



# UNIVERSIDAD VERACRUZANA

---

---

Instituto de Ingeniería  
Universidad Veracruzana

## Instituto de Ingeniería

“Desarrollo de un sistema de control de  
estudiantes del Diplomado en Redes de  
Computadoras de la Facultad de  
Estadística e Informática por medio  
de un agente”

**T E S I S**

Que para obtener el grado de:

**Maestra en Ciencias de la Computación**

Presenta

**Ivonne Rabatte Suárez**

Boca del Río, Ver.

Mayo de 2002



## UNIVERSIDAD VERACRUZANA INSTITUTO DE INGENIERIA

H. Veracruz, Ver., a 19 de Abril del 2002  
DI073/02

Al Candidato al Grado  
Lic. Ivonne Rabatte Suárez  
Presente

En atención a su solicitud relativa, me es grato transcribir a Usted a continuación el tema que aprobado por esta Dirección propuso el M.C. Genaro Rebolledo Méndez, para que lo desarrolle como tesis, para obtener el Grado de Maestra en Ciencias de la Computación.

### TEMA

"DESARROLLO DE UN SISTEMA DE CONTROL DE ESTUDIANTES DEL DIPLOMADO EN REDES DE COMPUTADORAS DE LA FACULTAD DE ESTADÍSTICA E INFORMÁTICA POR MEDIO DE UN AGENTE"

- I .- Introducción
  - II .- Marco Teórico y Conceptual
  - III .- Marco de Referencia
  - IV .- Hipótesis
  - V .- Objetivos
  - VI .- Análisis
  - VII .- Diseño
  - VIII .- Desarrollo
  - IX .- Implantación
  - X .- Verificación y Validación
  - XI .- Pruebas y Evaluación del Sistema
  - XII .- Mantenimiento
  - XIII .- Conclusiones
  - XIV .- Recomendaciones y Sugerencias
- Bibliografía  
Anexos  
Glosario

Sin otro particular, me es grato reiterarle la seguridad de mi más atenta y distinguida consideración.

Atentamente  
"Lis de Veracruz: Arte, Ciencia, Luz"

Dr. Bonifacio C. A. Peña Pardo  
Director

BPP/apm\*.

Instituto de Ingeniería  
Universidad Veracruzana



DEDICATORIAS FAMILIARES

*Al hombre más valiente de la Tierra,  
mi padre, Leonel Rabatte Ramírez,  
a quien amo y admiro cada día más.*



*A mi amiga y consejera, mi madre,  
Lourdes Suárez de Rabatte, quien me  
enseñó a luchar con ahínco y perseverancia.  
Como siempre, te expreso mi amor infinito.*

*A mis hijos adorados, Ivonne, César y Leonel,  
que son la bendición más grande que me pudo  
enviar Dios. Ella, mi reina y mi mejor amiga;  
ellos, mi adoración, con la certeza que estamos  
formando a los mejores hombres. Los tres, en  
conjunto, lo más valioso de mi vida y lo que  
más amo.*

# Tesis de Maestría

*A mi esposo, Arq. César Fernández del Campo Chenge, quien me ha cuidado y hecho feliz desde el día que lo conocí. El camino ha sido más fácil con tu compañía. Te amo.*



*A mis hermanas, Lourdes Rabatte de Ornelas, Norma Rabatte de San Román, Maricela Rabatte de Hernández y Cora Rabatte de Minero, quienes compartimos el amor y admiración a nuestros padres. Con todo mi cariño.*

*A mis suegros, Dr. César Fernández del Campo Salado y Sra. Marina Chenge de Fernández del Campo, con cariño y con infinito agradecimiento por haber formado al mejor de los hombres.*

## AGRADECIMIENTOS

El presente trabajo recepcional, sin duda perfectible, fue realizado con la invaluable ayuda de prestigiados académicos de la Universidad Veracruzana. El tema fue propuesto y dirigido por el M.C. Genaro Rebolledo Méndez, quien en la actualidad realiza estudios de Doctorado en Inglaterra. La distancia no es obstáculo para expresarle mi agradecimiento por el empeño y dedicación que siempre demostró.

A mis profesores de la maestría, M.C. Abigail Zamora Hernández, Dr. Ismael Muñoz Maya, M.C. Evaristo Pacheco Velazco, Ing. Jaime Segura Barrada, M.C. Abelardo Rodríguez León, pilares fundamentales para poder llegar a esta etapa.

A mis sinodales y asesores, M.C. Abigail Zamora Hernández, M.I. Alberto Pedro Lorandi Medina, Dr. José Luis Vargas López, quienes con profesionalismo y paciencia infinita revisaron una y otra vez los borradores y propuestas, y con voluntad y sabiduría, guiaron este trabajo hasta su conclusión. Mil gracias y mi aprecio permanente. Las palabras de aliento del Director del Instituto de Ingeniería de la Universidad Veracruzana, Dr. Bonifacio C. A. Peña Pardo, me estimularon para redoblar esfuerzos en cada una de las etapas del trabajo.

Este proyecto se enriqueció con el entusiasmo y talento de dos jóvenes talentosos recién egresados de la Facultad de Estadística e Informática, Rocío Erandi Barrientos Martínez y Javier González Saiz, quienes no escatimaron ni tiempo ni conocimientos para apoyarme. Reciban mi aprecio y gratitud.

A mi hija Ivonne, quien me apoyó con amor y muchísimo entusiasmo en todas las actividades que se requirieron para realizar un documento de esta naturaleza. Gracias mi amor.



## **“Desarrollo de un sistema de control de estudiantes del Diplomado en Redes de Computadoras de la Facultad de Estadística e Informática por medio de un agente”**

### **ÍNDICE**

	Pág.
<b>1. INTRODUCCIÓN</b>	1
1.1 Antecedentes	4
1.2 Delimitación de la Problemática	6
1.3 Justificación	7
<b>2. MARCO TEÓRICO Y CONCEPTUAL</b>	8
2.1 Teoría de Sistemas de Información	8
2.2 Metodología Orientada a Objetos	10
2.3 Agentes	12
<b>3. MARCO DE REFERENCIA</b>	14
<b>4. HIPÓTESIS</b>	15
<b>5. OBJETIVOS</b>	15
<b>6. ANÁLISIS</b>	16
6.1 Contexto del Dominio del Problema	18
6.2 Identificación de Objetos	23
6.2.1 Definición de objetos textualmente	23
6.2.2 Definición de objetos gráficamente	24
6.3 Definición de Atributos.	26

6.4 Definición de Servicios	28
6.5 Definición de Estructuras	31
6.6 Definición de Asociaciones	32
6.7 Definición de Temas	34
6.7.1 Notación Colapsada	34
6.7.2 Notación Parcialmente Extendida	35
<b>7. DISEÑO</b>	<b>35</b>
7.1 Diseño de la Componente del Dominio del Problema	38
7.2 Diseño de la Componente de Interacción Humana.	39
7.3 Diseño de la Componente del Manejo de Tareas.	42
7.4 Diseño de la Componente de Administración de Datos	46
7.5 Definición de Tablas	47
7.5.1 Atributos de los Objetos de la base de datos.	52
7.6 Diagrama Entidad – Relación	61
7.7 Diccionario de datos.	63
<b>8. DESARROLLO</b>	<b>67</b>
8.1 Recursos para el Desarrollo del Sistema	69
8.2 Diagrama Jerárquico de Árbol	71
<b>9. IMPLANTACIÓN</b>	<b>72</b>
<b>10. VERIFICACIÓN Y VALIDACIÓN</b>	<b>72</b>
<b>11. PRUEBAS Y EVALUACIÓN DEL SISTEMA</b>	<b>74</b>
11.1 Pruebas de Unidad.	74
11.1.1 Pruebas Funcionales.	75
11.1.2 Pruebas de Desempeño.	77
11.1.3 Pruebas de Caja Blanca.	78
11.1.4 Pruebas de Caja Negra.	80

Instituto de Ingeniería  
Universidad Veracruzana

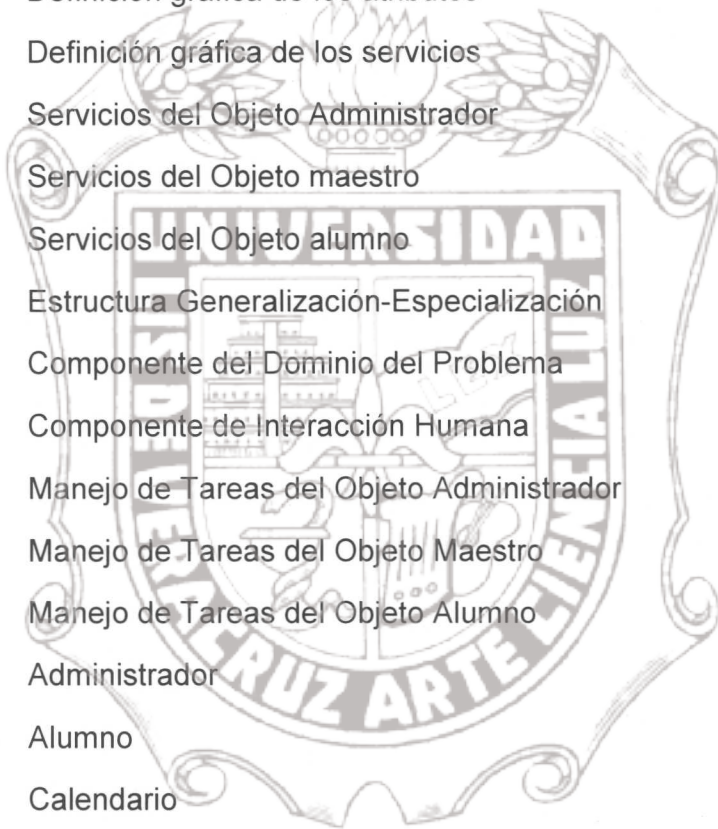
11.2 Pruebas de Validación.	85
11.3 Pruebas del Sistema	90
<b>12. MANTENIMIENTO</b>	96
<b>13. CONCLUSIONES</b>	97
<b>14. RECOMENDACIONES Y SUGERENCIAS</b>	99
<b>BIBLIOGRAFÍA</b>	100
<b>ANEXOS</b>	103
A. Instrucciones de Instalación.	105
B. Agentes de Interfaz.	106
C. Manual del Usuario.	107
D. Código.	118
<b>GLOSARIO</b>	148





## ÍNDICE DE FIGURAS

Número		Pág.
6.2.1	Identificación de Objetos	23
6.2.2	Definición gráfica de Objetos	24
6.3.1	Definición gráfica de los atributos	26
6.4.1	Definición gráfica de los servicios	28
6.4.2	Servicios del Objeto Administrador	29
6.4.3	Servicios del Objeto maestro	30
6.4.4	Servicios del Objeto alumno	31
6.5.1	Estructura Generalización-Especialización	32
7.1.1	Componente del Dominio del Problema	39
7.2.1	Componente de Interacción Humana	41
7.3.1	Manejo de Tareas del Objeto Administrador	42
7.3.2	Manejo de Tareas del Objeto Maestro	45
7.3.3	Manejo de Tareas del Objeto Alumno	46
7.4.1	Administrador	47
7.4.2	Alumno	47
7.4.3	Calendario	47
7.4.4	Kardex	47
7.4.5	Grupo	48
7.4.6	Horario	48
7.4.7	Maestro	48
7.4.8	Materia	48
7.4.9	Período	48
7.4.10	Atributos de la Tabla Administrador	49
7.4.11	Atributos de la Tabla Alumno	49



