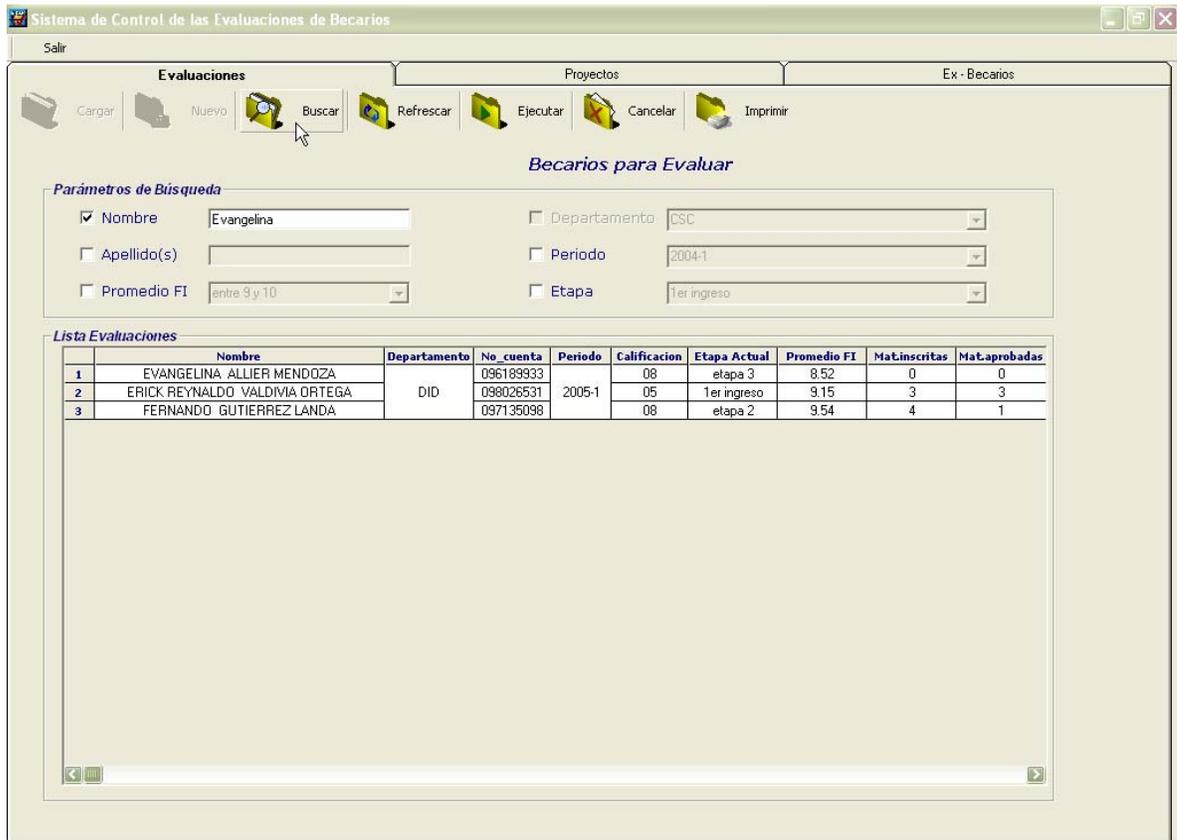


Figura 2.5.1 Realizando una búsqueda por nombre

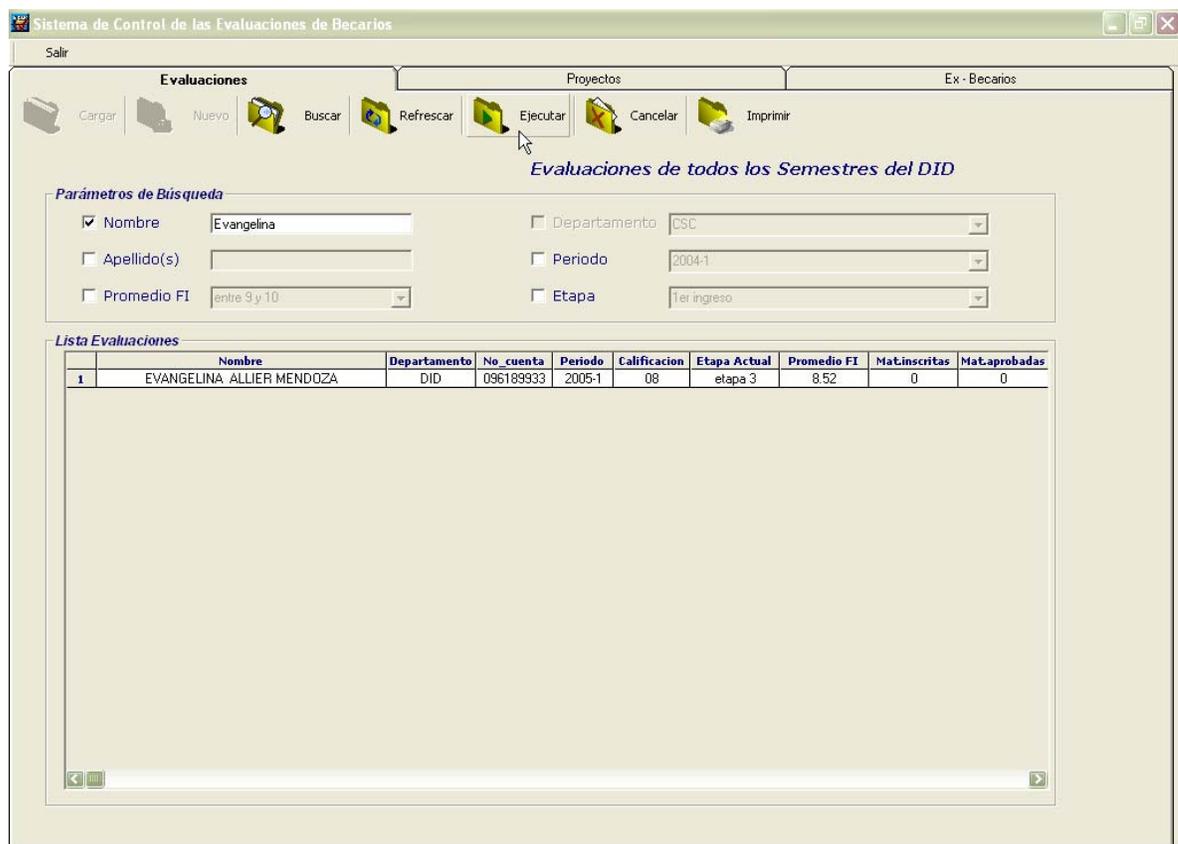


Figura 2.5.2 La pantalla luego de dar clic a Ejecutar.

En la figura 2.5.2, observamos que se pidió específicamente a un becario por su nombre y al dar clic a ejecutar nos trajo ese registro.

En caso de que no se cuente con un registro dentro de la base o bien se haya escrito el nombre o apellidos de manera incorrecta se mostrará la siguiente advertencia:

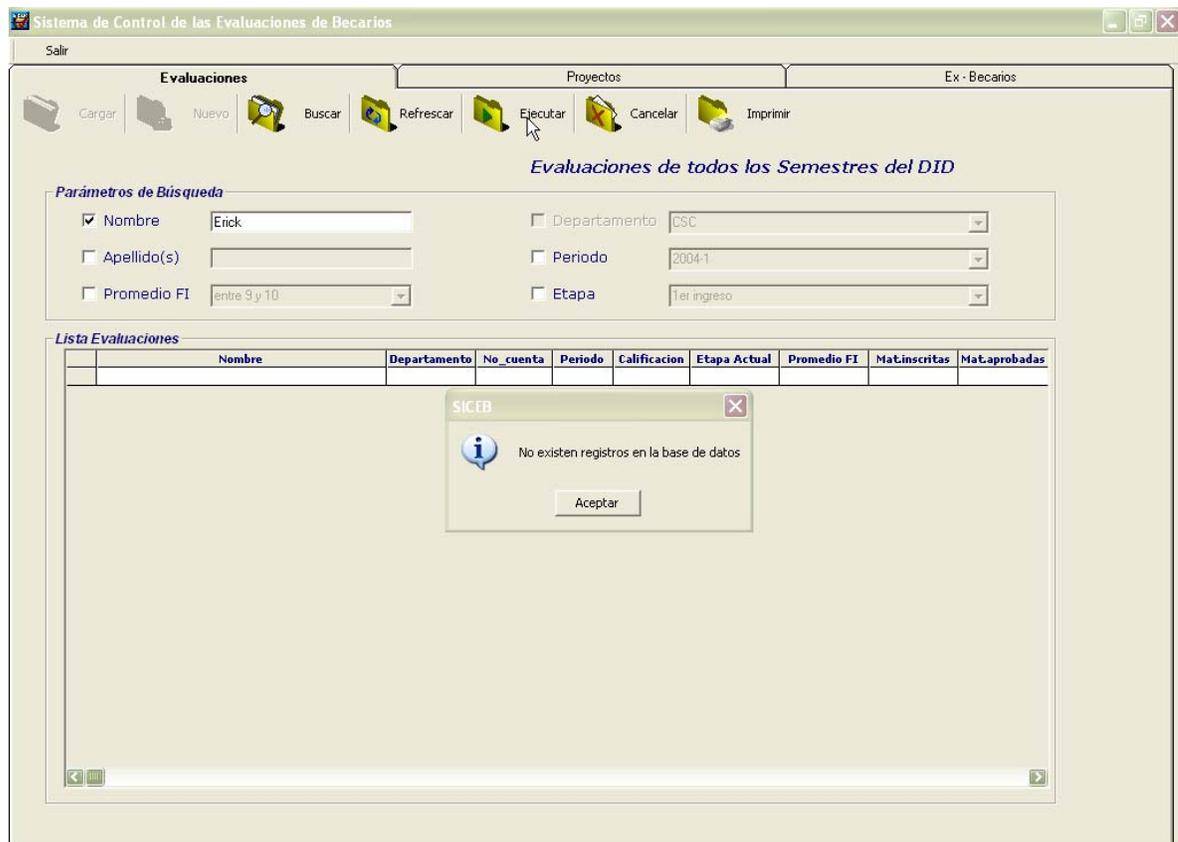


Figura 2.5.3 No existe un registro o bien datos erróneos.

En el ejemplo de la figura 2.5.3 el nombre del becario no fue escrito completo ya que tiene dos nombres, es por esa razón que no muestra ningún resultado al realizar la búsqueda y marca que “No existen registros en la base de datos”. Corrigiendo el error tendríamos:

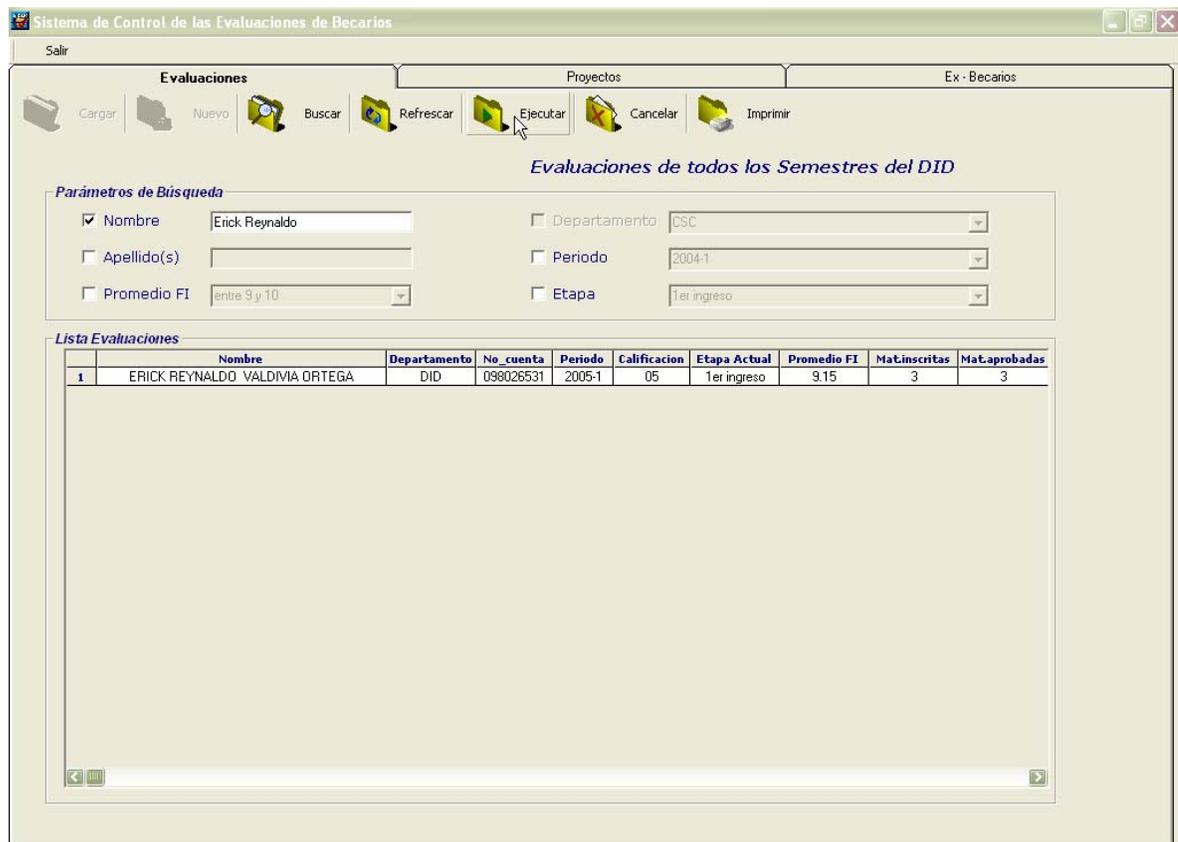


Figura 2.5.4 Escribiendo completo el nombre del becario.