Instituto de Ingeniería  
Universidad Veracruzana

7.4.12	Atributos de la Tabla Calendario	49
7.4.13	Atributos de la Tabla Kardex	50
7.4.14	Atributos de la Tabla Grupo	50
7.4.15	Atributos de la Tabla Horario	50
7.4.16	Atributos de la Tabla Inscripción	51
7.4.17	Atributos de la Tabla Maestro	51
7.4.18	Atributos de la Tabla Materia	51
7.4.19	Atributos de la Tabla Periodo	52
7.5.1.1	Plantilla de Atributos del Objeto Alumno	53
7.5.1.2	Plantilla de Atributos del Objeto Calendario	54
7.5.1.3	Plantilla de Atributos del Objeto Kardex	55
7.5.1.4	Plantilla de Atributos del Objeto Grupo	56
7.5.1.5	Plantilla de Atributos del Objeto Horario	57
7.5.1.6	Plantilla de Atributos del Objeto Inscripción	58
7.5.1.7	Plantilla de Atributos del Objeto Materia	59
7.5.1.8	Plantilla de Atributos del Objeto Maestro	60
7.5.1.9	Plantilla de Atributos del Objeto Periodo	61
7.6.1	Modelo Entidad-Relación	63
7.7.1	Diccionario de Datos	63
8.1.1	Recursos Humanos	69
8.1.2	Recursos de Hardware	69
8.1.3	Recursos de Software	70
8.2	Diagrama Jerárquico de Árbol	71
11.1.1	Pruebas Funcionales	76
11.1.2	Pruebas de Desempeño	77
11.1.3	Pruebas de Caja Blanca	78

11.1.4	Pruebas de Caja Negra	80
11.2.1	Escala de Calificaciones	87
11.2.2	Nivel de Usuario	87
11.2.3	Categoría de Usuario	88
11.2.4	Resultado del Cuestionario	89
11.2.5	Pruebas de Seguridad	92
11.3.2	Pruebas de Recuperación	93
11.3.3	Pruebas de Rendimiento	95





## ÍNDICE DE TABLAS

Número		Pág.
6.2.1	Identificación de objetos.	23
6.4.2	Servicios del objeto Administrador	29
6.4.3	Servicios del objeto Maestro	30
6.4.4	Servicios del objeto Alumno	31
7.3.1	Manejo de Tareas del Objeto Administrador	42
7.3.2	Manejo de Tareas del Objeto Maestro	45
7.3.3	Manejo de Tareas del Objeto Alumno	46
7.4.1	Administrador	47
7.4.2	Alumno	47
7.4.3	Calendario	47
7.4.4	Kardex	47
7.4.5	Grupo	48
7.4.6	Horario	48
7.4.7	Maestro	48
7.4.8	Materia	48
7.4.9	Periodo	48
7.4.10	Atributos de la tabla Administrador	49
7.4.11	Atributos de la tabla Alumno	49
7.4.12	Atributos de la tabla Calendario	49
7.4.13	Atributos de la tabla Kardex	50
7.4.14	Atributos de la tabla Grupo	50
7.4.15	Atributos de la tabla Horario	50



Número		Pág.
7.4.16	Atributos de la tabla Inscripción	51
7.4.17	Atributos de la tabla Maestro	51
7.4.18	Atributos de la tabla Materia	51
7.4.19	Atributos de la tabla Periodo	52
7.5.1.1	Atributos del objeto Alumno	53
7.5.1.2	Atributos del objeto Calendario	54
7.5.1.3	Atributos del objeto Kardex	55
7.5.1.4	Atributos del objeto Grupo	56
7.5.1.5	Atributos del objeto Horario	57
7.5.1.6	Atributos del objeto Inscripción	58
7.5.1.7	Atributos del objeto Materia	59
7.5.1.8	Atributos del objeto Maestro	60
7.5.1.9	Atributos del objeto Periodo	61
7.7.1	Diccionario de Datos	63
8.1.1	Recursos Humanos	69
8.1.2	Recursos de Hardware	69
8.1.3	Recursos de Software	70
11.1.1	Pruebas Funcionales	76
11.1.2	Pruebas de Desempeño	77
11.1.3	Pruebas de Caja Blanca	78
11.1.4	Pruebas de Caja Negra.	80
11.2.1	Escala de calificaciones.	87

Número		Pág.
11.2.2	Niveles de usuario.	87
11.2.3	Categoría de usuarios.	88
11.2.4	Resultados del cuestionario.	89
11.2.5	Valores de Aceptación.	90
11.3.1	Pruebas de Seguridad	92
11.3.2	Pruebas de Recuperación	93
11.3.3	Pruebas de Rendimiento	95





## 1. INTRODUCCIÓN

La Universidad Veracruzana (UV), a los 55 años de su creación se ha convertido en la principal institución de educación superior en el Estado de Veracruz. Desde su inicio, se ha identificado por un espíritu de justicia social y ha asumido el deber de ofrecer y hacer participar de los beneficios de la educación y la cultura nacional y universal a todos los sectores de la sociedad.

El conjunto de programas de licenciatura, nivel técnico y de educación continua, ubican a la UV dentro de las cinco universidades públicas de provincia con mayor diversificación en su oferta educativa, con una población estudiantil de 65 mil alumnos en sus sistemas escolarizado, abierto y a distancia.

En los últimos años, la UV ha puesto mayor énfasis en el seguimiento profesional de sus egresados a través de la educación continua, ya que dicho proyecto ofrece alternativas para seguirse preparando ya sea en postgrado, diplomado o especialidad. Es por eso que la Facultad de Estadística e Informática (FEI) ofrece el Diplomado en Redes de Computadoras ( DRC ), que es un área con mucho campo de acción y de renovación constante.

En este contexto, considerando la necesidad de llevar un control y una trayectoria escolar de manera automatizada de los alumnos del DRC que imparte la Facultad de Estadística e Informática, se implementará un sistema que cubra de manera fácil y oportuna dicha necesidad, ya que a pesar de que actualmente la UV cuenta con un Sistema Integral de Información Universitaria (Banner 2000) para un mejor control de la misma, ha dejado como una opción a largo plazo la inclusión de los diplomados y postgrados que la Universidad ofrece. [DGB00]

Durante el proceso de la investigación y desarrollo del presente trabajo, se modificaron algunas premisas planteadas en el protocolo, sobresaliendo el planteamiento inicial de realizar el sistema a través de multiagentes, decidiéndose finalmente desarrollarlo a través de un sistema monoagente, ya que se pensó en hacer el sistema en dos partes: la contable y la administrativa, contando cada una de ellas con un agente que interactuara entre sí, pero el usuario final dijo no necesitar por el momento la parte contable y sólo nos limitaremos al control administrativo del DRC con su propio agente. Es por eso que se cambia de un sistema multiagente a un sistema monoagente.

El alcance de la investigación se circunscribe a desarrollar el software que permitirá llevar a cabo el control administrativo y el fácil acceso a la información del DRC de la FEI. Sin duda, trabajos posteriores podrán desarrollar áreas no contempladas en el presente trabajo.

SICADRC, que es el nombre con el cual designaremos el software aquí desarrollado ( Sistema de control de alumnos del Diplomado en Redes de Computadoras ) que controlará lo referente a la administración y control escolar del Diplomado, para poder conocer de forma rápida y de fácil acceso la información que se requiera. Para ello se recurrió al uso de un agente, cuyo principal propósito es servir de guía y apoyo a los usuarios que interactúen con el sistema.

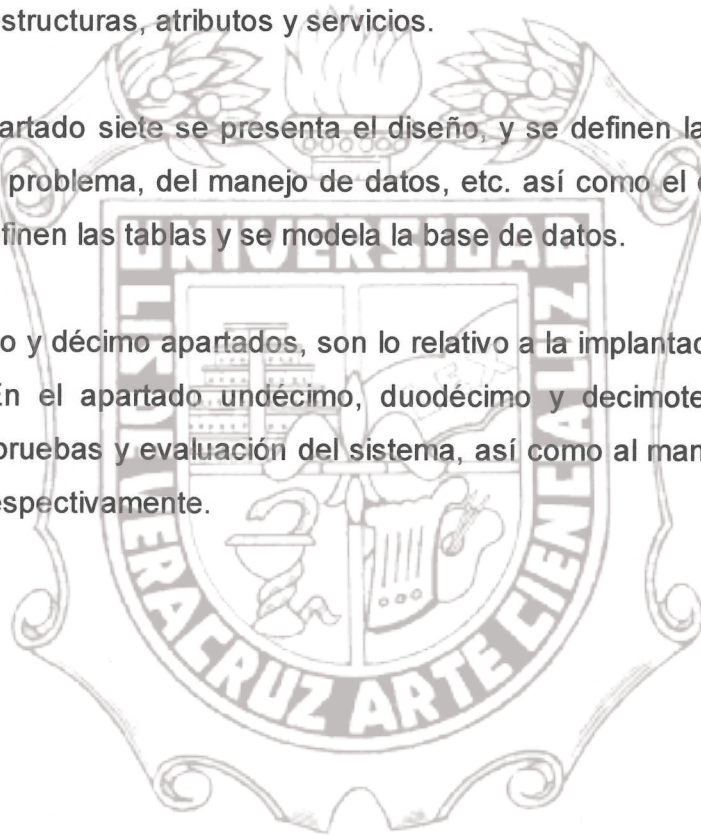
El presente trabajo se encuentra dividido en trece apartados que se centran en la planeación, análisis, diseño, desarrollo, pruebas y mantenimiento del sistema.

En los apartados uno al cinco, se presentan los aspectos generales que estuvieron involucrados en la planeación del proyecto, como son la problemática, justificación, hipótesis, objetivos y aspectos de funcionamiento del proyecto.

En el apartado seis se refleja el análisis. En el se define lo que será el dominio del contexto del problema, las necesidades y requerimientos sobre los cuales se va a trabajar. Se aplica el modelado orientado en lo que al análisis corresponde, el cual incluye actividades tales como identificación de objetos, asociaciones, estructuras, atributos y servicios.

En el apartado siete se presenta el diseño, y se definen las componentes del dominio del problema, del manejo de datos, etc. así como el diseño de datos en donde se definen las tablas y se modela la base de datos.

El noveno y décimo apartados, son lo relativo a la implantación, verificación y validación. En el apartado undécimo, duodécimo y decimotercero, se hace mención a las pruebas y evaluación del sistema, así como al mantenimiento y las conclusiones respectivamente.





## 1.1 Antecedentes

SICADRC es un proyecto que se pensó como solución inmediata a la necesidad que reclama el DRC de la Facultad de Estadística e Informática de llevar un control de los alumnos.

Como parte del programa de trabajo de la UV 1998-2001, "Consolidación y Proyección hacia el Siglo XXI", del Rector Dr. Víctor A. Arredondo Álvarez, se encuentra definida la estrategia de mejoramiento de la gestión universitaria en función de la automatización creciente de los procedimientos académicos y administrativos.

Para cumplir con este proyecto prioritario, la Universidad adquirió los soportes tecnológicos, de equipo y programas mas avanzados, que nos permitan desempeñar de manera protagónica la instrumentación del programa de normalización de la información administrativa y financiera establecida en los términos de referencia de la SEP y estar a la vanguardia a nivel Internacional.

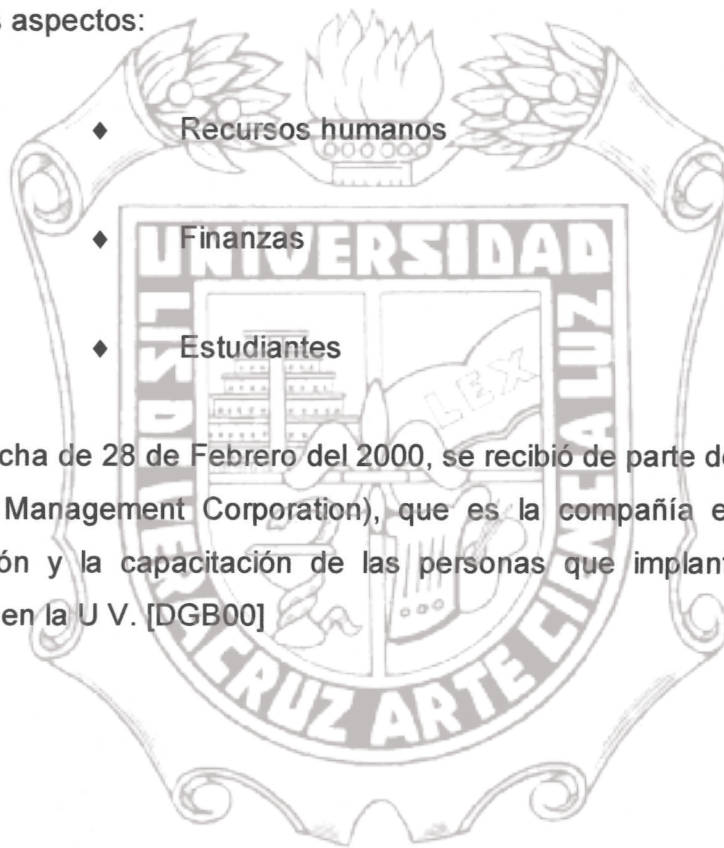
Para este fin, la UV adquirió el Sistema Integral de Información para la Administración Universitaria (Banner 2000) que permite la simplificación operativa para hacer más eficientes los servicios de información académica y administrativa, apoyar la adopción de modelos educativos más flexibles, agilizar las funciones sustantivas y adjetivas de la Institución, descentralizado los procesos que se realizan actualmente en la administración central de la Universidad. [DGB00]

El plan General de Trabajo y el alcance de Banner 2000, esta sustentado en el modelo académico actual y se divide en dos etapas:

La primera fue de Enero a Diciembre de 1999 y la segunda de Enero a Mayo del 2000, sustentando el plan en el modelo académico actual y comprende los siguientes aspectos:

- ◆ Recursos humanos
- ◆ Finanzas
- ◆ Estudiantes

Con fecha de 28 de Febrero del 2000, se recibió de parte de SCT (Software & Resource Management Corporation), que es la compañía encargada de la implementación y la capacitación de las personas que implantan actualmente Banner 2000 en la U V. [DGB00]



## 1.2 Delimitación de la Problemática.

A partir del 2 de octubre del año 2000, dentro del programa de educación continua de la UV, se ofrece en la Facultad de Estadística e Informática un DRC, el cual pretende cubrir las necesidades de actualización en el área de diseño, implementación y gestión de redes de computadoras de los egresados de dicha Facultad y de algunas carreras afines, así como a empleados de empresas paraestatales y privadas.

La aportación de este trabajo consiste en que SICADRC utiliza el concepto de Agente, ya que está implementado a través de un sistema monoagente, el cuál será capaz de actuar como secretario virtual para que asista en la administración del diplomado.

Dicho agente se encargará de controlar la administración del curso como son: los horarios, las materias y los maestros que se encuentren impartiendo algún curso del diplomado y a su vez también se encargara de llevar estadísticas de calificaciones y reportes.

El control del agente se basa en un solo super usuario que estará alimentando la base de datos con los cuales el agente utilizará cuando un usuario se conecte al sistema, en este caso supongamos que un maestro quiere saber su horario y las materias que le tocará impartir en este curso, el agente se encargará de mostrarle sus materias asignadas; así como pedir las calificaciones de cada alumno que se encuentre cursando la materia.

El agente al final de cada curso vuelve a reconocer la base de datos para así llevar un control nuevo en el siguiente curso.



Este diplomado no tiene la seguridad de ofrecerse de manera continua, ya que se encuentra condicionado a la demanda de estudiantes que se inscriban en los periodos correspondientes. Por ello, no se permiten las bajas temporales y por consiguiente, este rubro no estará comprendido en el programa.

En la actualidad el diplomado cuenta con tres grupos, uno que asiste de lunes a sábado de 7:00 a 9:00 hrs., otro de lunes a sábado de 19:00 a 21:00 hrs. y el tercero los sábados de 9:00 a 14:00 y de 18:00 a 21:00 hrs., contando con un total de 25 alumnos por grupo, los cuales son seleccionados con una entrevista previa con el coordinador del diplomado y son ellos mismos los que deciden a que horario asistirán. No existen requerimientos adicionales ni procedimientos para el registro de alumnos.

### 1.3 Justificación

El presente proyecto surge de la necesidad de tener un software para el control de alumnos del DRC de la Facultad de Estadística e Informática (FEI), debido a que dicho control actualmente se lleva a cabo de manera manual y no se encuentra contemplado ni a corto ni a mediano plazo su implementación dentro del proyecto de Banner 2000.



## 2. MARCO TEÓRICO Y CONCEPTUAL

### 2.1 Teoría de Sistemas de Información.

Un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Es el Conjunto total de procedimientos, operaciones, funciones y difusión de datos o información en una organización. [SEN92]

Las tres partes fundamentales de un sistema de información son el sistema de computación, el sistema de numeración y el sistema Operativo. Estos elementos son de naturaleza diversa y normalmente incluyen:

- El equipo computacional, es decir, el hardware que es necesario para que el sistema de información pueda operar. Lo constituyen las computadoras y el equipo periférico que pueden conectarse a ellas.
- El recurso humano, que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema, alimentándolo con datos o utilizando los resultados que genere.
- Los datos o información fuente, que son introducidos en el sistema, son todas las entradas que necesita él sistema de información para generar como resultado la información que se desea.
- Los programas, que son procesados y producen diferentes tipos de resultados. Los programas son la parte del software del sistema de información que hará que los datos de entrada introducidos sean procesados correctamente y generen los resultados que se esperan.

Es útil mencionar brevemente cada una de las fases de la investigación de sistemas de información [KEN96]:

1. Fase de estudio preliminar. Durante esta fase, con un sistema de información existente se descubre un problema o una oportunidad de desarrollar útilmente un nuevo sistema, y se lleva a cabo una cantidad limitada de investigación preliminar para ver si un proyecto de sistemas está garantizado.
2. Fase de análisis de sistemas. Durante la fase de análisis, se identifica un problema u oportunidad asociada con el sistema, se examinan los puntos débiles y fuertes del sistema antiguo, y se determina para qué serviría un nuevo sistema.
3. Fase de diseño de sistemas. Durante esta fase se diseña un nuevo sistema o una aplicación computarizada para satisfacer las necesidades que se han determinado durante la fase de análisis. Asimismo se completarán, tanto los estudios de hardware como el diseño del software.
4. Fase de implantación. Esta fase involucra la programación, instalación de equipo y otras actividades relacionadas con la implantación de un sistema diseñado.
5. Fase de Pruebas. Una vez que se ha generado un código, inician las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales, es decir, la realización de las pruebas para la detección de errores y el sentirse

seguro de que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

6. Fase de Mantenimiento. El software indudablemente sufrirá cambios después de haber sido entregado al cliente, se producirán cambios porque se han encontrado errores, ya que el software debe adaptarse para acoplarse a los cambios de su entorno externo o porque el cliente requiere mejoras funcionales o de rendimiento. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo.

## 2.2 Metodología Orientada a Objetos

¿Qué es un objeto? : Cualquier entidad individual discreta, que pertenece al menos a una clase, que está dotado de propiedades, que es capaz de realizar determinados procesos y cuya simple apariencia oculta la complejidad de su construcción.

Un objeto está compuesto de:

- **Identidad única:** Es la propiedad de un objeto, la cual permite distinguirlo de los demás objetos existentes.
- **Estado:** Se define sobre la base de los valores de sus propiedades.
- **Comportamiento:** Es como un objeto actúa y reacciona, en términos de los cambios de estado y las operaciones que realiza.

La orientación de objetos es una técnica para modernizar sistemas, superficialmente el término orientado a objetos, significa que el software se



organiza como una colección de objetos discretos que incorporan tanto estructura de datos como comportamiento y que están relacionados entre sí.

Según Peter Wegner, se requieren satisfacer las siguientes características para así lograr un enfoque completamente orientado a objetos:

- Abstracción
- Encapsulamiento
- Clasificación
- Polimorfismo
- Herencia

La **abstracción** es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema mientras suprime otras. Una buena abstracción es la que enfatiza los detalles que son significativos al usuario, y suprime aquellos que son al menos en ese instante, irrelevantes.

El **encapsulamiento** es el proceso de ocultación de todos los detalles de un objeto que no contribuyen a sus características esenciales. La abstracción y el encapsulamiento son conceptos complementarios: Mientras que la abstracción se concentra en el comportamiento observable de un objeto, el encapsulamiento nos permite concentrarnos en la implementación que da origen a dicho comportamiento y el encapsulamiento se logra a través del ocultamiento de la información, el cual, es el proceso de ocultar todos los secretos de un objeto, que no contribuyen a sus características esenciales, típicamente, la estructura de un objeto está oculta, así como también la implementación de sus métodos.

La **clasificación** significa que los objetos con la misma estructura de datos y comportamiento, son agrupados dentro de una misma clase. Una clase es una



abstracción que describe las propiedades importantes, e ignora el resto, cada clase describe un posible conjunto infinito de objetos, se dice que cada objeto es una instancia de su clase, cada instancia de una clase tiene valores propios para cada uno de los atributos.

**Polimorfismo** significa que la misma operación puede tener un comportamiento diferente en clases diferentes.

**Herencia** es la manera en la que las clases se comparten atributos y operaciones, basándose en una relación de jerarquía.

### 2.3 Agentes.

Los agentes inteligentes es un tema ampliamente estudiado, debido a la gran ayuda que proporciona para la solución de problemas complejos. La teoría de agentes proporciona herramientas, técnicas y metáforas que son útiles en la conceptualización e implementación de software de muchos tipos. Un agente es un sistema computacional situado en algún ambiente, capaz de actuar autónomamente y dotado de flexibilidad para alcanzar sus objetivos [JEN98].

Analicemos esta definición en la que resaltan tres conceptos:

1. - **Situacionalidad:** Puede trabajar en un ambiente percibiendo y respondiendo a los cambios que ocurran dentro del ambiente.

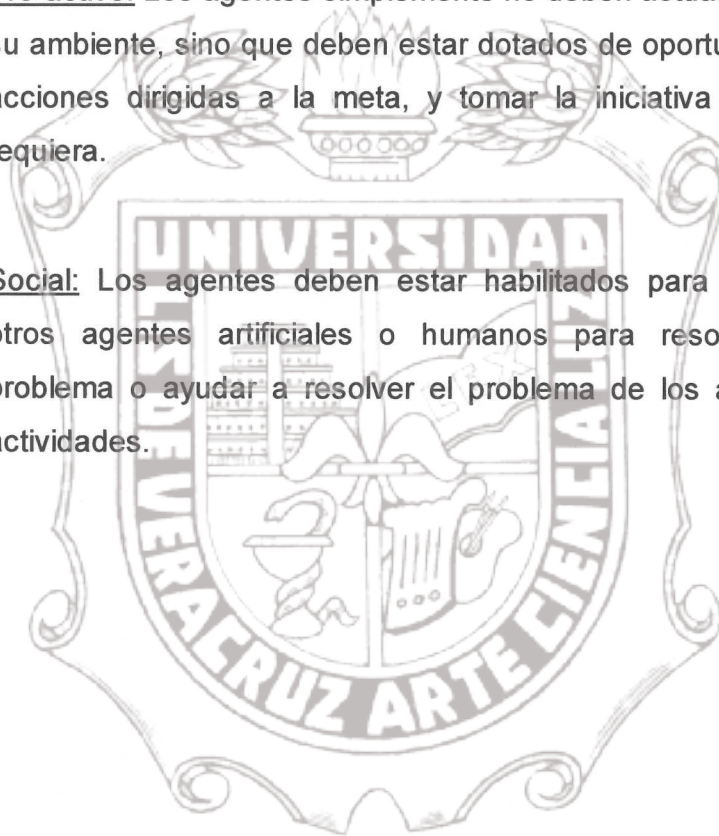
2. - **Autonomía:** Actúa por sí solo, es decir, toma sus propias decisiones sin la intervención directa de ningún humano o de otros agentes. En sentido más estricto, la autonomía se relaciona con la capacidad de aprendizaje del agente de su propia experiencia.

### 3. – Flexibilidad: Se presenta cuando el sistema es:

Sensitivo o sensible: Los agentes perciben su ambiente y responden a los cambios que ocurran en él.

Pro-activo: Los agentes simplemente no deben actuar en respuesta a su ambiente, sino que deben estar dotados de oportunismo, ejecutar acciones dirigidas a la meta, y tomar la iniciativa cuando así se requiera.

Social: Los agentes deben estar habilitados para interactuar con otros agentes artificiales o humanos para resolver su propio problema o ayudar a resolver el problema de los agentes en sus actividades.



### 3. MARCO DE REFERENCIA

La UV a través de la coordinación de Banner, está implantando el software Banner 2000, el cual, entre otras cosas, ofrecerá a los académicos y personal administrativo toda la información pertinente al control de ingreso de alumnos y al control de su trayectoria escolar, así como también realizará tareas relacionadas con los trámites escolares y la administración de las diferentes dependencias de la UV. Sin embargo, el control y las ventajas que ofrece el uso de Banner 2000, por el momento se enfocará solamente a las licenciaturas y a los técnicos medio superiores, dejando a un lado los programas de especialización y postgrado.