Y observamos que ahora si nos trae un registro existente en la base.

2.6.- *Cancelar*: Sirve para deshabilitar la búsqueda por parámetros esto es en caso de que por algún motivo deseemos suspenderla incluso cuando ya la hayamos ejecutado.

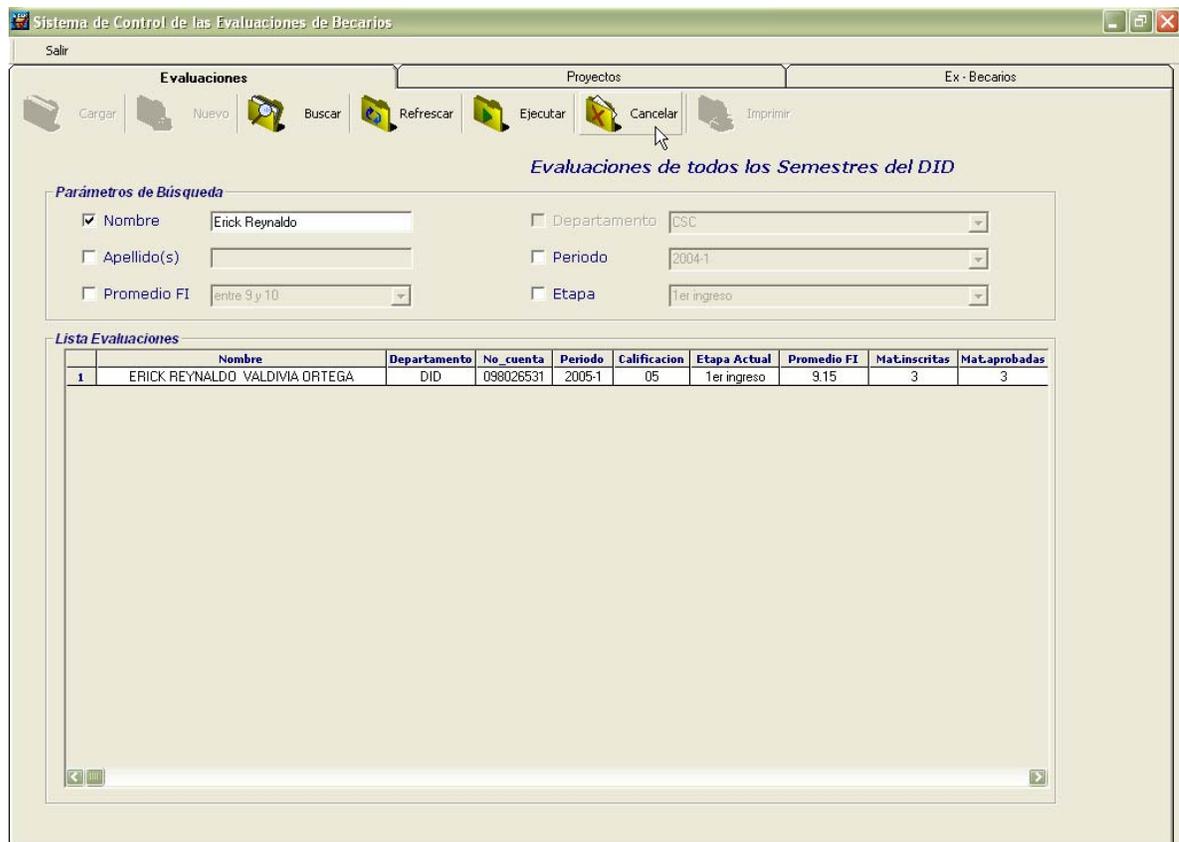


Figura 2.6.1 Cancelando una búsqueda.

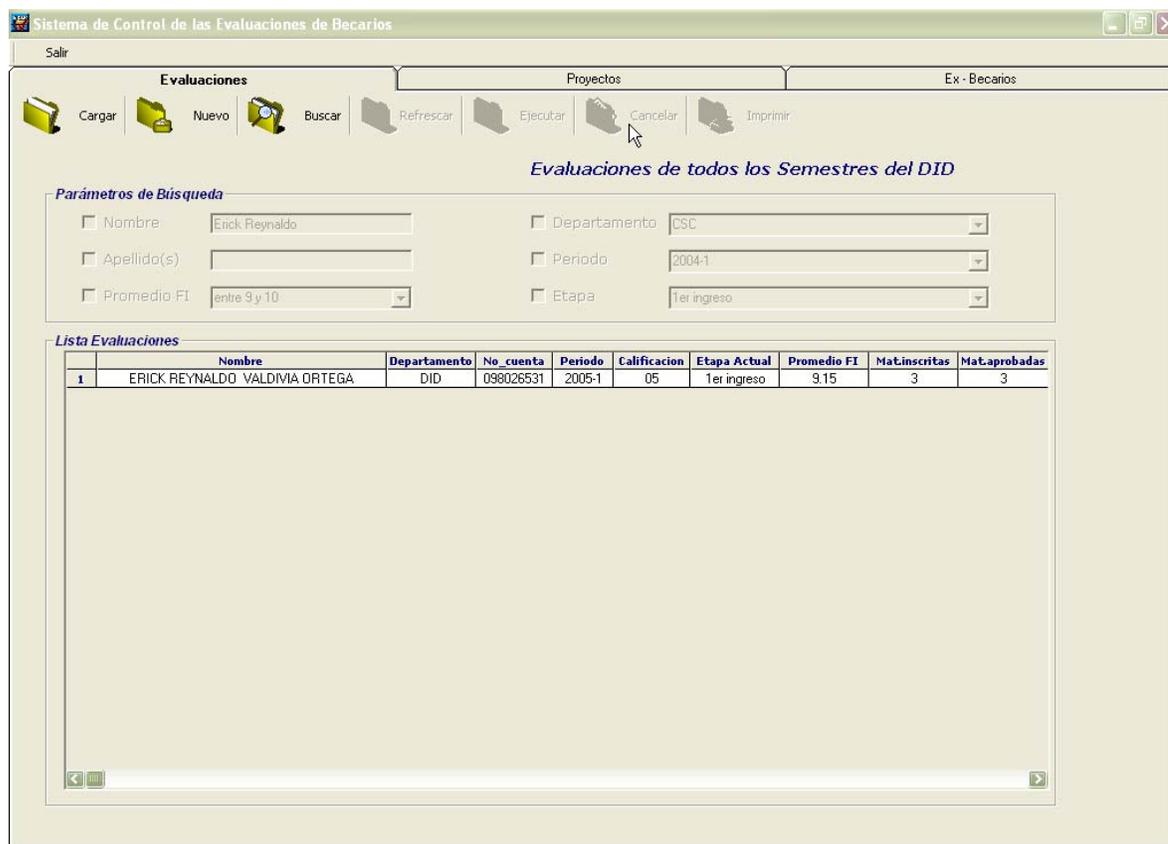


Figura 2.6.2 Búsqueda Cancelada

En la figura 2.6.2 se observa que una vez que se dio clic al botón de Cancelar, tanto el de Búsqueda y el de Ejecutar son deshabilitados.

## 2.7.- Imprimir

Dentro de la pantalla principal del SICEB, si existen registros dentro del Sistema se permitirá realizar la impresión del reporte; actualmente esta opción sólo esta habilitada para el administrador del sistema.

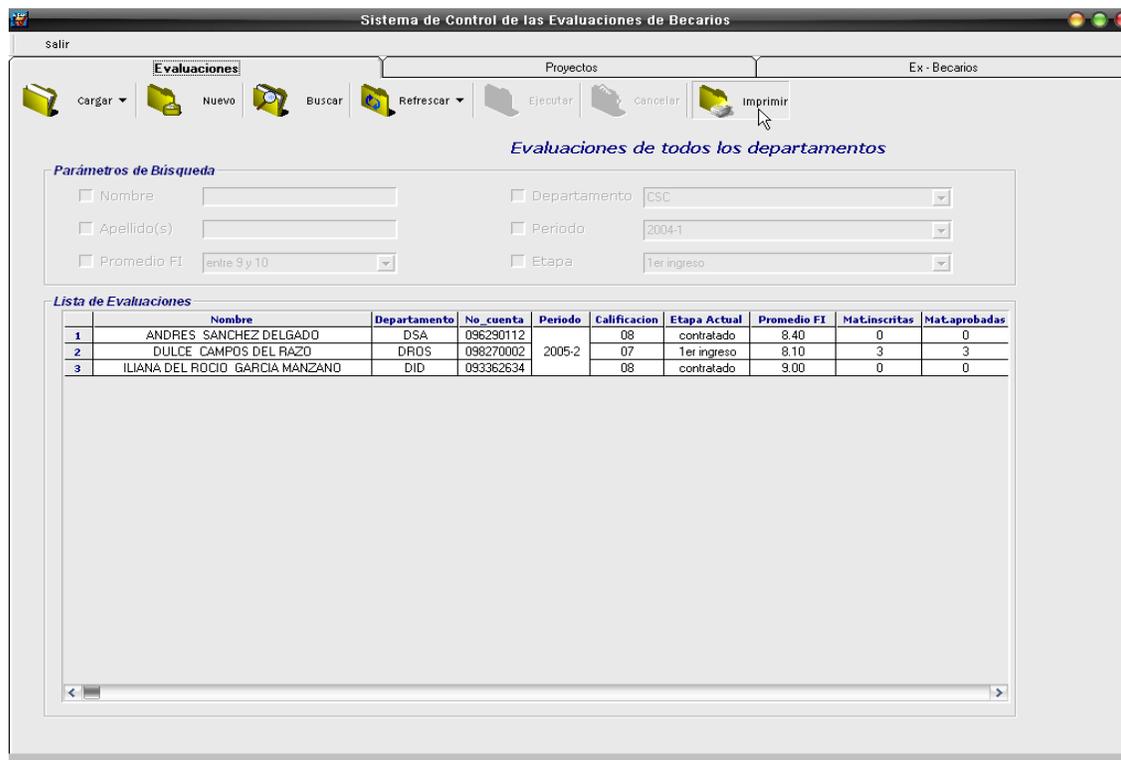


Figura 2.7.1 Si existen registros en la base de datos, el botón de imprimir se habilita, y permite mostrar la pantalla de impresión