A la fecha, la Facultad de Estadística e Informática no cuenta con un software para el control de alumnos de sus licenciaturas y será hasta el presente ciclo escolar septiembre 2001- Febrero 2002, que se implemente el sistema Banner 2000 para llevar a cabo dicho control. Como ya se estableció en la justificación, dentro de Banner 2000, por el momento no está contemplado el control de alumnos del DRC que se imparte en la FEI.

Por este motivo y dado que se pretende tener un control automatizado de los alumnos del diplomado, se tomaron como referencia algunos aspectos que incluye el Sistema Integral de Información Universitaria (Banner 2000) del módulo de estudiantes, para el desarrollo e implementación SICADRC.



## 4. HIPÓTESIS

La implantación de un sistema de control escolar apoyado en un agente en la Facultad de Estadística e Informática para el DRC, permitirá a los interesados obtener información requerida sin intermediación de terceros que retarden el acceso a la misma.

## 5. OBJETIVOS

### General

Desarrollar un sistema para resolver las necesidades administrativas del DRC impartido en la Facultad de Estadística e Informática de la Universidad Veracruzana.

### Específicos

- Se utilizará en el DRC de la FEI de la UV
- Podrá utilizarse en otros diplomados que se impartan.
- Ahorrará tiempo a las personas que necesiten utilizar la información del sistema.
- Agilizará las consultas de información requeridas por los usuarios.
- Reducirá la carga de trabajo del personal del diplomado.
- Existirá una cuota mínima para el uso de este software.





## 6. ANÁLISIS

El paradigma que se utilizará para llevar a cabo el desarrollo de SICADRC es el orientado a objetos, considerando de éste la metodología de **Peter Coad** y **Edward Yourdon**, la cual cuenta con las siguientes características en la etapa del análisis:

- Contexto del dominio del problema.
- Definición de objetos:
  - ✓ Textualmente.
  - ✓ Gráficamente.
- Definición de atributos.
- Definición de servicios.
- Definición de estructuras.
- Definición de temas o subjects.

Para poder tener una mejor comprensión de los objetos que se están definiendo, se presenta la descripción de los mismos tanto en forma textual como gráfica. La definición de atributos (datos o campos) y la definición de servicios (métodos u operaciones); nos brindan una mejor comprensión de lo que es realmente el objeto mismo. Según la metodología de Coad y Yourdon, la definición de estructuras nos ayudan a acotar la complejidad del problema. Estas pueden ser:

- Estructuras Generalización-Especialización.
- Estructuras Todo-Parte.
- Estructuras de asociación.

La definición de temas (subjects), nos sirve para realizar una representación de la agrupación que pueden tener los objetos por sus similitudes, características, etc. Para tal caso dichos temas se pueden representar de la siguiente manera: notación colapsada y notación parcialmente expandida.

Esta metodología utiliza conceptos como objetos, atributos, clases, estructuras de especialización. Las etapas que se deben llevar a cabo para el desarrollo del software son:

- Análisis
- Diseño
- Implantación
- Pruebas
- Mantenimiento

El análisis es un proceso de descubrimiento, refinamiento, modelado y especificación. En esta etapa, se refina a detalle el ámbito de software, se crean modelos de los requisitos de datos, flujo de información, de control y del comportamiento operativo. [PRE97]

Tanto el desarrollador como el cliente tienen un papel activo en el análisis y especificación de requisitos. El cliente trata de definir lo que desea del software y el desarrollador actúa como interrogador, consultor y como persona que resuelve problemas. [PRE97]

En esta etapa se recopila información de todas las personas que están involucradas en los procesos que realiza la organización para la cual se desarrollará el software. Se reúnen detalles, opiniones, posibles soluciones e ideas que mejorarán de alguna forma los procesos. Todas estas actividades se realizan con el objeto de que los analistas puedan estudiar los requerimientos, para así poder identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con las características operacionales ( p. ej : entradas y salidas).

El análisis del sistema pudiera parecer una tarea fácil, pero es importante, ya que se pueden hacer malas interpretaciones de las necesidades de los usuarios u obtener poca información, es por eso que el desarrollador o analista debe valerse de los medios adecuados para obtener información relevante, que le será de utilidad para refinar la definición del software y construir los modelos de dominio de datos, que van a ser tratados por el software [PRE97].

## 6.1 Contexto del dominio del problema.

Con una entrevista al administrador del DRC que se imparte en la FEI se pudo constatar cómo se lleva a cabo el control de los alumnos del diplomado tomando en cuenta el calendario, kardex, horario y grupo al cual pertenecen y cuáles son las deficiencias con las que cuenta dentro del proceso administrativo del diplomado.

Un aspecto fundamental para requerir la implementación de un sistema informático para el DRC, es que no existe una persona específica que se responsabilice de llevar a cabo el control del diplomado, esto es, que se encargue de los trámites administrativos o cualquier otro servicio para los alumnos o maestros.



Adicionalmente, dicho proceso se realiza de manera manual por los prestadores de servicio social, motivo por el cual muchas veces la información puede extraviarse y el acceso a ella no siempre es tan oportuno como se quisiera, lo cual ocasiona que haya desfases en las materias o los períodos escolares, pues los maestros solo pueden basarse en esta información.

La entrevista que se llevo a cabo consta de las siguientes preguntas:

**1. ¿Cuál es la principal causa o justificación para requerir un sistema informático para el DRC?**

R= Existe un aspecto fundamental el cual es que realmente no existe una persona fija que tenga el control del diplomado de redes de computadoras, con esto nos referimos a que se encargue de los trámites necesarios o servicios ya sea para inscripción, calificaciones, horarios o cualquier otro servicio para el alumno.

**2. ¿Que horario de clases se imparten actualmente?**

R=Existen 3 turnos con los cuales estamos impartiendo clases, un turno matutino (Lunes a Sábado de 7:00 AM. a 9:00 AM.) Un turno vespertino (Lunes a Sábado de 7:00 PM a 9:00 PM.) y uno los fines de semana (Sábados de 9:00 AM. a 2:00 PM. y 6:00 PM. 9:00 PM.)

**3. ¿Se aplica algún examen de admisión?**

R=Sé decidió que no se aplicara examen puesto que se trata de que la comunidad estudiantil, principalmente los egresados, acudan a este diplomado y que aprendan. Cabe mencionar que se aplica una entrevista para evaluar el grado de conocimiento del ingresante para así poder percatarse desde que nivel de enseñanza se va a comenzar.



#### 4. ¿Que método de evaluación se aplica a los alumnos para aprobar una materia?

R= La idea de este diplomado, como se mencionó anteriormente, es de que el egresado se acerque nuevamente a estudiar, incluso tenemos personas de mas de 40 años y sabemos que es muy difícil tratar de superarse tomando en cuenta un oficio que se desempeña diariamente; precisamente por eso el método de evaluación es a criterio del maestro y no se pasa lista de asistencia.

#### 5. ¿Que periodo de clases tiene este diplomado?

R= Se programó el curso del diplomado de redes a no más de un año; puesto que este es nuestro primer año, todavía tenemos a prueba que tanto se abarca durante este tiempo, para que posteriormente tomando en cuenta esta experiencia, se mejore la calidad del curso y su periodo.

El plan de estudios del DRC se encuentra dividido en 7 módulos con una duración de 30 horas cada uno, el contenido de cada uno de los módulos es el siguiente:

##### 1. Introducción a las redes de computadoras

- Conceptos de comunicaciones
- Tipos de redes de computadoras
- Topología de redes
- Elementos de una red
- Medios de transmisión
- Ventajas y desventajas del uso de redes de computadoras
- Aplicaciones

- Inter conectividad de redes

## 2. Protocolos LAN/WAN

- Ethernet
- Fast Ethernet
- Token Ring
- FDDI, CDDI
- ATM

## 3. TCP/IP

- Capas del modelo
- Capa de red
- Capa de transporte
- Capa de enlace
- Capa de aplicación

## 4. Redes Microsoft

- Protocolos de NetBIOS
- Windows NT
- Administración de usuarios
- DHCP
- Servidor de WINS
- Internet Information Server

## 5. Linux

- Instalación
- Administración del servidor



- DNS
- DHCP
- Correo electrónico
- SAMBA
- FTP
- NFS
- Servidor WEB
- Servidor PROXY

## 6. Programación en la web

- Conceptos
- HTML
- XML
- CGI's
- PHP, ASP
- JavaScript, VBScript
- Java

## 7. Políticas, procedimientos y seguridad en la red

- Políticas y procedimientos para la red
- Conceptos básicos de seguridad
- Herramientas de seguridad
- Seguridad en Linux





## 6.2 Identificación de Objetos

Haciendo un análisis de la problemática que se tiene en el Diplomado de Redes de Computadoras de la Facultad de Estadística e Informática, se han identificado los posibles objetos:

Administrador	Objeto
Alumno	Objeto
Calendario	Objeto
Kardex	Objeto
Grupo	Objeto
Horario	Objeto
Maestro	Objeto
Materia	Objeto
Período	Objeto

Tabla 6.2.1 Identificación de objetos.

### 6.2.1 Definición de objetos textualmente.

- **Administrador**: Persona que se encarga de manejar el sistema.
- **Alumno**: Persona que cursa el DRC.
- **Calendario**: Son las fechas en las cuales se realizan actividades ya establecidas dentro del DRC.
- **Kardex**: Es el documento donde se lleva el control de calificaciones del alumno.
- **Grupo**: Lugar físico donde el alumno toma sus clases
- **Horario**: Son las horas en que el alumno debe realizar alguna actividad.

- **Maestro:** Persona que imparte clases a los alumnos del DRC.
- **Materia:** Es la asignatura que se imparte en el DRC por medio de un maestro.
- **Período:** La cantidad de tiempo que se necesita para cursar el DRC.

## 6.2.2 Definición de objetos gráficamente.

Una vez definidos los objetos textualmente, se hace una representación gráfica de los mismos, como se muestra a continuación.

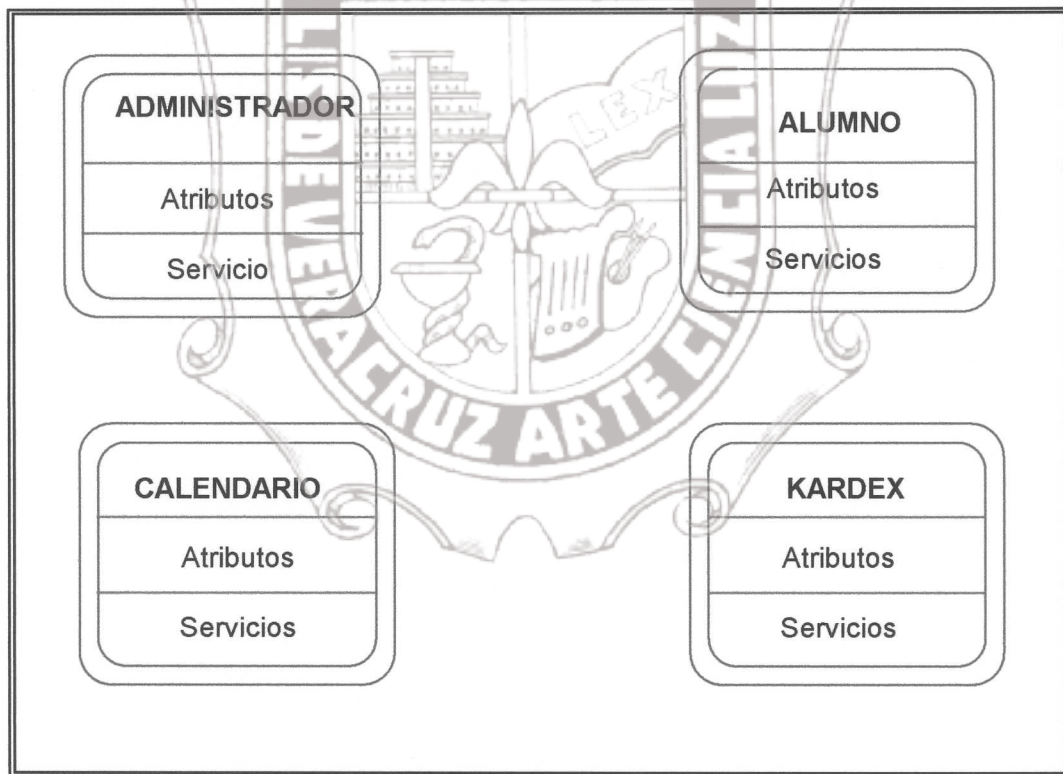


Figura No 6.2.2.1 Definición gráfica de objetos

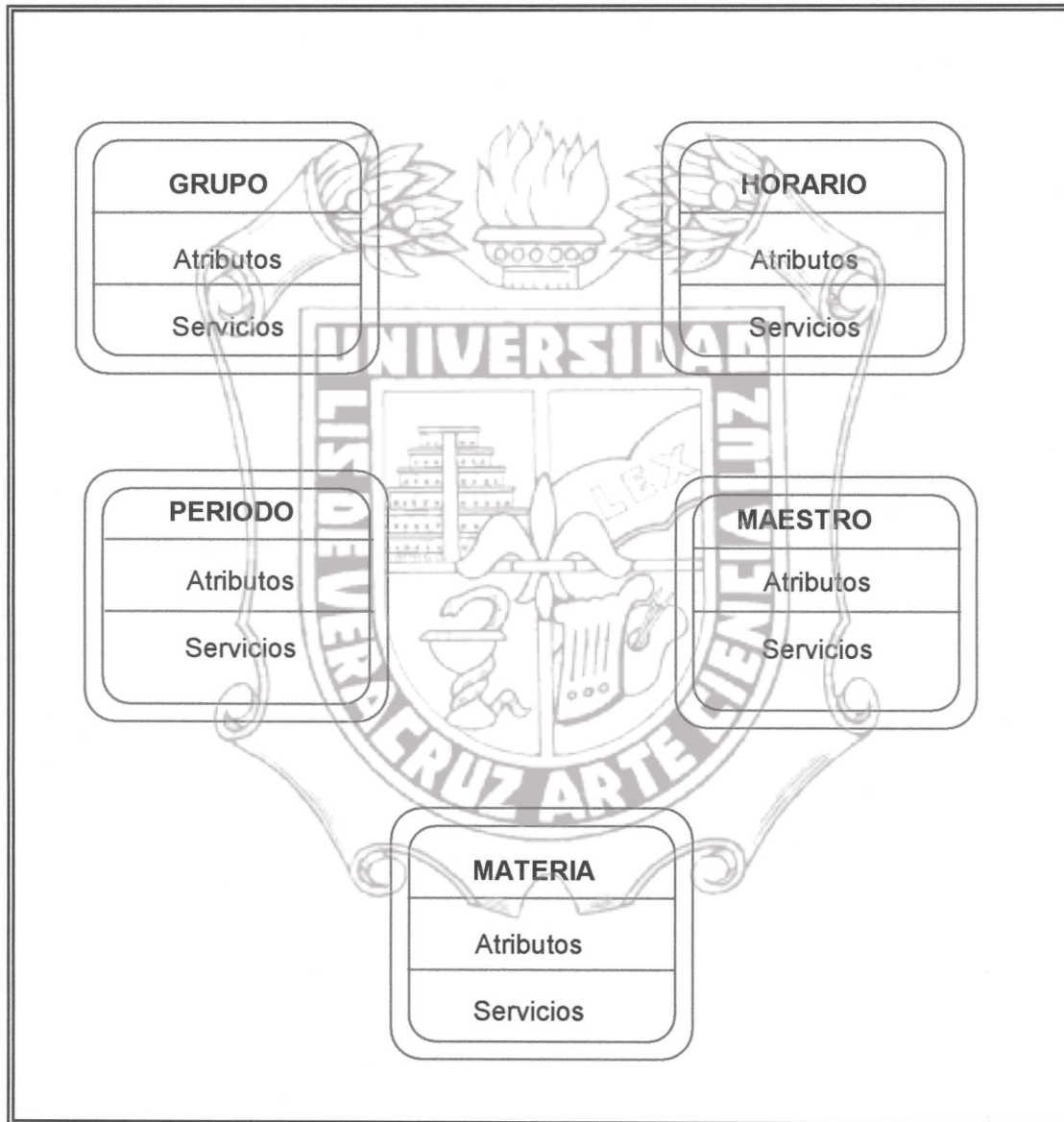


Figura No 6.2.2.1 Definición gráfica de objetos (continuación)



## 6.3 Definición de atributos.

En este apartado se muestran los atributos de cada uno de los objetos con el propósito de dar una idea mas clara de la información que conforma a cada uno de ellos, ya que los atributos son los que nos describen de manera única a cada objeto.

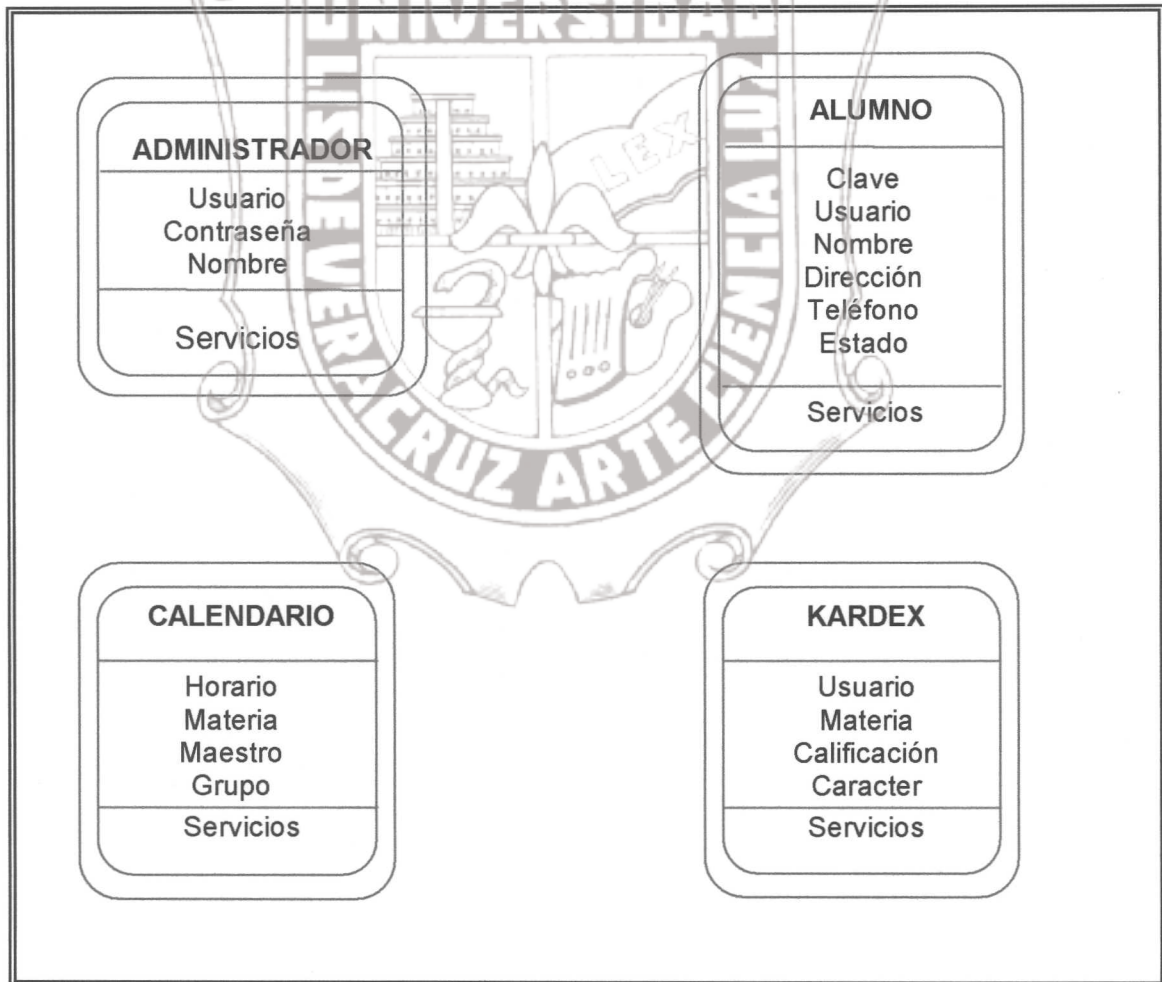


Figura No 6.3.1. Definición gráfica de los atributos.

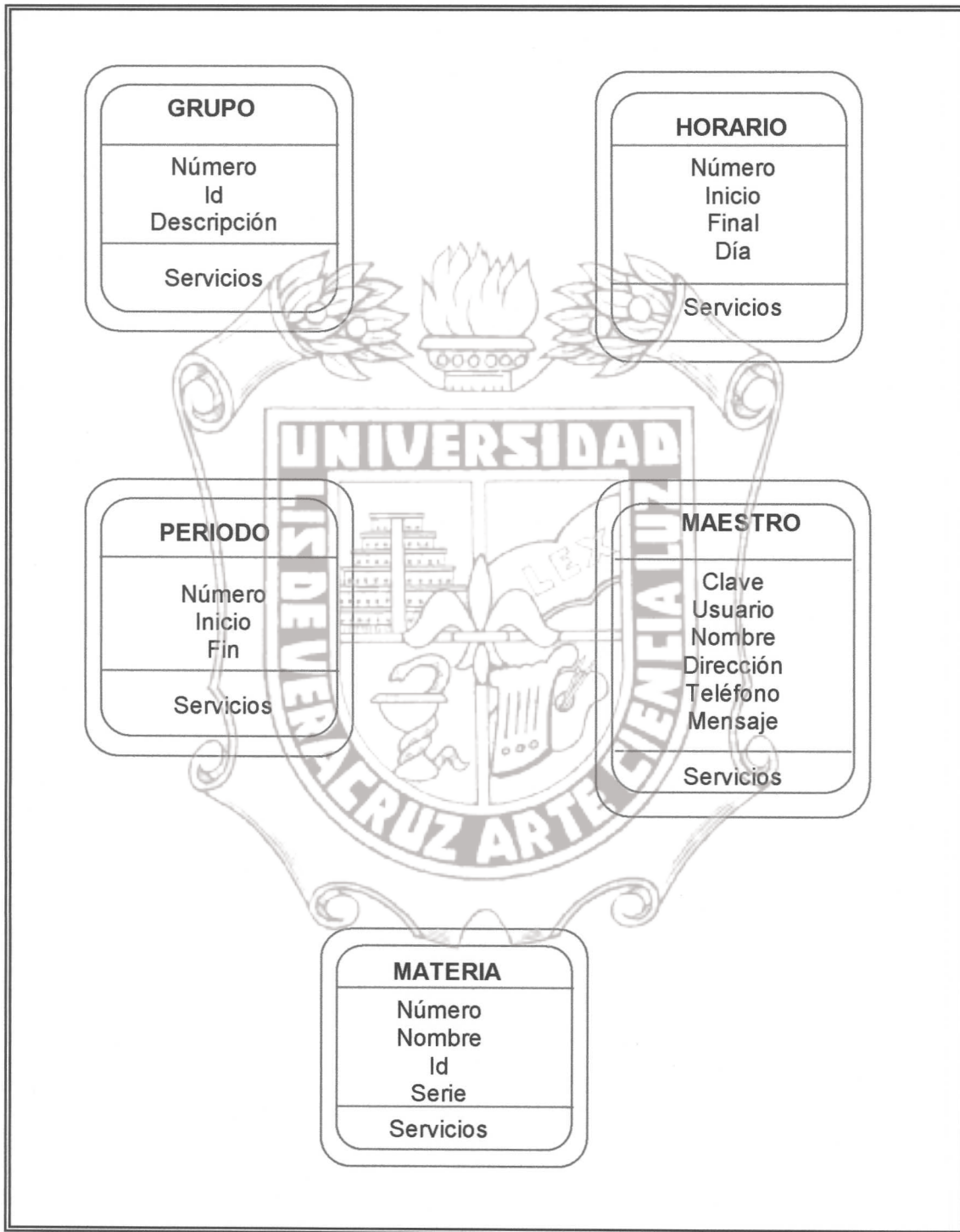


Figura No 6.3.1 Definición gráfica de los atributos (continuación)

## 6.4 Definición de Servicios.

Por último se definen los servicios o métodos de los que hará uso cada uno de los objetos, para posteriormente establecer la comunicación que habrá entre cada uno de ellos a través del paso de mensajes, a continuación se enlistan los principales objetos que serán los responsables de que SICADRC funcione con los servicios que cada uno de ellos representa.