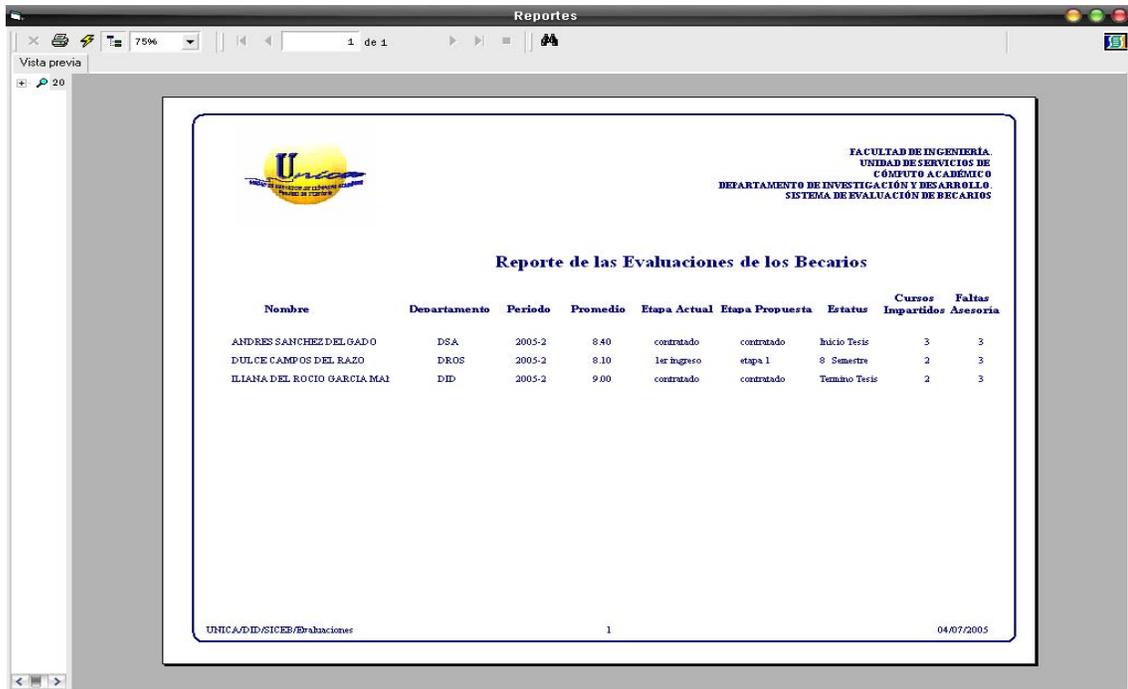


Figura 5.7.2 Pantalla de impresión

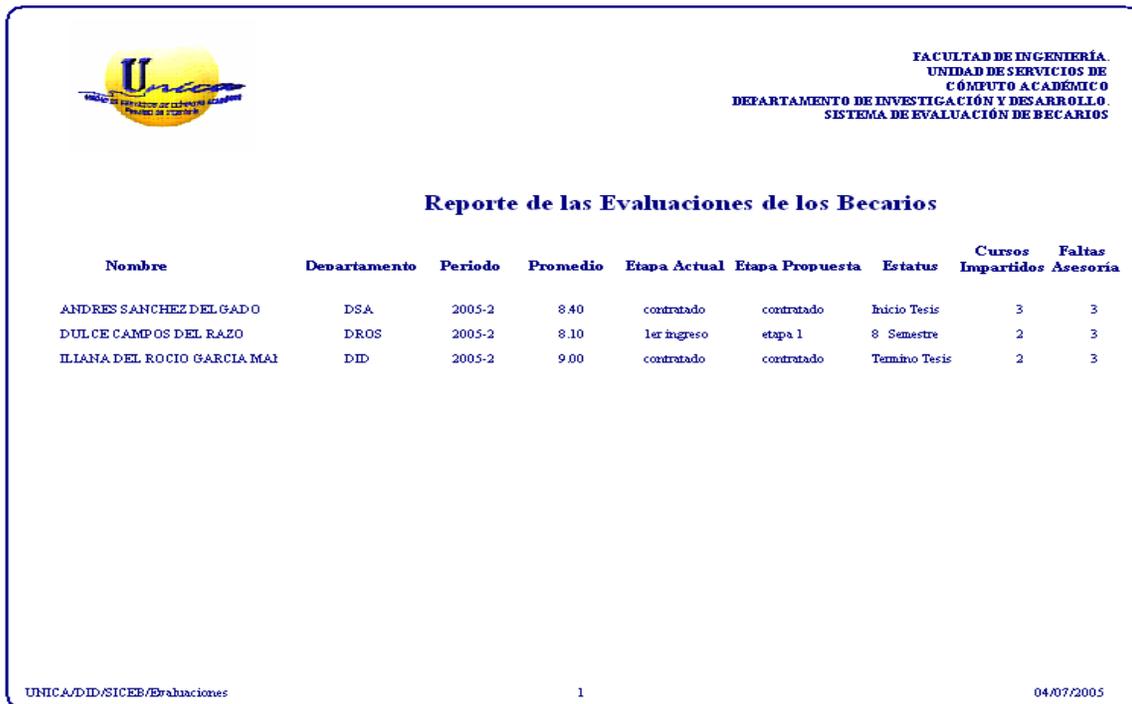


Figura 5.7.3 Vista del reporte una vez impreso.

Evaluaciones Individuales

Facultad de Ingeniería  
Secretaría General  
Unidad de Servicios de Cómputo Académico

**Evaluación de Becarios**

Nombre del Becario: ANDRES SANCHEZ DELGADO		No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005-2		Estatus: Inicio Tesis
Etapas actual: contratado	Etapas propuesta: contratado	Evaluación: 08	

Característica	Desarrollo	Cantidad
Rendimiento Académico	Materias Inscritas	0
	Materias Aprobadas	0
	Materias Reprobadas	0
Asistencia y Puntualidad en UNICA	Asistencias	100
	Retardos	1
	Faltas	14
Iniciativa	Disposición para colaborar	Muchas veces
	Realización de propuestas	Algunas veces
	Compromiso con UNICA	Algunas veces
Cursos	Intersemestrales	1
	Internos	0
	Especiales	0
	Prebecarios	1
	Sábados y Domingos	1
	Semestrales	0

*Figura 5.6.8.4 Pantalla de impresión del reporte individual.*

Evaluación de Becarios			
Nombre del Becario: ANDRÉS SANCHEZ DELGADO		No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005.2	Estatus: Inicio Tesis	
Etapas actual: contratado	Etapas propuestas: contratado	Evaluación: 08	
Categorías	Detalle	Cantidad	
Rendimiento Académico	Materias Insatisfechas	0	
	Materias Aprobadas	0	
	Materias Reprobadas	0	
Asistencia y Puntualidad en UNICA	Asistencias	100	
	Retardos	1	
	Faltas	14	
Iniciativa	Disposición para colaborar	Muchas veces	
	Realización de propuestas	Algunas veces	
	Compromiso con UNICA	Algunas veces	
Cursos	Intereses en cursos	1	
	Intensos	0	
	Especiales	0	
	Prácticos	1	
	Sábados y Domingos	1	
	Seminarios	0	
	Elaboración de Manuales	2	
	Cursos Asistidos	0	
	Nuevas propuestas	1	
	Calidad en el trabajo	Responsabilidad	Algunas veces
Confiable		Muchas veces	
Actitud de Servicio		Algunas veces	
Proyectos	Líder de Proyectos	0	
	Participación en Proyectos	0	
	Oportunidad de Avance	Mínimo	
	Resultados	Regular	
Asesorías	Asistencias	100	
	Retardos	2	
	Faltas	3	
Comentarios			
Notas	El becario está a consideración por el número de faltas en asesoría		

UNICA/DID/SICEB 12/08/2001

Figura 5.6.8.5 Vista del reporte ya impreso

### 3.- Visualizar las Evaluaciones

Cuando se da clic al botón de Cargar, se visualizan las evaluaciones de una manera general, mostrándolas con los datos más relevantes y específicos para darse una idea del desempeño general.

Si se requiere visualizar la evaluación completa de un becario se hace lo siguiente:

- 1) Se da clic a Cargar, para visualizar las evaluaciones existentes
- 2) Sobre el nombre del becarios del cual que se quiere profundizar se da doble clic (figura 3.1)
- 3) Aparecerá la pantalla de Evaluación que se muestra en la figura 3.2.

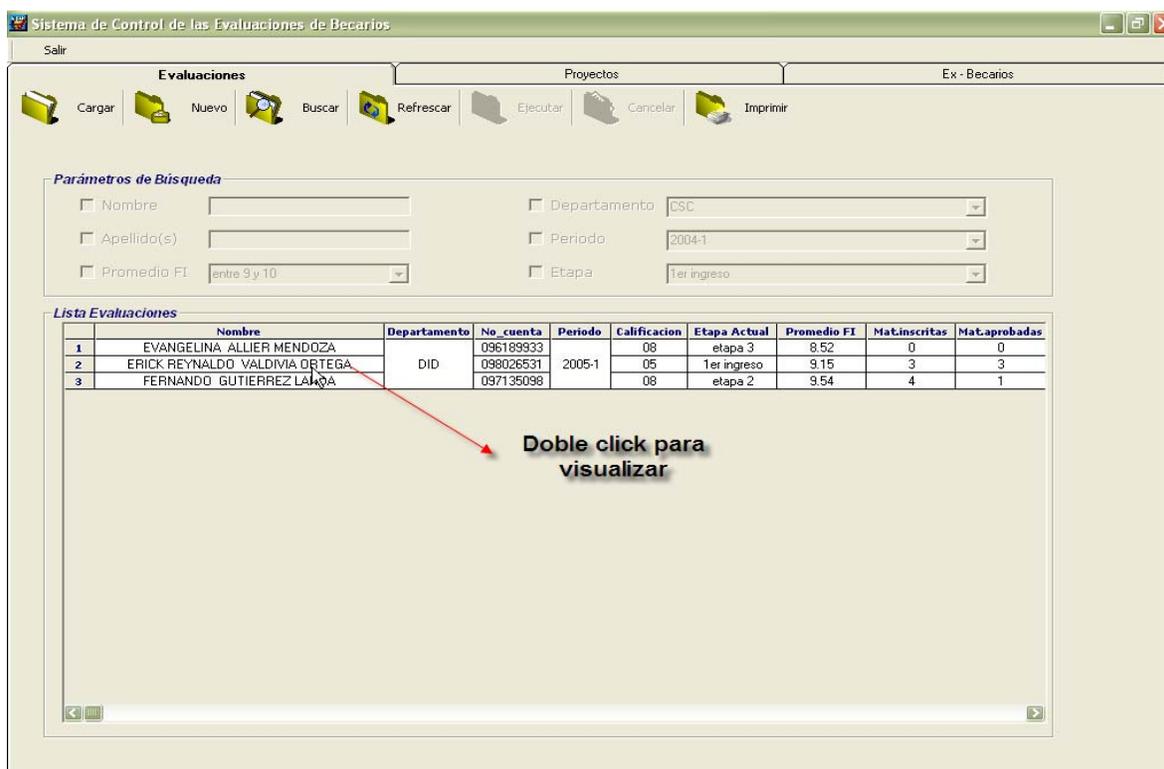


Figura 3.1 Doble clic para profundizar

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: ERICK REYNALDO VALDIVIA ORTEGA No. Cta. 098026531 Promedio: 9.15

Departamento: DID Semestre que concluyo: 2005-1 Período: 2004/8/15 a 2004/1/6 Evaluación: 05

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 08/01/2005

**Evaluación**

<p><b>Rendimiento Académico</b></p> <p>Materias inscritas: 3</p> <p>Materias aprobadas: 3</p> <p>Materias reprobadas: 0</p> <p>Estatus: Lleva Materias</p>	<p><b>Cursos</b></p> <p>Intersemestrales: 0</p> <p>Sábados y Domingos: 0</p> <p>Internos: 0</p> <p>Especiales: 0</p> <p>Semestrales: 0</p> <p>Cursos Asistidos: 0</p> <p>Prebecarios: 1</p> <p>Elaboración de Manuales: 3</p> <p>Nuevas Propuestas: 1</p>	<p><b>Proyectos</b></p> <p>Lider de Proyectos: 0</p> <p>Proyectos: 2</p> <p>Oportunidad de Avance: Oportuno</p> <p>Resultados: Oportunos</p>	<p><b>Asesorías</b></p> <p>Asistencia: 90</p> <p>Retardos: 1</p> <p>Faltas: 2</p>
<p><b>Asistencia y Puntualidad</b></p> <p>Asistencia: 75</p> <p>Retardos: 1</p> <p>Faltas: 5</p> <p><small>Nota: los días hábiles para este periodo fueron 80</small></p>	<p><b>Calidad en el Trabajo</b></p> <p>Responsabilidad: Siempre</p> <p>Confiabilidad: Siempre</p> <p>Actitud de Servicio: Siempre</p>	<p><b>Notificaciones</b></p>	<p><b>Comentarios</b></p> <p>Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad.</p>
<p><b>Iniciativa</b></p> <p>Disposición para colaborar: Algunas veces</p> <p>Realización de propuestas: Algunas veces</p> <p>Compromiso con UNICA: Algunas veces</p>			

Primero Anterior Siguiente Ultimo

Figura 3.2 Evaluación

### 3.1.- Elementos de la Pantalla de Evaluación:

Una vez dentro de la pantalla que muestra la figura 3.2, podemos visualizar todos los campos de una evaluación que ya fue realizada dentro de ellos tenemos:

### ***3.1.1.- Datos personales:***

Esta sección muestra los datos de cada becarios como son: nombre (s), apellidos, número de cuenta, promedio de la facultad, departamento, semestre que se evaluó (semestre que concluyó), periodo que comprendió (día, mes y año de inicio y fin), evaluación , etapa en la que estaba durante el semestre que se realizó la evaluación, etapa propuesta a la que pasaría el becario, generación a la que pertenece y fecha de ingreso a la unidad.

### ***3.1.2.- Evaluación***

Esta se divide en nueve aspectos:

*Rendimiento Académico:* Donde se contemplan el número de materias inscritas, aprobadas, y reprobadas del becario, así como el caso en que ya haya terminado los créditos y el estatus que es en relación a su tesis, si ya la inició, la terminó, aún no la inicia, o bien ya se tituló.

*Asistencia y Puntualidad:* En cuanto a sus horas de trabajo dentro de la Unidad es decir las asistencias, faltas y retardos que tenga el becario a sus horas de trabajo.

*Iniciativa:* Es en cuanto al desempeño y las ganas de trabajar del becario teniendo: su disposición para colaborar, realización de propuestas de trabajo y compromiso con UNICA.

*Cursos:* En esta sección se visualizan todos los cursos que el becarios haya dado a lo largo del semestre, indicando de que tipo fueron: semestrales, intersemestrales, sábados y domingos, internos, especiales, de prebecarios, además como los cursos a los que asistió (en general ), cuántos manuales elaboró y cuantas propuestas de nuevos cursos dio.

*Calidad en el Trabajo:* Si el becario es responsable, confiable y tiene actitud de servicio ante sus obligaciones dentro de la Unidad.

*Proyectos:* Se consideran los proyectos en los que el becario sea líder, en los que participe, el avance que tuvo y los resultados que entrego.

*Asesorías:* Se reportan las faltas, asistencias y retardos.

*Notificaciones:* Este campo puede aparecer vacío si el becario no incurrió en ninguna falta al reglamento (ver Evaluando Becarios ).

*Comentarios:* Sección que sirve para aclarar algunos puntos o bien realizar comentarios favorables, explicaciones etc., del desempeño del becario.

### **3.1.3.- Barra de Herramientas de las Evaluaciones**

Así como la pantalla principal del sistema, en la pantalla de Evaluación se cuenta con una barra de herramientas para poder manipular las evaluaciones.



*Figura 3.1.3.1 Barra de Herramientas de las Evaluaciones.*

Esta está conformada por ocho botones los cuales tienen como función:

- *Regresar:* Nos envía a la pantalla principal, aquellas modificaciones que se realicen serán perdidas si no son guardados.
- *Evaluar:* Botón que se habilita cuando se va a realizar una nueva evaluación o bien cuando se va a modificar una existente, sirve para de acuerdo a los campos de la evaluación generar notificaciones o bien validar que no falte algún campo del registro.
- *Guardar:* Permite guardar el registro en la base de datos una vez que esta todo completo; se activa cuando el evento del botón evaluar se lleva a cabo sin que falte ningún campo.
- *Modificar:* Permite corregir o cambiar algún(os) dato(s) de la evaluación, sólo se pueden modificar las evaluaciones del periodo que se este evaluando.
- *Refrescar:* Sirve para actualizar la consulta si se realizó alguna modificación y ver los cambios hechos.
- *Borrar:* Permite eliminar el registro actual.