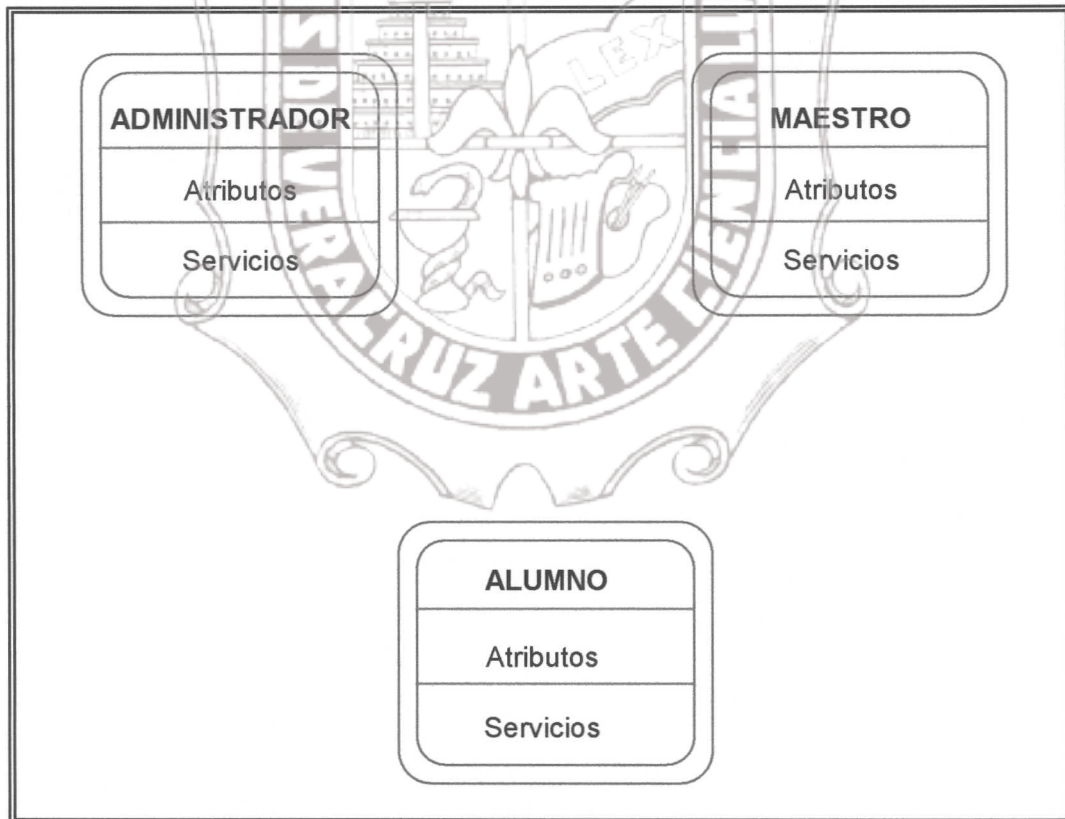


Figura No 6.4.1 Definición gráfica de los servicios.



## ADMINISTRADOR

Alta Alumno ( )	Agrega un alumno a la Base de datos.
Alta Maestro ( )	Agrega un maestro a la Base de datos.
Alta Materia ( )	Agrega una materia a la Base de datos.
Alta Horario ( )	Agrega un nuevo periodo de horas a la Base de datos.
Alta Calendario ( )	Agrega un nuevo calendario de clases a la Base de datos.
Alta Grupo ( )	Agrega un nuevo grupo a la Base de datos.
Baja Alumno ( )	Elimina alumnos de la Base de datos.
Baja Maestro ( )	Elimina maestros de la Base de datos.
Baja Materia ( )	Elimina materias de la Base de datos.
Baja Horario ( )	Elimina periodos de horas de la Base de datos.
Baja Calendario ( )	Elimina un calendario de clases a la Base de datos.
Baja Grupo ( )	Elimina grupos de la Base de datos.
Cambia Alumno ( )	Modifica los datos de un alumno existente en la Base de datos.
Cambia Maestro ( )	Modifica los datos de un maestro existente en la Base de datos.
Cambia Materia ( )	Modifica los datos de una materia existente en la Base de datos.
Cambia Horario ( )	Modifica los datos de un periodo de horas existente en la Base de datos.
Cambia Calendario ( )	Modifica los datos de un calendario de clases existente en la Base de datos.
Cambia Grupo ( )	Modifica los datos de un grupo existente en la Base de datos.

Tabla No 6.4.2 Servicios del objeto Administrador

Consulta Alumno ()	Busca los datos de un alumno que exista en la Base de datos.
Consulta Maestro ()	Busca los datos de un maestro que exista en la Base de datos.
Consulta Materia ()	Busca los datos de una materia que exista en la Base de datos.
Consulta Horario ()	Busca los datos de un periodo de horas que exista en la Base de datos.
Consulta Calendario ()	Busca los datos de un calendario de clases que exista en la Base de datos.
Consulta Grupo ()	Busca los datos de un grupo que exista en la Base de datos.
Realiza Inscripción ()	Agrega los datos del alumno al concentrado de inscripción.

Tabla No 6.4.2 Servicios del objeto Administrador (continuación)

## MAESTRO

Alta Calificación ()	Agrega las calificaciones al Kardex del alumno.
Cambia Calificación ()	Modifica las calificaciones existentes en el Kardex del alumno.
Consulta Calificación ()	Busca las calificaciones en el Kardex del alumno.
Envía Mensaje ()	Permite remitir mensajes a sus alumnos.
Recibe Mensaje ()	Permite recibir mensajes de sus alumnos

Tabla No 6.4.3 Servicios del objeto Maestro

## ALUMNO

Consulta Calificación ()	Busca sus calificaciones en su Kardex correspondiente.
Consulta Calendario ()	Busca su correspondiente calendario de clases existente en la Base de datos.
Consulta Kardex ()	Revisa los datos almacenados en su Kardex correspondiente.
Envía Mensaje ()	Permite remitir mensajes a sus maestros.
Recibe Mensaje ()	Permite recibir mensajes de sus maestros.

Tabla No 6.4.4 Servicios del objeto Alumno

## 6.5 Definición de Estructuras

Para un dominio de problema específico, las abstracciones suelen estar relacionadas formando lo que se llaman estructuras. En el sistema de control de alumnos del DRC de la Facultad de Estadística e Informática, se encontró solamente un tipo de estructura, que es la Generalización-Especialización la cual representa una clasificación como se puede apreciar en la figura 6.5.1.

La justificación de dicha estructura en este proyecto es porque las clases y objetos Alumno, Maestro y Administrador, tienen los mismos atributos y servicios pero además cada uno de ellos tiene atributos propios que los diferencian.



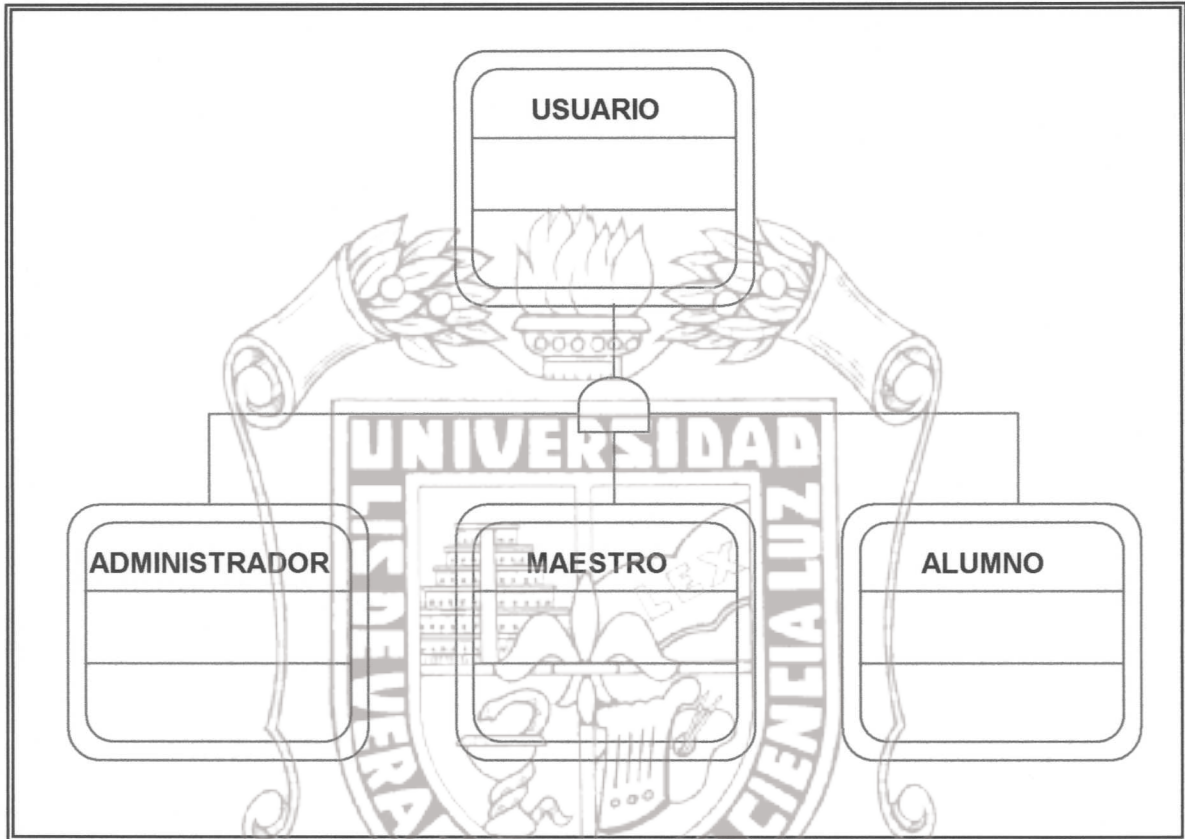
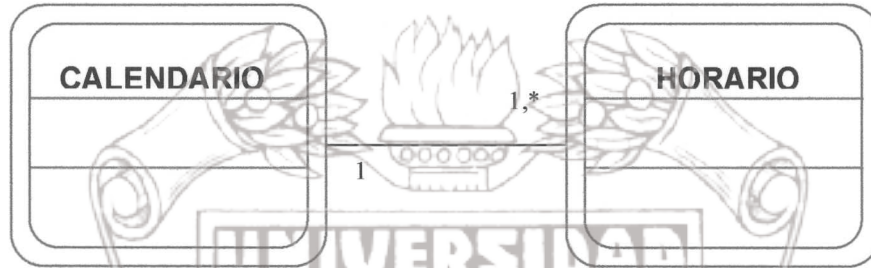


Figura 6.5.1 Estructura Generalización-Especialización

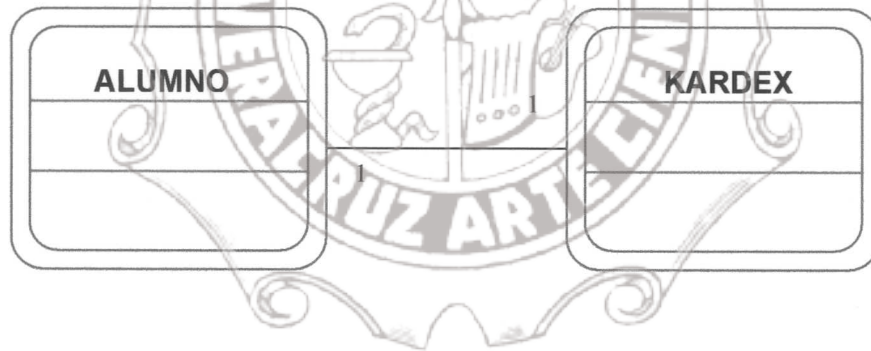
## 6.6 Definición de Asociaciones.

Una asociación identifica la relación que existe entre dos abstracciones [COAD91]. Para cada asociación se debe especificar el papel de cada participante, así como cualquier cardinalidad que exista. Las asociaciones que se pueden encontrar en el sistema son:

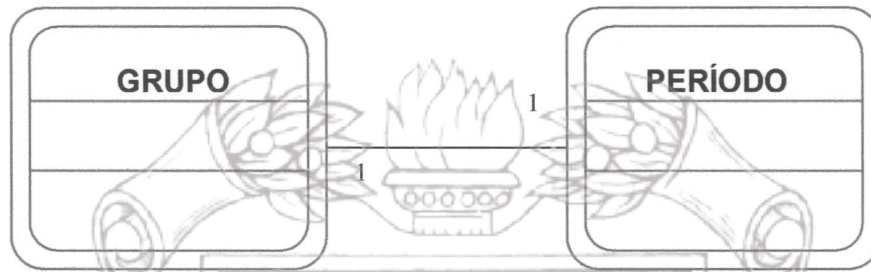
Un calendario de clases puede involucrar uno o muchos horarios, maestros, grupos y materias en un periodo y un horario puede estar involucrado en uno o ningún calendario, un maestro puede o no estar involucrado en un calendario y una materia puede estar o no involucrada en un calendario.



- ◆ Un alumno va a estar inscrito en un solo Kardex y un Kardex solo puede tener inscrito un alumno.



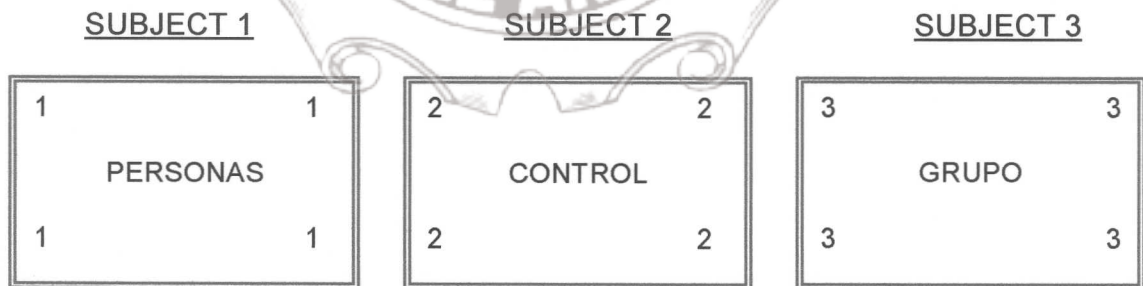
Un grupo puede realizar muchas inscripciones por periodo y una inscripción puede ser realizada por uno o muchos grupos.



## 6.7 DEFINICIÓN DE TEMAS

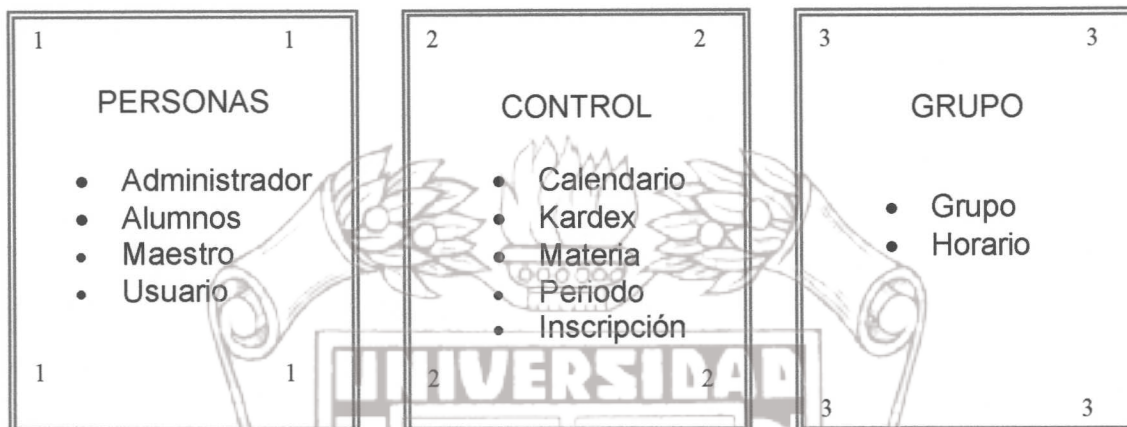
La definición de temas se hizo con la finalidad de obtener una visión más clara del dominio del problema, para saber que es de lo que estamos hablando y que es lo que estamos tratando.

### 6.7.1 Notación Colapsada





## 6.7.2 Notación Parcialmente Expandida



## 7. DISEÑO

Dentro del diseño de un sistema, se ven los detalles que establecen la forma en que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. En este proceso se lleva a cabo la traducción de requisitos a una representación del software que se pueda evaluar por calidad, antes de la generación del código [PRE97].

A medida que pasa el tiempo, las tecnologías de objetos sustituirán a los enfoques clásicos de desarrollo de software. Una pregunta importante es ¿por qué?.

Algunas personas argumentarían que los profesionales del software sencillamente añoraban un nuevo enfoque, pero esta visión es muy simplista. Las tecnologías de objeto llevan un número de beneficios inherentes que proporcionan ventajas a los niveles de dirección y técnico. El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente descompuesta.

Esto lleva a menores efectos colaterales cuando se deben hacer cambios, son más fáciles de adaptar y escalar, las tecnologías de objeto llevan a reutilizar y la reutilización lleva a un desarrollo de software más rápido y programas de mayor calidad.

El Diseño Orientado a Objetos al igual que otras metodologías, permite crear una representación del mundo real que puede ser realizado en software y a diferencia de otras, producen un diseño que interconecta objetos de datos. Los objetos proporcionan un mecanismo para representar el ámbito de la información, mientras que las operaciones describen el procesamiento asociado con el ámbito de la información. Los mensajes proporcionan el medio mediante el cual son invocadas las operaciones.

Las funciones que el agente desempeñará son mostradas a continuación:

El propósito de incluir un agente dentro de SICADRC es contar con un secretario virtual que le permita a cada uno de los usuarios interactuar y obtener beneficios del mismo con mayor facilidad, tomando en cuenta que el mismo agente realiza actividades de forma autónoma en algún determinado tiempo o acción del mismo usuario, con esto se hace más flexible el uso y la realización de actividades dentro del sistema.

El tipo de agente que se implementará en este sistema es el de un agente de interfaz (ver anexo B Pág. 109) que realizara las siguientes actividades:

#### **Para el administrador:**

- Generar si así lo desea el administrador, el mismo horario de clases para el siguiente periodo, tomando como base el horario del periodo anterior.

- Asignar a los maestros las mismas asignaturas.
- En el caso en que se termine algún periodo del diplomado y no se halla dado de alta otro, preguntará al administrador si desea dar de alta otro periodo con la misma duración del periodo anterior.

#### Para los maestros:

- Enviar a todos los alumnos de dicho maestro algún mensaje que éste desee enviar; el maestro solo escribirá el mensaje y el agente se encargará de enviarlo a todos los alumnos.
- Avisar si tiene mensaje de algún alumno.
- Avisar cuando se haga algún cambio administrativo con sus materias.

#### Para los alumnos:

- Avisar cuando tenga mensaje de algún maestro.
- Avisar cuando tenga calificaciones nuevas.
- Enviar mensajes personalizados a los maestros.

Por otro lado, el diseño también nos permite especificar interfases creando un prototipo del sistema, proporcionando detalles de implementación, de como se harán las abstracciones de los datos y de los procedimientos; el diseño no es otra cosa mas que refinar los objetos y clases definidos anteriormente en el análisis, con detalles adicionales. El diseño orientado a objetos contempla las siguientes etapas:

- Diseño de la componente del dominio del problema
- Diseño de la componente de interacción humana (interfaz)
- Diseño de la componente del manejo de tareas.



- Diseño de la componente de administración de datos.

**El Diseño de la componente del dominio del problema**, muestra los objetos, clases, estructuras y temas en una interrelación, de tal manera que se pueda apreciar la forma en que interactúan los diversos componentes del sistema.

**El Diseño de la componente de interacción humana**, nos permite identificar los objetos que intervendrán en nuestra interfaz del sistema, considerando la relación que exista entre los mismos.

**El Diseño de la Componente del Manejo de Tareas**, nos sirve para representar una descripción breve de las tareas o funciones que el sistema va a realizar, explicando en cada una de ellas los elementos que pueden intervenir.

**En el Diseño de la Componente de Administración de Datos**, se presenta y propone la descripción de los campos (nombre, descripción, tipo, longitud, etc.) con que contará la Base de Datos, además del diseño de la misma Base de Datos.

## 7.1 Diseño de la Componente del Dominio del Problema

En esta componente se agrupan todas las clases específicas al dominio del problema. En esta componente se representa de una manera más clara la forma en como están integradas todas las clases y objetos contenidos en los temas y la forma que éstos se relacionan entre sí. (ver fig. 7.1.1)

## 7.2 Diseño de la Componente de Interacción Humana.

Para el diseño del estilo de ventanas del sistema se utiliza el API o toolkit de JAVA para poder manejar etiquetas y listas de textos que son controlados por los applets de JAVA y muestran un control de datos en cada evento del sistema. (ver fig. 7.2.1)