- *Cancelar*: En caso de que no queramos realizar alguna modificación.
- *Imprimir*: Generar el reporte de la evaluación del becario.

### **3.1.4.- Barra de Navegación**

Esta barra se encuentra ubicada en la parte inferior de la ventana de evaluación y nos permite navegar o recorrer los registros existentes de manera que no tengamos que estar regresando a la pantalla principal cada vez que queramos ver la evaluación de otro becario. Cuenta con los botones de:

- *Primero*: Del registro en el que nos encontremos nos mueve al primero.
- *Anterior*: El registro inmediato anterior al actual.
- *Siguiente*: El registro inmediato posterior al actual.
- *Último*: Del registro en el que nos encontramos nos mueve al último.



*Figura 3.1.4.1 Barra de navegación*

### **3.1.5.- Botón de Evaluación Especial**

Algunos departamentos solicitaron que aparte de la evaluación general deseaban contar con una evaluación especial para las funciones que realizan. Estos son: el DROS y el DSC. Si bien el botón aparece para los demás departamentos, este es inservible y no afecta a la ejecución del programa.

Para visualizar la evaluación especial se da clic en el botón que lleva ese nombre y dependiendo del departamento se visualizará una nueva pantalla

3.1.5.1.- Pantalla de la Evaluación Especial del DROS



Figura 3.1.5.1.1 Vista de la Evaluación Especial del DROS

The screenshot displays a web interface titled "Evaluación Especial del DROS". At the top, there is a navigation bar with five buttons: "Regresar" (with a left arrow), "Modificar" (with a pencil icon), "Guardar" (with a floppy disk icon), "Cancelar" (with a trash can icon), and "Refrescar" (with a circular refresh icon). Below the navigation bar, the main content area is titled "Evaluación DROS" and contains five distinct sections, each with a title and a table of metrics.

Número de atención de solicitud de eventos de soporte técnico atendidos	
Número de Atendidos	23
Número de Atendidos Parcialmente	2

Realización de Respaldos de Servidores	
Número de respaldos	1
Cantidad de información en MB respaldada	6

Número de sistemas para la administración al que le da mantenimiento	
Cantidad	4
Actualización del sistema	3

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo	
Cantidad	1

Número de atención a incidentes de operación de servidores y redes fuera del horario	
Cantidad de Eventos	3
Número de horas invertidas	4

Figura 3.1.5.1.2 Vista de la Pantalla de Evaluación Especial

### 3.1.5.2.- Pantalla de la Evaluación Especial del DSC

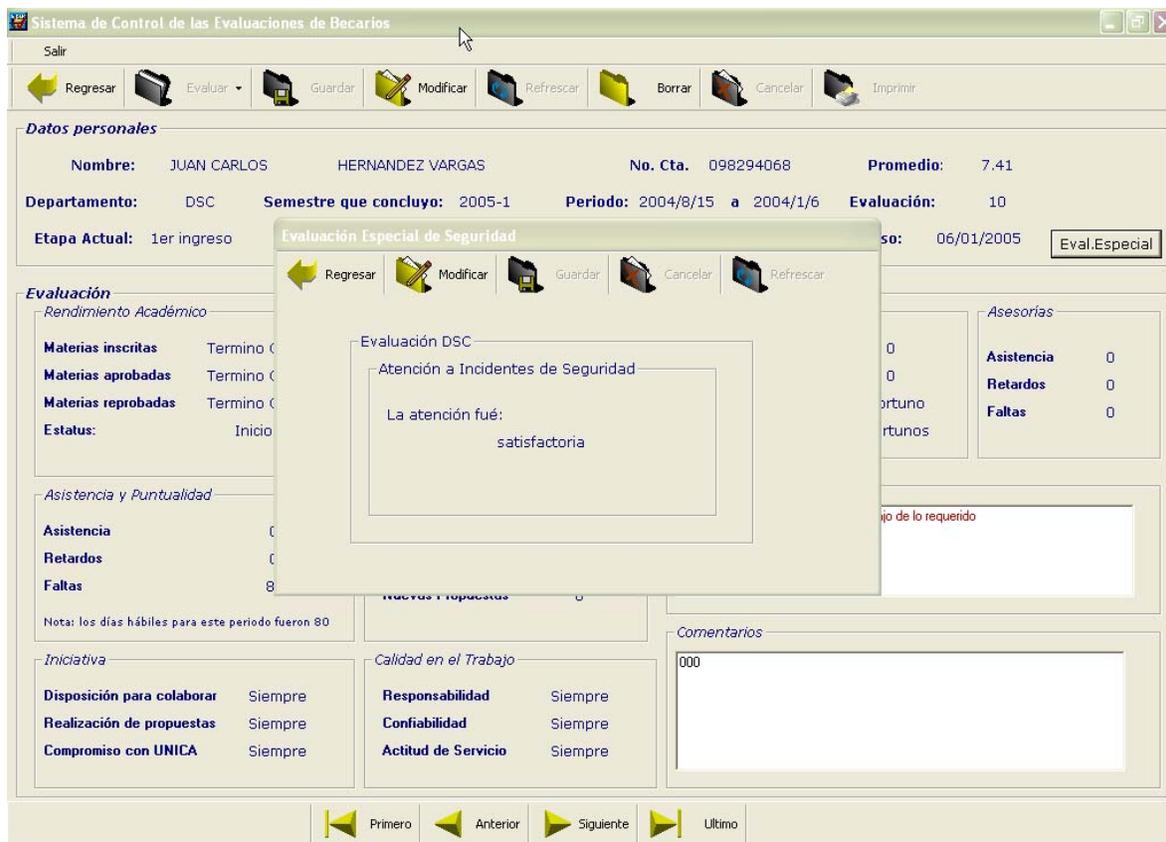


Figura 3.1.5.2.1 Vista de la evaluación especial del DSC



Figura 3.1.5.2.2 Vista de la pantalla de Evaluación Especial del DSC sola.

### 3.2.- Modificando una Evaluación ya existente.

Estando dentro de esta pantalla para visualizar la evaluación de un becario es posible que se desee modificar algún dato. Esto lo conseguimos al dar clic sobre el botón Modificar de la barra de herramientas de la pantalla



Figura 3.2.1 Botón de Modificar

Sistema de Control de las Evaluaciones de Becarios

Salir

Regresar Evaluar Guardar **Modificar** Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: ERICK REYNALDO VALDIVIA ORTEGA No. Cta. 098026531 Promedio: 9.15  
 Departamento: DID Semestre que concluye: 2005-1 Período: 2004/8/15 a 2004/1/6 Evaluación: 05  
 Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 08/01/2005 Eval.Especial

**Evaluación**

Rendimiento Académico		Cursos		Proyectos		Asesorías	
Materias inscritas	3	Intersemestrales	0	Lider de Proyectos	0	Asistencia	90
Materias aprobadas	3	Sábados y Domingos	0	Proyectos	2	Retardos	1
Materias reprobadas	0	Internos	0	Oportunidad de Avance	Oportuno	Faltas	2
Estatus:	Lleva Materias	Especiales	0	Resultados	Oportunos		
		Semestrales	0				
		Cursos Asistidos	0				
		Prebecarios	1				
		Elaboración de Manuales	3				
		Nuevas Propuestas	1				

Asistencia y Puntualidad

Asistencia 75  
Retardos 1  
Faltas 5

Nota: los días hábiles para este periodo fueron 80

Iniciativa

Disposición para colaborar Algunas veces  
Realización de propuestas Algunas veces  
Compromiso con UNICA Algunas veces

Calidad en el Trabajo

Responsabilidad Siempre  
Confiabilidad Siempre  
Actitud de Servicio Siempre

Notificaciones

Comentarios

Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad.

Primero Anterior Siguiente Ultimo

Figura 3.2.2 Estado de la pantalla antes de Modificar

Una vez dado clic a este botón obtendremos la siguiente pantalla:

Figura 3.2.3 Pantalla luego de dar clic al botón de Modificar

Si observamos la figura 3.2.3, notaremos que muchos de los campos que aparecían como etiquetas, han cambiado a ser cajas de texto o bien cajas de opción múltiples, pudiendo modificar el (los) dato(s) que se deseen.

Una vez realizada la modificación se dará clic en el botón de Evaluar, para realizar la validación de los campos:

Sistema de Control de las Evaluaciones de Becarios

Salir

Regresar Evaluar Guardar Modificar Refrescar Borrar Cancelar Imprimir

**Datos personales**

Nombre: ERICK REYNALDO VALDIVIA ORTEGA No. Cta. 098026531 Promedio: 9,15

Departamento: DID Semestre que concluyo: 2005-1 Período: 2004/8/15 a 2004/1/6 Evaluación: 05

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 08/01/2005 Eval, Especial

**Evaluación**

*Rendimiento Académico*

Materias inscritas Terminó Créditos

Materias aprobadas Terminó Créditos

Materias reprobadas 0

Estatus: Inicio Tesis

*Asistencia y Puntualidad*

Asistencia 75

Retardos 1

Faltas 5

Nota: los días hábiles para este periodo fueron 80

*Calidad en el Trabajo*

Responsabilidad Siempre

Confiabilidad Siempre

Actitud de Servicio Siempre

*Cursos*

Intersemestrales 0

Sábados y Domingos 0

Internos 0

Especiales 0

Semestrales 0

Cursos Asistidos 0

Prebecarios 1

Elaboración de Manuales 3

Nuevas Propuestas 1

*Proyectos*

Lider de Proyectos 0

Proyectos 2

Oportunidad de Avance Oportuno

Resultados Oportunos

*Asesorías*

Asistencia 90

Retardos 1

Faltas 3

*Notificaciones*

Se cambió el número de faltas de 2 a 3

*Comentarios*

Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad.

Primero Anterior Siguiente Ultimo

Figura 3.2.4 Realizando una modificación y dando clic para evaluar

Figura 3.2.5 Se realizó la validación

Como podemos observar en la figura 2.3.5, luego de haber realizado una modificación y dado clic al botón de Evaluar aparece una notificación debido a que el número de faltas (3) que se están reportando en las asesorías es el máximo permitido.

Una vez realizada la evaluación se procede a dar clic al botón de Guardar.

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar | Evaluar | Guardar | Modificar | Refrescar | Borrar | Cancelar | Imprimir

**Datos personales**

Nombre: ERICK REYNALDO VALDIVIA ORTEGA No. Cta. 098026531 Promedio: 9,15

Departamento: DID Semestre que concluyo: 2005-1 Periodo: 2004/8/15 a 2004/1/6 Evaluación: 05

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 08/01/2005 Eval.Especial

**Evaluación**

*Rendimiento Académico*

Materias inscritas: Terminó Créditos  
 Materias aprobadas: Terminó Créditos  
 Materias reprobadas: 0  
 Estatus: Inicio Tesis

*Cursos*

Intersemestrales: 0  
 Sábados y Domingos: 0  
 Internos: 0  
 Especiales: 0  
 Semestrales: 0  
 Cursos Asistidos: 0  
 Prebecarios: 1

*Proyectos*

Lider de Proyectos: 0  
 Proyectos: 2  
 Oportunidad de Avance: Oportuno  
 Resultados: Oportunos

*Asesorías*

Asistencia: 90  
 Retardos: 1  
 Faltas: 3

*Asistencia y Puntualidad*

Asistencia: 75  
 Retardos: 1  
 Faltas: 5

Nota: los días hábiles para este periodo fueron 80

*Iniciativa*

Disposición para colaborar: Siempre  
 Realización de propuestas: Siempre  
 Compromiso con UNICA: Siempre

*Calidad en el Trabajo*

Responsabilidad: Siempre  
 Confiabilidad: Siempre  
 Actitud de Servicio: Siempre

*Elaboración de Manuales*: 3  
*Nuevas Propuestas*: 1

*Notificaciones*

El becario esta a consideración por el número de faltas en asesoría

*Comentarios*

Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad.

Primero | Anterior | Siguiente | Ultimo

Figura 3.2.6 Guardando las modificaciones

**Sistema de Control de las Evaluaciones de Becarios**

Salir

Regresar | Evaluar | Guardar | Modificar | Refrescar | Borrar | Cancelar | Imprimir

**Datos personales**

Nombre: ERICK REYNALDO VALDIVIA ORTEGA No. Cta. 098026531 Promedio: 9,15

Departamento: DID Semestre que concluyo: 2005-1 Periodo: 2004/8/15 a 2004/1/6 Evaluación: 05

Etapa Actual: 1er ingreso Etapa Propuesta: etapa 1 Generación: 48 Fecha de Ingreso: 08/01/2005 Eval.Especial

**Evaluación**

*Rendimiento Académico*

Materias inscritas: Terminó Créditos  
 Materias aprobadas: Terminó Créditos  
 Materias reprobadas: 0  
 Estatus: Inicio Tesis

*Cursos*

Intersemestrales: 0  
 Sábados y Domingos: 0  
 Internos: 0  
 Especiales: 0  
 Semestrales: 0  
 Cursos Asistidos: 0  
 Prebecarios: 1

*Proyectos*

Lider de Proyectos: 0  
 Proyectos: 2  
 Oportunidad de Avance: Oportuno  
 Resultados: Oportunos

*Asesorías*

Asistencia: 90  
 Retardos: 1  
 Faltas: 3

*Asistencia y Puntualidad*

Asistencia: 75  
 Retardos: 1  
 Faltas: 5

Nota: los días hábiles para este periodo fueron 80

*Iniciativa*

Disposición para colaborar: Siempre  
 Realización de propuestas: Siempre  
 Compromiso con UNICA: Siempre

*Calidad en el Trabajo*

Responsabilidad: Siempre  
 Confiabilidad: Siempre  
 Actitud de Servicio: Siempre

*Elaboración de Manuales*: 3  
*Nuevas Propuestas*: 1

*Notificaciones*

El becario esta a consideración por el número de faltas en asesoría

*Comentarios*

Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad.

Primero | Anterior | Siguiente | Ultimo

**SICEB**

Para ver los cambios de click a Refrescar

Aceptar

Figura 3.2.7 Guardando las modificaciones

Al guardar las modificaciones, aparece una pantalla que nos indica que para verlas, se debe de dar clic al botón de Refrescar:



Figura 3.2.8 Refrescar los cambios.

Una vez refrescada la consulta la pantalla vuelve al aspecto que tenía al momento de visualizar una evaluación.

The screenshot shows the 'Sistema de Control de las Evaluaciones de Becarios' application. At the top, there is a menu bar with options: Salir, Regresar, Evaluar, Guardar, Modificar, Refrescar, Borrar, Cancelar, and Imprimir. Below the menu bar, the 'Datos personales' section displays information for ERICK REYNALDO VALDIVIA ORTEGA, including his ID number (098026531), average score (9.15), department (DID), semester (2005-1), period (2004/8/15 to 2004/1/6), evaluation score (05), current stage (1er ingreso), proposed stage (etapa 1), generation (48), and date of entry (08/01/2005). The 'Evaluación' section is divided into several categories: 'Rendimiento Académico' (Materias inscritas, aprobadas, reprobadas, Estatus), 'Asistencia y Puntualidad' (Asistencia: 75, Retardos: 1, Faltas: 5), 'Cursos' (Intersemestrales, Sábados y Domingos, Internos, Especiales, Semestrales, Cursos Asistidos, Prebecarios, Elaboración de Manuales, Nuevas Propuestas), 'Proyectos' (Lider de Proyectos, Proyectos, Oportunidad de Avance, Resultados), 'Asesorías' (Asistencia: 90, Retardos: 1, Faltas: 3), 'Notificaciones' (El becario esta a consideración por el número de faltas en asesorías), and 'Comentarios' (Erick, acaba de ingresar a la UNICA, actualmente esta en el proyecto de evaluación de becarios y creando los inventarios de la uidad). The 'Faltas' value of 3 in the 'Asesorías' section is circled in red, with a red arrow pointing to it and the text 'El número de faltas se guardó'. Another red arrow points to the notification message, with the text 'La notificación queda guardada'. At the bottom, there are navigation buttons: Primero, Anterior, Siguiete, and Ultimo.

Figura 3.2.9 Visualizando los cambios

Dentro del Sistema se pueden visualizar tanto las evaluaciones nuevas, como las anteriores, sin embargo aquellas diferentes al periodo que se esta evaluando no se pueden Modificar, debido a que de lo contrario, se cambiaría la base de datos, ya que de las evaluaciones se genera un reporte impreso; si se modificaran las evaluaciones pasadas no coincidirían los datos que se encuentran en la base con los reportes generados.

Si se trata de hacer lo anterior saldrá la siguiente pantalla:

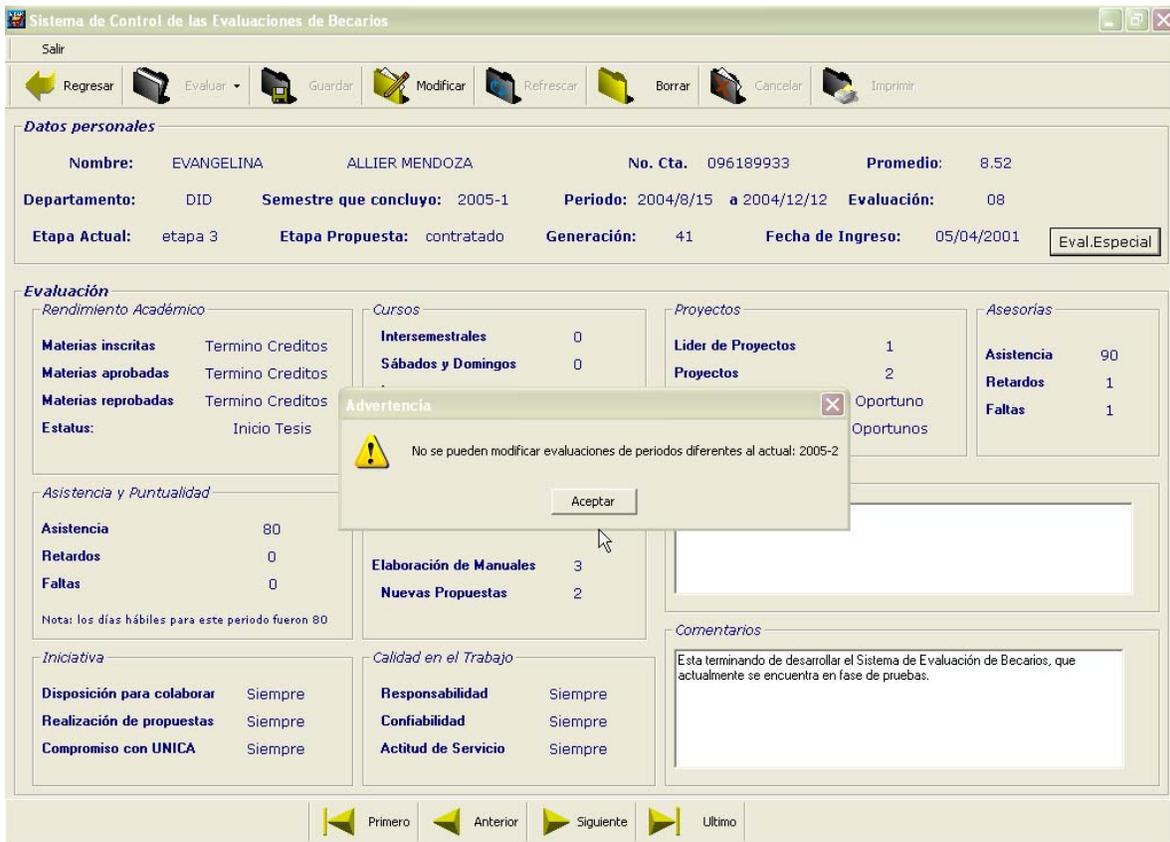
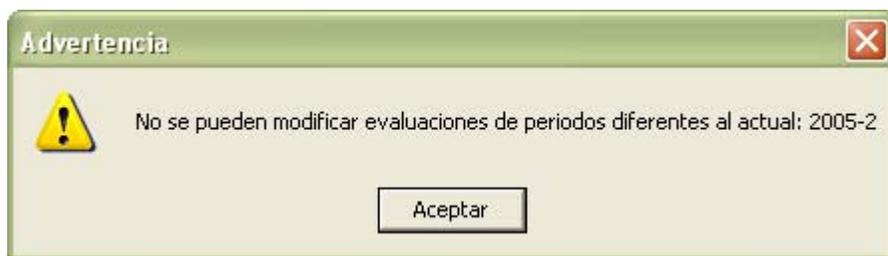


Figura 3.2.10 No se puede modificar



*Figura 3.2.11 Vista de la pantalla de advertencia.*

De esta manera aseguramos que lo que se vaya almacenando en la base sea coherente con lo que se tenga en los reportes, de la misma manera si se trata de modificar una evaluación especial de periodo diferente, no será permitido.

### ***3.2.1.- Modificando una Evaluación Especial***

Para aquellos departamentos, que cuentan con evaluaciones especiales, el procedimiento para modificarlos es el mismo.

Se da clic al botón de Evaluación Especial; tomemos el caso del DROS cuando visualizamos datos en ella su aspecto es el de la figura siguiente:

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

**Número de atención de solicitud de eventos de soporte técnico atendidos**

Número de Atendidos	23
Número de Atendidos Parcialmente	2

**Realización de Respaldos de Servidores**

Número de respaldos	1
Cantidad de información en MB respaldada	6

**Número de sistemas para la administración al que le da mantenimiento**

Cantidad	4
Actualización del sistema	3

**Número de Aportaciones al sistema de conocimientos de Administración en Cómputo**

Cantidad	1
----------	---

**Número de atención a incidentes de operación de servidores y redes fuera del horario**

Cantidad de Eventos	3
Número de horas invertidas	4

*Figura 3.2.1.1 Pantalla de evaluación Especial del DROS.*

De la misma manera que para la evaluación general, se da clic al botón de modificar y los datos pasarán a cajas de texto para que puedan ser modificados.

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos	23
Número de Atendidos Parcialmente	2

Realización de Respaldos de Servidores

Número de respaldos	1
Cantidad de información en MB respaldada	6

Número de sistemas para la administración al que le da mantenimiento

Cantidad	4
Actualización del sistema	3

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad	1
----------	---

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos	3
Número de horas invertidas	4

*Figura 3.2.1.2 Modificando la evaluación especial*

Una vez hechas las modificaciones, se da clic en guardar y posteriormente en refrescar para ver los cambios realizados.

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos	
Número de Atendidos	100
Número de Atendidos Parcialmente	2

Realización de Respaldos de Servidores	
Número de respaldos	1
Cantidad de información en MB respaldada	20

Número de sistemas para la administración al que le da mantenimiento	
Cantidad	4
Actualización del sistema	3

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo	
Cantidad	1

Número de atención a incidentes de operación de servidores y redes fuera del horario	
Cantidad de Eventos	3
Número de horas invertidas	4

**Se cambiaron estos dos datos**

Figura 3.2.1.3 Datos que se cambian

Evaluación Especial del DROS

Regresar    Modificar    Guardar    Cancelar    Refrescar

Evaluación DROS

Número de atención de solicitud de eventos de soporte técnico atendidos

Número de Atendidos	100
Número de Atendidos Parcialmente	2

Realización de Respaldos de Servidores

Número de respaldos	1
Cantidad de informa	

Número de sistemas para la

Cantidad	
Actualización del sist	3

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo

Cantidad	1
----------	---

Número de atención a incidentes de operación de servidores y redes fuera del horario

Cantidad de Eventos	3
Número de horas invertidas	4

SICEB

Se guardó la modificación, refresque la consulta para ver los cambios

Aceptar

Figura 3.2.1.5 Aviso de que se realizaron los cambios

**Evaluación Especial del DROS**

Regresar    Modificar    Guardar    Cancelar    Refrescar

**Evaluación DROS**

Número de atención de solicitud de eventos de soporte técnico atendidos	
Número de Atendidos	100
Número de Atendidos Parcialmente	2

Realización de Respaldos de Servidores	
Número de respaldos	1
Cantidad de información en MB respaldada	20

Número de sistemas para la administración al que le da mantenimiento	
Cantidad	4
Actualización del sistema	3

Número de Aportaciones al sistema de conocimientos de Administración en Cómputo	
Cantidad	1

Número de atención a incidentes de operación de servidores y redes fuera del horario	
Cantidad de Eventos	3
Número de horas invertidas	4

**Se guardaron los cambios**

Figura 3.2.1.4 Visualizando los cambios efectuados.

## 4.- Generando una nueva evaluación

En la pantalla principal del SICEB tenemos en la barra de herramientas el botón de Nuevo el cual nos permite ver los becarios que faltan de ser evaluados.

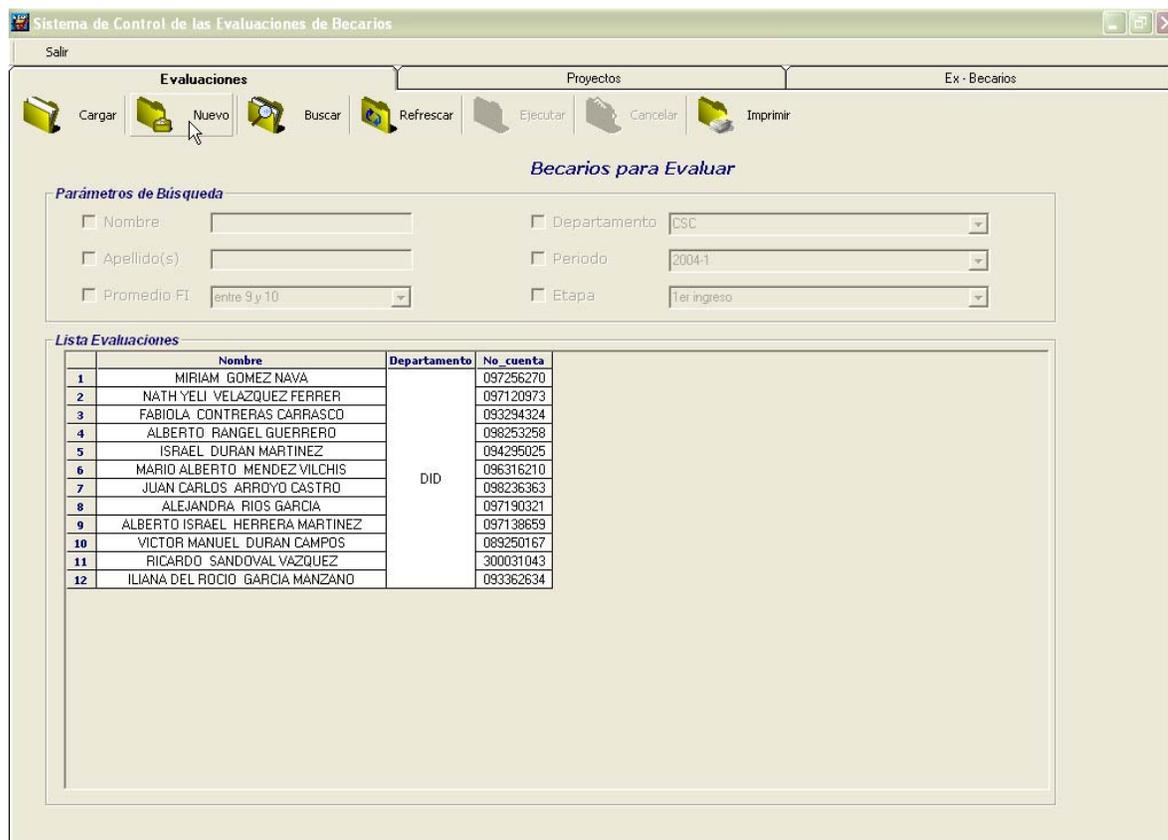


Figura 4.1 Becarios que faltan de evaluación

De esta manera tenemos una vista rápida de los becarios; para poder realizar la evaluación basta dar doble clic sobre el nombre.

Figura 4.2 Nueva evaluación

Si nos damos cuenta, esta pantalla es la misma que se utiliza para visualizar las evaluaciones ya existentes sólo que ahora aparecen las cajas de texto necesarias, así como las cajas de opción múltiple para poder evaluar al los becarios. Los puntos que se evalúan son:

#### 4.1 Datos personales:

**Promedio:** Se introduce el promedio del becario dentro de la Facultad por ejemplo: 8.55

**Evaluación:** Calificación asignada por cada jefe de departamento al becario según su desempeño. Esta va desde 00 a 10.

*Etapa Actual:* Etapa en la que se encuentra el becario al momento de evaluarlo.

*Etapa Propuesta:* Etapa a la que se propone que pase el becario en el semestre siguiente, la etapa propuesta también puede ser una etapa inferior a la actual.

*Generación:* Se especifica el número de la generación en la que el becario entró al programa. Este dato sólo es requerido la primera vez que se evalúa al becario, las siguientes veces ya no se pedirá y se visualizará al momento de cargar las evaluaciones.

*Fecha de Ingreso:* Al igual que el dato anterior, sólo será requerido la primera vez que se evalúe a un becario, y consiste en el año/mes/día en que el becario inició sus labores dentro de UNICA (no se cuenta el periodo de prebecario).

#### 4.2 Evaluación:

*Rendimiento Académico:* En este aspecto se lleva un control de las materias inscritas, aprobadas y reprobadas, el número de materias inscritas puede ser desde 1 a 7, teniendo la opción a elegir si el becario ya terminó los créditos, en el momento que se elija en el combo de materias inscritas una opción diferente a la de terminó créditos, el combo de materias aprobadas se habilitará, para elegir las materias que aprobó el becario, mientras que el combo Estatus queda deshabilitado marcando la opción de que el becario lleva materias, si se elige en el combo de materias inscritas la opción de Terminó Créditos, el combo de materias aprobadas, queda inhabilitado y se habilita el de estatus, para poder elegir entre: Inicio Tesis, No inicio Tesis, Terminó Tesis y Titulado.

Rendimiento Académico

**Materias inscritas** Termino Creditos

**Materias aprobadas** Termino Créditos

**Materias reprobadas**

**Estatus:** Inicio Tesis

Figura 4.2.1 Rendimientos Académico

El número de materias reprobadas, se calcula automáticamente al terminara la evaluación y dar clic al botón de evaluar.

Sólo se permite que el becario repruebe una materia al semestre, en caso de que sean más, se considerará como un caso especial, y se generará una notificación en la evaluación del becario.

*Asistencia y Puntualidad:* Aquí se colocan las asistencias, faltas y retardos que el becario haya tenido en sus horas de trabajo en UNICA. En la parte inferior de esta sección se encuentra una nota, que nos indica cuantos días hábiles existieron en el periodo, recordemos que los días laborables en UNICA no son los mismos que los del semestre. Este dato es introducido por el administrador del sistema antes de iniciar las evaluaciones.

Con respecto a las asistencias, el jefe de cada departamento colocará este número, los retardos contarán al momento de que se busque al becario en sus horas de trabajo y este no se encuentre, las faltas se calculan de la misma manera que las materias reprobadas, de manera automática al dar clic al botón de Evaluar las faltas serán tomadas como los días hábiles menos las asistencias.

Asistencia y Puntualidad

Asistencia

Retardos

Faltas

Nota: los días hábiles para este periodo fueron 80

Días hábiles del periodo

Figura 4.2.2 Asistencia y Puntualidad

*Iniciativa:* En este aspecto se consideran tres puntos a evaluar: disposición para colaborar, realización de propuestas, compromiso con UNICA, cada uno de ellos con cuatro puntos para poder elegir:

- Siempre
- Muchas veces
- Algunas veces
- Pocas veces

Iniciativa

Disposición para colaborar

Realización de propuestas

Compromiso con UNICA

Figura 4.2.3 Iniciativa

*Cursos:* En esta sección se visualizan el número de cursos que el becario haya dado durante el semestre, agrupándolos por:

- *Semestrales:* Aquellos que se dan durante el curso del semestre
- *Sábados y Domingos:* Aquellos que se dan durante el semestre pero únicamente los fines de semana.

- *Internos:* Cursos que son sólo para los mismo becarios y personal de UNICA, no están abiertos al público en general.
- *Intersemestrales:* Aquellos que son impartidos en el periodo entre un semestre y otros (vacaciones ).
- *Especiales:* Cursos a empresas u otros organismos que son pedidos por ellos o bien alguna división de la Facultad.

Los datos anteriores, por encontrarse en otro sistema no son introducidos por el usuario y por lo tanto tampoco pueden ser modificados.

En la parte de cursos asistidos se hace un resumen de los cursos que el becario haya tomado ya que esa es una parte fundamental de su desarrollo.

Los cursos para prebecarios, así como la elaboración de manuales y nuevas propuestas deben ser introducidos por el usuario:

- *Prebecarios:* Se refiere a los cursos de preparación para los nuevos becarios que haya dado.
- *Elaboración de manuales:* Cuantos manuales (de algún sistema o bien de programación) elaboró el becario durante el periodo que se esta evaluando.
- *Nuevas Propuestas:* Es un dato numérico, que representa cuantas propuestas de cursos nuevos dio el becario a lo largo del semestre, esto con la finalidad de ampliar la gama de temas que se abordan en los cursos.

Cursos	
<b>Intersemestrales</b>	1
<b>Sábados y Domingos</b>	0
<b>Internos</b>	0
<b>Especiales</b>	0
<b>Semestrales</b>	0
<b>Cursos Asistidos</b>	0
<b>Prebecarios</b>	<input type="text"/>
<b>Elaboración de Manuales</b>	<input type="text"/>
<b>Nuevas Propuestas</b>	<input type="text"/>

Figura 4.2.4 Cursos

*Calidad en el Trabajo:* Se evalúa el trabajo que el becario haya hecho, tomando como puntos bases, la responsabilidad que mostró para realizar sus actividades, la confiabilidad y la actitud de servicio al igual que en la “iniciativa” cada opción cuenta con cuatro rangos para evaluar:

- Siempre
- Muchas veces
- Algunas veces
- Pocas veces

Calidad en el Trabajo	
<b>Responsabilidad</b>	<input type="text" value="Siempre"/>
<b>Confiabilidad</b>	<input type="text" value="Siempre"/>
<b>Actitud de Servicio</b>	<input type="text" value="Siempre"/>

Figura 4.2.5 Calidad en el trabajo

*Proyectos:* Aquí se llevará un record de los proyectos en los que participa el becario y su desempeño dentro de ellos en general, básicamente lo que se desea saber es: de cuantos proyectos es líder el becario, en cuantos participa, cual ha sido el avance en general y los resultados. Para los dos últimos puntos oportunidad y resultados, se tienen cuatro opciones:

- Oportuno
- Regular
- Mínimo
- Ninguno



The screenshot shows a form titled "Proyectos" with the following fields:

- Lider de Proyectos:** A text input field.
- Proyectos:** A text input field.
- Oportunidad de Avance:** A dropdown menu with "Oportuno" selected.
- Resultados:** A dropdown menu with "Oportunos" selected.

Figura 4.2.6 Proyectos

*Asesorías:* Se lleva un control del desempeño del becario en las asesorías que da. Se evalúan tres aspectos: Asistencia, Faltas y Retardos, en cuanto a las faltas, si estas son mayores o iguales a tres, se generará una notificación.



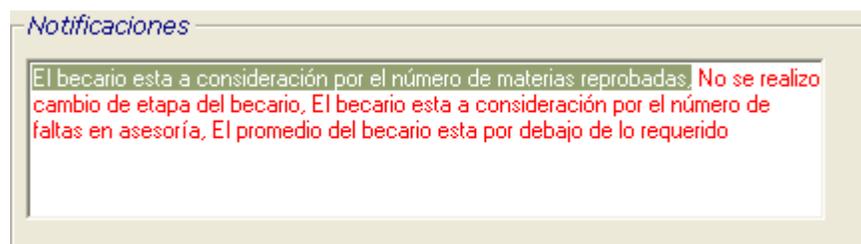
The screenshot shows a form titled "Asesorías" with the following fields:

- Asistencia:** A text input field.
- Retardos:** A text input field.
- Faltas:** A text input field.

Figura 4.2.7 Asesorías

*Notificaciones:* Se mostraran notas cuando algún becario no cumpla con los requisitos especificados dentro del reglamento. Las notificaciones pueden ser:

- Promedio: Si el promedio en la Facultad es menor a 8 se genera la notificación : “El promedio del becario esta por debajo de lo requerido”.
- Etapa: Si por su desempeño el becario no es acreedor a pasar a una etapa mayor se genera la notificación: “No se realizo cambio de etapa del becario”.
- Faltas a asesoría: Se genera la notificación: “El becario esta a consideración por el número de faltas en asesoría”.
- Materias Reprobadas: Un becario no puede reprobado al semestre más de dos materias, por lo tanto cuando esto ocurre se genera la notificación de: “1 becario esta a consideración por el número de materias reprobadas”.



*Figura 4.2.8 Notificaciones*

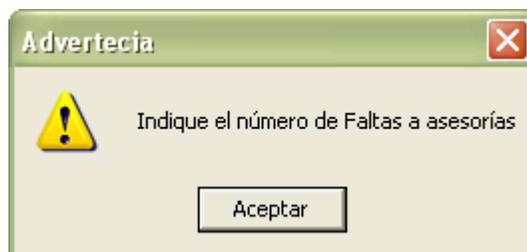
*Comentarios:* Este espacio esta reservado para realizar tanto comentarios negativos como positivos del becario o para justificar alguna de las notificaciones, por ejemplo que haya faltado a asesorías por motivos personales, por enfermedad o bien por estar dando algún curso.



*Figura 4.2.9 Comentarios*

Una vez llenados todos los campos anteriores se da clic en el botón de Evaluar de la barra de herramientas esto nos sirve para generar los datos automáticos del sistema como lo son: materias reprobadas, faltas a UNICA y las notificaciones que puedan surgir.

En dado caso de que faltase algún campo de llenar al dar clic a Evaluar se generará una notificación:



*Figura 4.2.10 Falta indicar las Faltas en Asesorías*

La evaluación terminada luego de dar clic a Evaluar tiene el aspecto que nos muestra la figura 4.1.3:

Figura 4.2.11 Evaluación realizada

Luego de Evaluar, se da clic a Guardar para registrar los datos en la base, e ir formando el historial de evaluaciones del becario.

Figura 4.2.12 Guardar evaluación

Una vez guardada la evaluación, podemos seguir evaluando a los demás becarios que faltan simplemente utilizando la barra de navegación que está en la parte inferior al desplazarnos, las cajas de texto se limpian y todo queda listo para evaluar al siguiente becaño como lo podemos apreciar en la figura 4.15

Figura 4.2.13 Siguiente evaluación

## 5. Borrar Evaluación

Dentro de la pantalla de Evaluación, cuando se esta visualizando o bien cuando acabamos de hacer una nueva evaluación existe la opción de borrarla, al hacerlo este se elimina de la base de datos, sin embargo los datos personales no son borrados pues estos son tomados del SICC, los datos que se eliminan son los de la evaluación.

En el momento de dar clic a borrar, aparece una pantalla de confirmación si se elige Si el registro será borrado, de lo contrario se conservará.

Figura 5.1 Confirmación de borrado de registro

Si se borró el registro, aparecerá la pantalla que muestra la figura 5.2 donde se pide que se refresque la pantalla para ver los cambios.

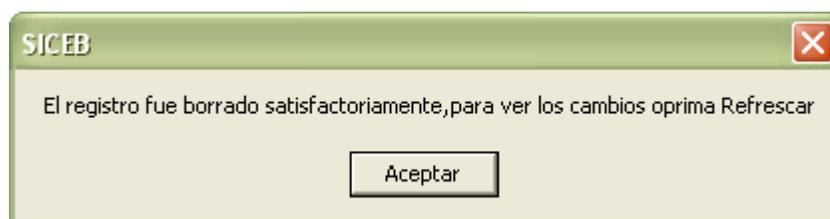


Figura 5.2 Registro Borrado

Si se diera el caso que se borran todos los registros uno por uno, en el momento de refrescar la consulta, aparecerá un aviso que nos dice que no existen registros en la base de datos, y automáticamente nos regresará a la pantalla principal.

## 6. Opciones del Administrador.

El administrador del SICEB es el Departamento de Servicios Académicos, él puede visualizar las evaluaciones de los demás departamentos y modificarlos si así se requiera más no puede evaluar a los becarios que no pertenezcan al DSA.

Por esta razón cuenta en la barra de herramientas de la pantalla principal con una opción para cargar las evaluaciones como se muestra en la figura 6.



Figura 6.1 Cargar todas las evaluaciones.

El resultado de esta acción nos mostrará en pantalla lo siguiente:

**Evaluaciones de todos los departamentos**

**Parámetros de Búsqueda**

Nombre

Departamento

Apellido(s)

Periodo

Promedio FI

Etapa

**Lista de Evaluaciones**

	Nombre	Departamento	No. cuenta	Periodo	Calificación	Etapas Actual	Promedio FI	Mat.Inscritas	Mat.Aprobadas
1	ANDRES SANCHEZ DELGADO	DSA	096290112	2005-1	08	etapa 2	8.52	0	0
2	EVANGELINA ALLIER MENDOZA	DSA	096189933		08	etapa 3	8.52	0	0
3	ALEJANDRA RIOS GARCIA	DID	097190321		01	etapa 1	8.00	5	1
4	PATRICIA ANEZI ALTAMIRANO DIAZ	DRDS	097125514		10	etapa 2	9.85	4	3
5	JAVIER HERNANDEZ BARRAZA	DRDS	098257737		08	Ter ingreso	7.41	5	4
6	JUAN CARLOS HERNANDEZ VARGAS	DSC	098234068		10	Ter ingreso	7.41	0	0

**Se muestran las evaluaciones de todos los departamentos**

Figura 6.2 Todas las evaluaciones

Mientras que si sólo cargamos sin la opción de todos, nos mostrará la siguiente pantalla:

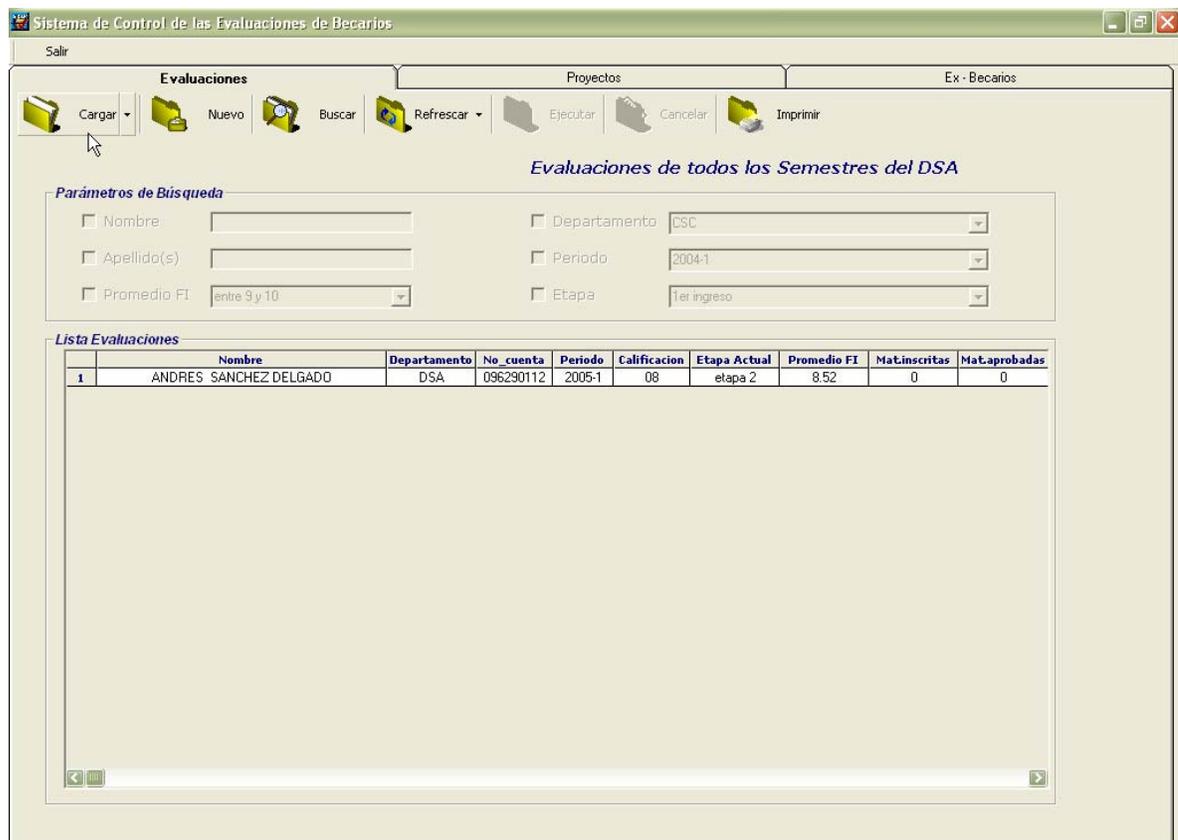


Figura 6.3 Cargar sólo las evaluaciones del DSA

El DSA, es el único autorizado para realizar las búsquedas por departamento, pudiendo combinar la búsqueda en un departamento con otros parámetros por ejemplo: Departamento: DID, Nombre: Evangelina.

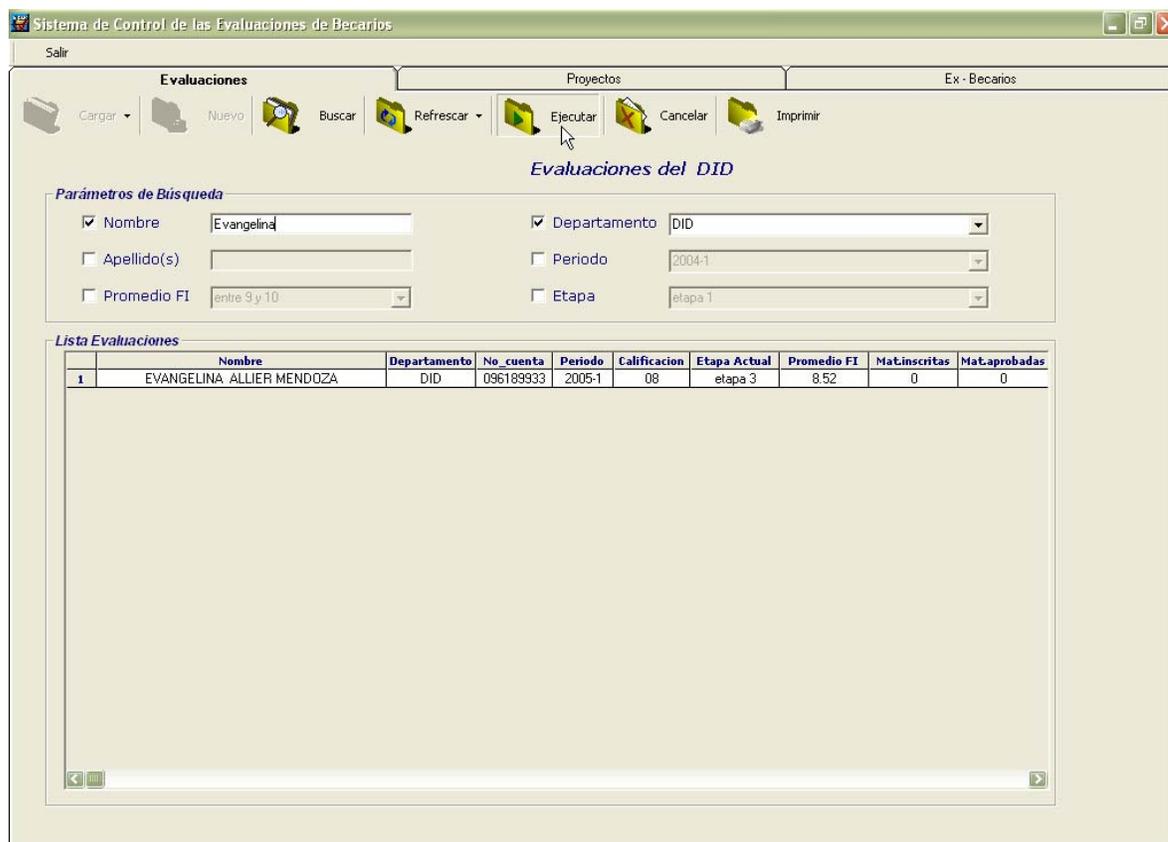


Figura 6.4 Búsqueda por parámetros.

Otra opción del DSA es que cuando se va a evaluar un periodo nuevo debe de introducir los siguientes datos:

- Día de inicio del periodo
- Mes de inicio del periodo
- Día de fin del periodo
- Mes fin del periodo
- Días hábiles del periodo

Los días hábiles son requeridos por el sistema ya que con base en ellos es que se sacan las faltas que el becario haya tenido a sus horas en UNICA y los demás campos nos sirve para saber las fechas que comprendieron el periodo que se evalúa, la pantalla de petición de estos datos es la que muestra la figura 6.5

Periodo

Por favor indique las fechas de inicio y término del semestre 2005-1

Inicio

Día: 15

Mes: 08

Fin

Día:

Mes:

Por favor indique cuantos días hábiles tuvo este periodo

Días Hábiles

Aceptar

Figura 6.5 Datos del periodo

Estos datos son fundamentales para poder iniciar la evaluación de los becarios por los demás departamentos, por lo tanto si el DSA no los introduce los demás no podrán hacer la evaluación y en el momento de entrar al sistema aparecerá el siguiente aviso:

SICEB

No se puede evaluar a los becarios, pues el administrador no ha dado de alta los periodos

Aceptar

Figura 6.6 No se pueden evaluar

Sin embargo si se pueden visualizar las demás evaluaciones en caso de requerirse, la pantalla principal aparecerá con el botón Nuevo de la barra de herramientas desactivado.

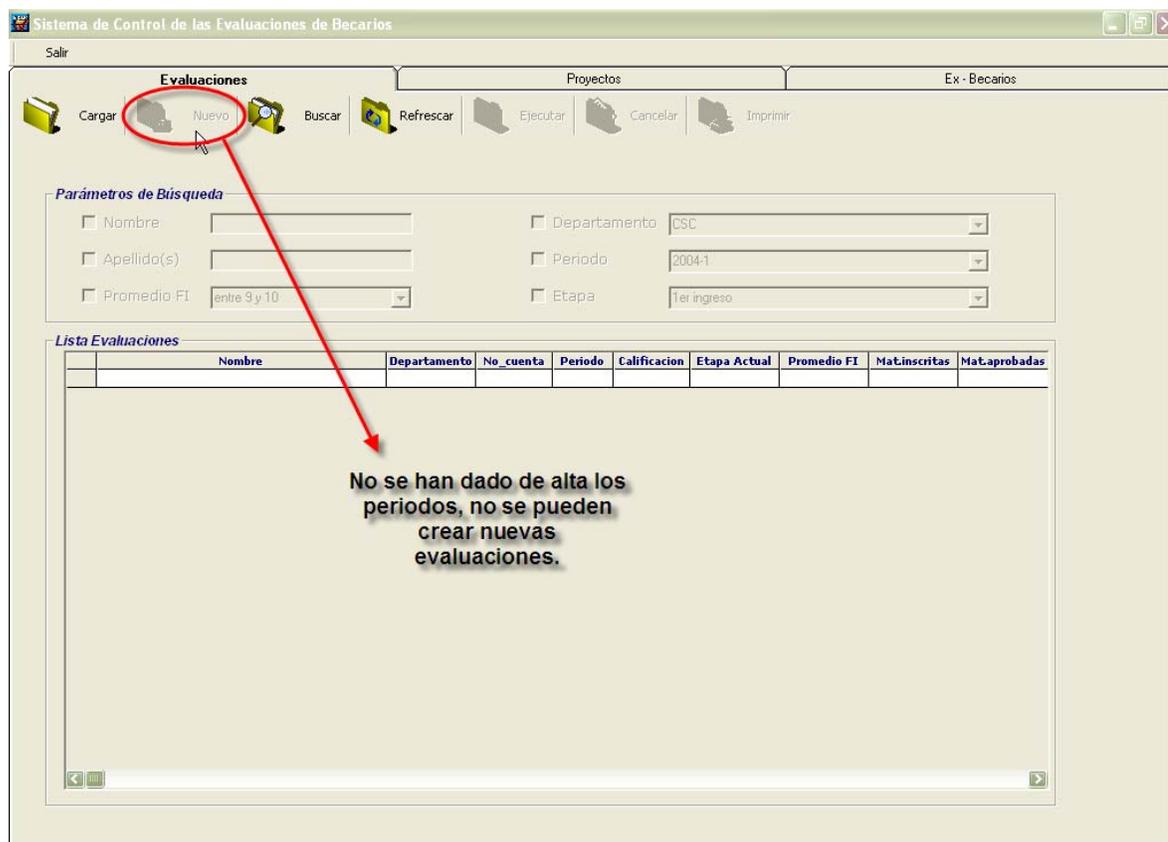


Figura 6.7 Puedo visualizar, más no crear nuevas evaluaciones

## 7.- Impresión de Reportes.

Una de las finalidades del sistema es la creación de reportes, esta opción actualmente esta permitida únicamente para el administrador del sistema; tenemos dos tipos de reportes, uno general donde salen todos los becarios con los datos más relevantes y uno individual que muestra todos los datos de la evaluación.

La impresión de reportes estará permitida siempre y cuando existan registros en la base de datos.

Para poder entrar a ellos basta dar clic al botón imprimir en la pantalla principal y seleccionar la opción que se desea:

- a) Reporte General
- b) Reporte Individual

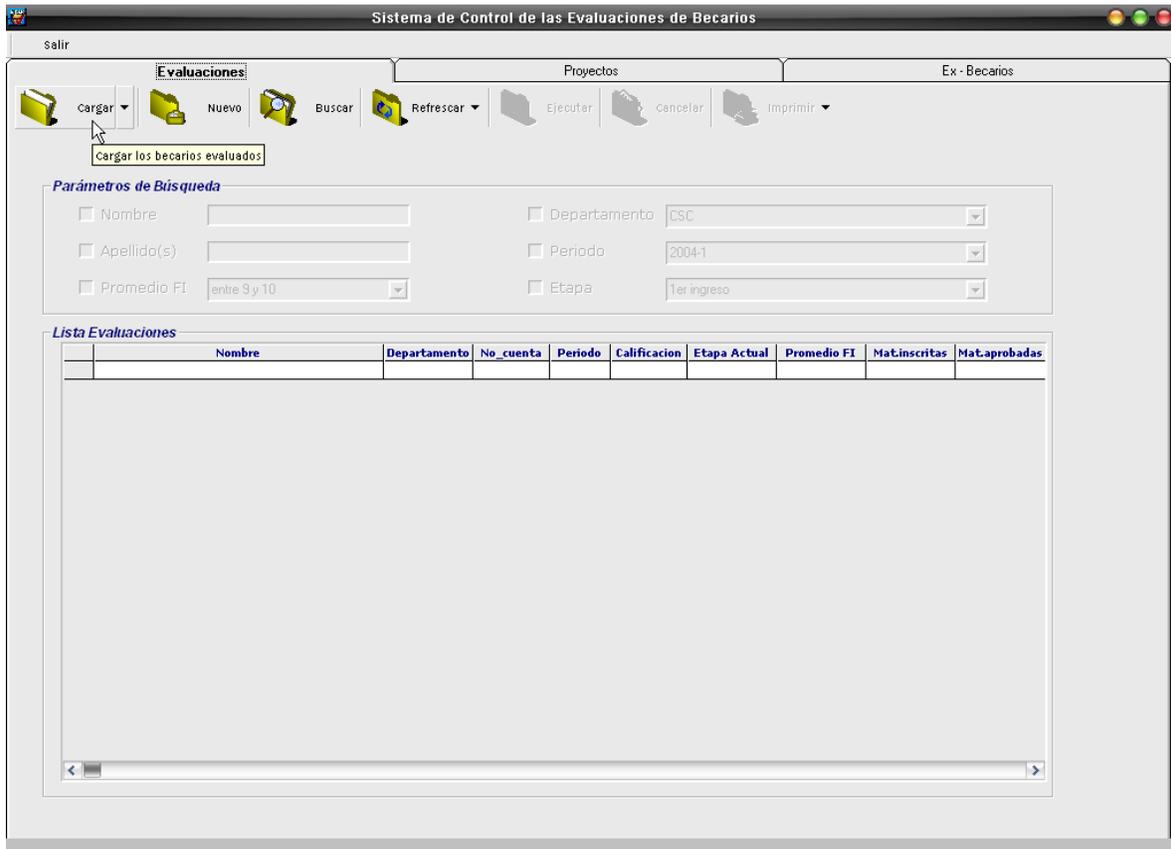


Figura 7.1 Pantalla principal, donde vemos que el mouse esta colocado en el botón de cargar, mientras que el de imprimir permanece deshabilitado.

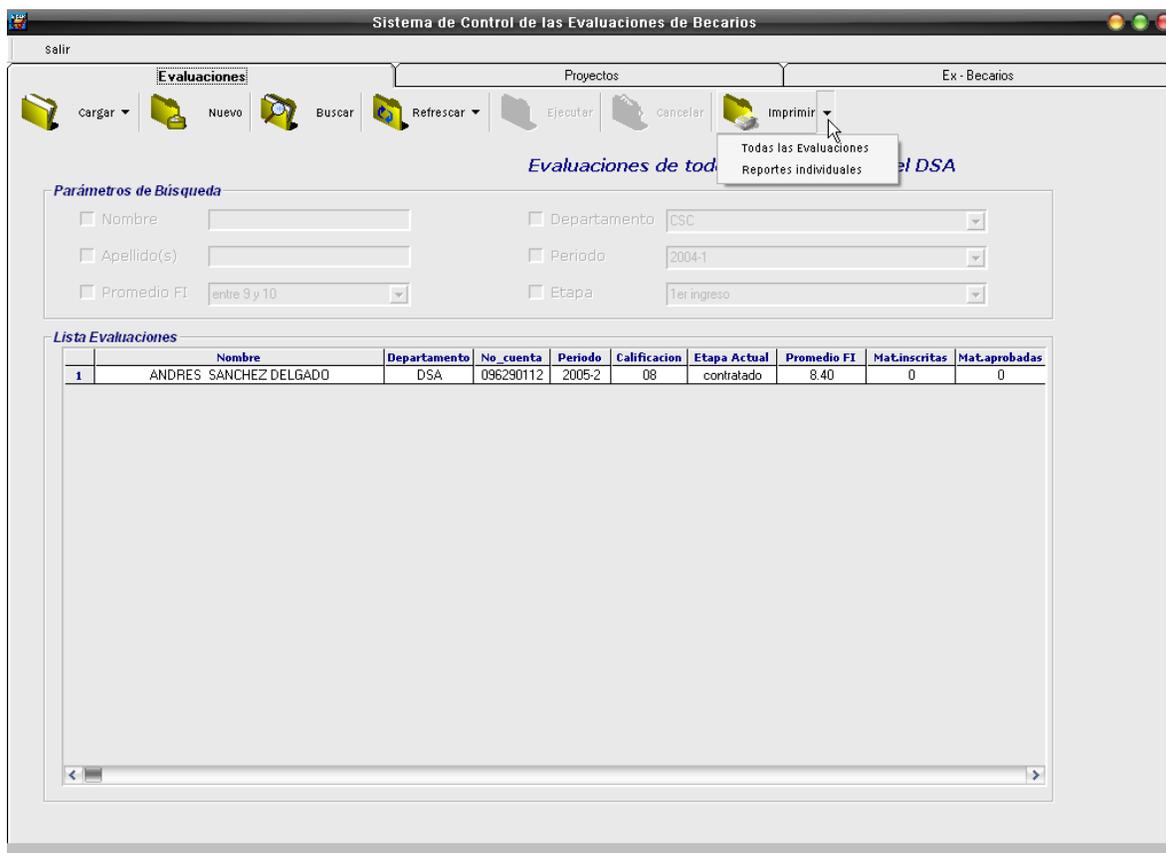


Figura 7.2 Al cargar datos, el botón de imprimir se habilita, y podemos escoger alguna de las dos opciones.

Reportes

Vista previa

75%

1 de 1

UNICA

FACULTAD DE INGENIERÍA  
UNIDAD DE SERVICIOS DE  
CÓMPUTO ACADÉMICO  
DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO  
SISTEMA DE EVALUACIÓN DE BECARIOS

**Reporte de las Evaluaciones de los Becarios**

Nombre	Departamento	Periodo	Promedio	Etapas Actual	Etapas Propuesta	Estatus	Cursos Impartidos	Faltas Asesoría
ANDRES SANCHEZ DELGADO	DSA	2005-2	8.40	contratado	contratado	Inicio Tesis	3	3
DULCE CAMPOS DEL RAZO	DROS	2005-2	8.10	1er ingreso	etapa 1	8 Semestre	2	3
ELIANA DEL ROCIO GARCIA MAI	DID	2005-2	9.00	contratado	contratado	Termino Tesis	2	3

UNICA/DID/SICEB/@rahuciones 1 04/07/2005

Figura 7.3 Se eligió la opción de Reporte General, esta es la vista de la pantalla.



FACULTAD DE INGENIERÍA.  
UNIDAD DE SERVICIOS DE  
CÓMPUTO ACADÉMICO  
DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO.  
SISTEMA DE EVALUACIÓN DE BECARIOS

### Reporte de las Evaluaciones de los Becarios

Nombre	Departamento	Periodo	Promedio	Etapa Actual	Etapa Propuesta	Estatus	Cursos Impartidos	Faltas Asesoría
ANDRES SANCHEZ DEL GADO	DSA	2005-2	8.40	contratado	contratado	Inicio Tesis	3	3
DULCE CAMPOS DEL RAZO	DROS	2005-2	8.10	1er ingreso	etapa 1	8 Semestre	2	3
ILLIANA DEL ROCIO GARCIA MAI	DID	2005-2	9.00	contratado	contratado	Termino Tesis	2	3

UNICA/DID/SICEB/Evaluaciones 1 04/07/2005

Figura 7.4 Vista del Reporte General una vez impreso.

Evaluaciones Individuales

Facultad de Ingeniería  
Secretaría General  
Unidad de Servicios de Cómputo Académico

**Evaluación de Becarios**

Nombre del Becario: ANDRES SANCHEZ DELGADO		No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005-2	Estatus: Inicio Tesis	
Etapa actual: contratado	Etapa propuesta: contratado	Evaluación: 08	

Característica	Desarrollo	Cantidad
Rendimiento Académico	Materias Inscritas	0
	Materias Arpobadas	0
	Materias Reprobadas	0
Asistencia y Puntualidad en UNICA	Asistencias	100
	Retardos	1
	Faltas	14
Iniciativa	Disposición para colaborar	Muchas veces
	Realización de propuestas	Algunas veces
	Compromiso con UNICA	Algunas veces
Cursos	Intersemestrales	1
	Internos	0
	Especiales	0
	Prebecarios	1
	Sábados y Domingos	1
	Semestrales	0

Figura 7.4 Vista de la pantalla cuando se elige imprimir el Reporte Individual

<b>Evaluación de Becarios</b>			
Nombre del Becario: ANDRÉS SANCHEZ DELGADO		No. de cuenta: 096290112	Promedio Actual: 8.40
Departamento: DSA	Semestre que concluyó: 2005.2	Estatus: Inicio Tesis	
Etapas actual: cotizado	Etapas propuestas: cotizado	Evaluación: 08	
Categorías	Detalle	Cantidad	
Rendimiento Académico	Materias Insatisfechas	0	
	Materias Aprobadas	0	
	Materias Reprobadas	0	
Asistencia y Puntualidad en UNICA	Asistencias	100	
	Retardos	1	
	Faltas	14	
Iniciativa	Disposición para colaborar	Muchas veces	
	Realización de propuestas	Algunas veces	
	Compromiso con UNICA	Algunas veces	
Cursos	Intereses en cursos	1	
	Intemos	0	
	Especiales	0	
	Prácticas	1	
	Sábados y Domingos	1	
	Seminarios	0	
	Elaboración de Manuales	2	
	Cursos Asistidos	0	
	Nuevas propuestas	1	
	Calidad en el trabajo	Responsabilidad	Algunas veces
Confiable		Muchas veces	
Actitud de Servicio		Algunas veces	
Proyectos	Líder de Proyectos	0	
	Participación en Proyectos	0	
	Oportunidad de Avance	Mínimo	
	Resultados	Regular	
Asesorías	Asistencias	100	
	Retardos	2	
	Faltas	3	
Comentarios			
Notas	El becario está a consideración por el número de faltas en asesoría		
UNICA/DID/SICEB		12/08/2001	

Figura 7.5 Vista del Reporte Individual ya impreso.

---

# Bibliografía

JACOBSON, Ivar [et all]. 2000. El Proceso Unificado de Desarrollo de Software. Madrid. Addison Wesley. 464 p.

PRESSMAN, Roger S. 1997. Ingeniería de Software: un enfoque práctico. 4ª.edición. México, Mcgraw-Hill. 570 p.

RUMBAUGH, James [et all]. 1996. Modelado y Diseño Orientados a Objetos. España. Prentice – Hall. 635 p.

SCHMULLER, Joseph. 2000. Aprendiendo UML en 24 horas. México. Pearson Educacion. 448 p.

KENDALL & KENDALL. Análisis y Diseño de Sistemas 3ª Edición.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO. DIRECCIÓN DE SISTEMAS DPTO. DE INNOVACIÓN. 2004. Herramientas CASE para el Análisis y Diseño Estructurado. [en línea].

<<http://sistemas.dgsca.unam.mx/publica/pdf/casestru.pdf>>

QUALITRAIN EXPRESS. PROGRAMACIÓN ORIENTADA A OBJETOS EN VISUAL BASIC (partes 1 y 2). [en línea].

<<http://www.qualitrain.com.mx/objeIndirecto/programacion1.htm>>

<<http://www.qualitrain.com.mx/objeIndirecto/programacion2.htm>>

MONOGRAFIAS.HERRAMIENTAS CASE [en línea]

<<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml>>

---

TESIS: SISTEMA DE INSCRIPCIÓN Y CONTROL DE CURSOS PARA  
LA UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO (SICC).  
Flores Díaz José Armando. México 2004.