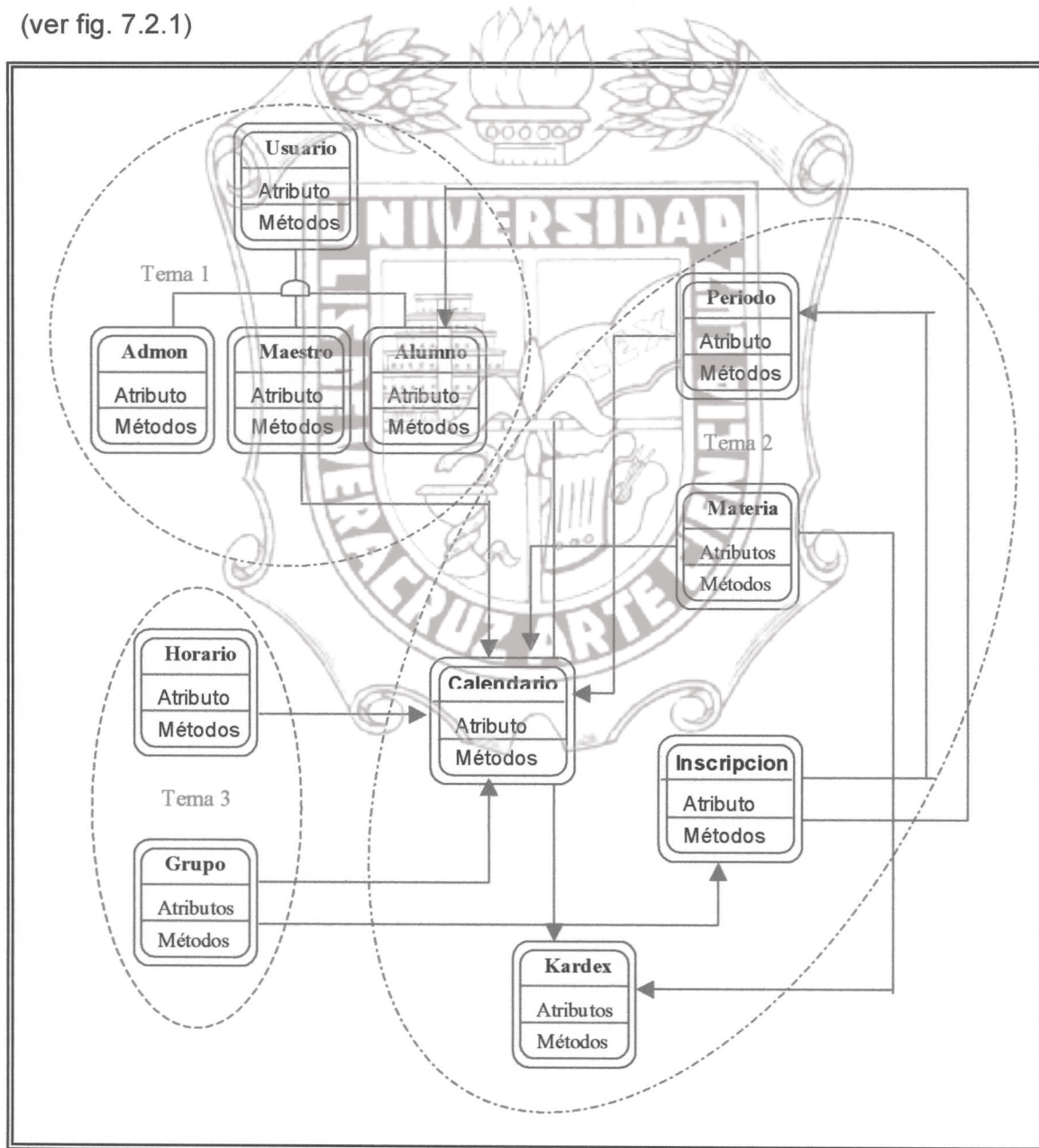


Figura No 7.1.1 Componente del dominio del problema

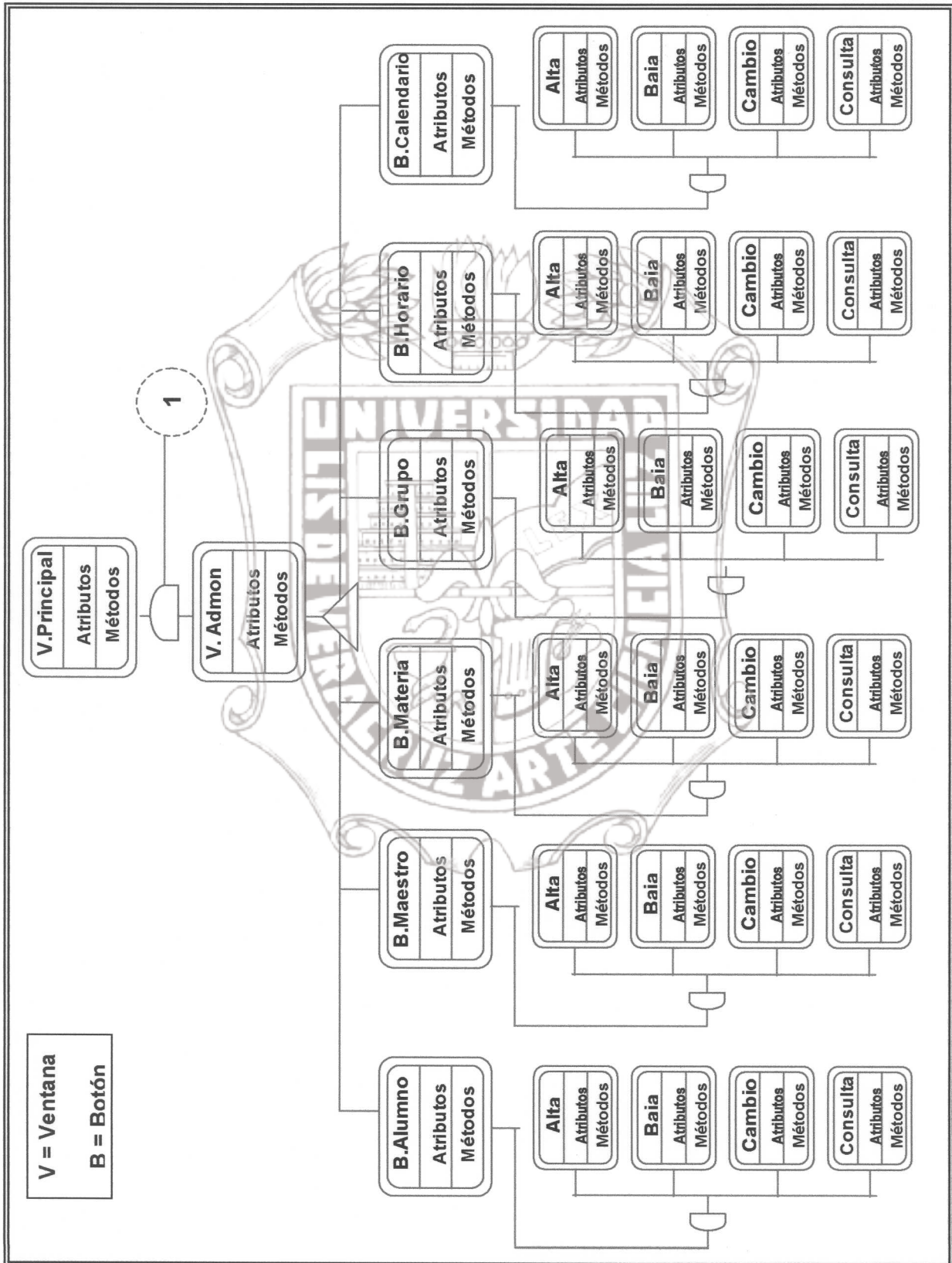


Figura N° 7.2.1 Componente de Interacción Humana



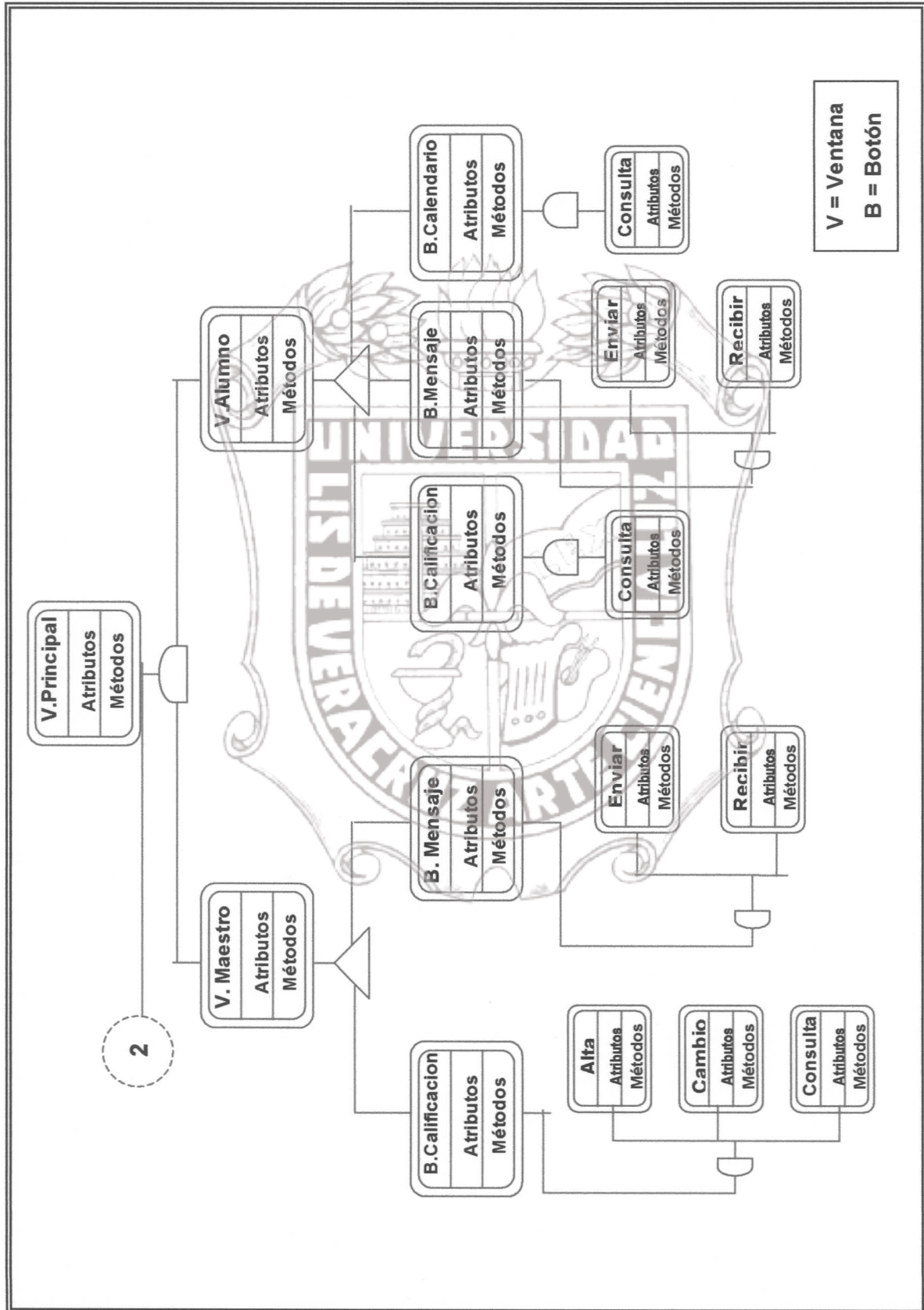


Figura N° 7.2.1 Componente de Interacción Humana (continuación)

## 7.3 Diseño de la Componente del Manejo de Tareas

En esta etapa del diseño, se describen algunos de los servicios que se emplearán en el sistema, los cuales se presentan por módulos, nos indica qué es lo que hace cada uno de ellos, la prioridad que tienen, los servicios de los que hace uso así como la comunicación que tienen con otros objetos.

En esta sección presentaremos a los objetos Administrador, Maestro y Alumno, que son los objetos base del SICADRC.

### ADMINISTRADOR

SERVICIO	DESCRIPCIÓN	PSEUDO CÓDIGO
Alta Alumno ()	Agrega un alumno a la Base de datos.	Se busca al alumno en la tabla en caso de no existir se da de alta y es cambio
Alta Maestro ()	Agrega un maestro a la Base de datos.	Se busca al maestro en la tabla en caso de no existir se da de alta y es cambio
Alta Materia ()	Agrega una materia a la Base de datos.	Se busca la materia en la tabla en caso de no existir se da de alta y es cambio
Alta Horario ()	Agrega un nuevo periodo de horas a la Base de datos.	Se busca el horario en la tabla en caso de no existir se da de alta y es cambio
Alta Calendario ()	Agrega un nuevo calendario de clases a la Base de datos.	Se busca el calendario en la tabla en caso de no existir se da de alta y es cambio
Alta Grupo ()	Agrega un nuevo grupo a la Base de datos.	Se busca el grupo en la tabla en caso de no existir se da de alta y es cambio
Baja Alumno ()	Elimina alumnos de la Base de datos.	Si existe el alumno en la tabla, se borra el registro

Tabla 7.3.1 Manejo de tareas del Objeto Administrador



SERVICIO	DESCRIPCIÓN	PSEUDO CÓDIGO
Baja Maestro ()	Elimina maestros de la Base de datos.	Si existe el maestro en la tabla, se borra el registro, sino no se realiza ninguna operación
Baja Materia ()	Elimina materias de la Base de datos.	Si existe la materia en la tabla, se borra el registro, sino no se realiza ninguna operación
Baja Horario ()	Elimina periodos de horas de la Base de datos.	Si existe el horario en la tabla, se borra el registro, sino no se realiza ninguna operación
Baja Calendario ()	Elimina un calendario de clases a la Base de datos.	Si existe el calendario en la tabla, se borra el registro, sino no se realiza ninguna operación
Baja Grupo ()	Elimina grupos de la Base de datos.	Si existe el grupo en la tabla, se borra el registro, sino no se realiza ninguna operación
Cambia Alumno ()	Modifica los datos de un alumno existente en la Base de datos.	Si existe el alumno en la tabla, se modifica el registro, sino existe no se hace nada
Cambia Maestro ()	Modifica los datos de un maestro existente en la Base de datos.	Si existe el maestro en la tabla, se modifica el registro, sino existe no se hace nada
Cambia Materia ()	Modifica los datos de una materia existente en la Base de datos.	Si existe la materia en la tabla, se modifica el registro, sino existe no se hace nada
Cambia Horario ()	Modifica los datos de un periodo de horas existente en la Base de datos.	Si existe el horario en la tabla, se modifica el registro, sino existe no se hace nada
Cambia Calendario()	Modifica los datos de un calendario de clases existente en la Base de datos.	Si existe el calendario tabla, se modifica el registro, sino existe no se hace nada

Tabla 7.3.1 Manejo de tareas del Objeto Administrador (continuación)



SERVICIO	DESCRIPCIÓN	PSEUDO CÓDIGO
Cambia Grupo ()	Modifica los datos de un grupo existente en la Base de datos.	Si existe el grupo en la tabla, se modifica el registro, sino existe no se hace nada
Consulta Alumno ()	Busca los datos de un alumno que exista en la Base de datos.	Si existe el alumno en la tabla, se despliega el contenido del registro, sino no hace nada
Consulta Maestro ()	Busca los datos de un maestro que exista en la Base de datos.	Si existe el maestro en la tabla, se despliega el contenido del registro, sino no hace nada
Consulta Materia ()	Busca los datos de una materia que exista en la Base de datos.	Si existe la materia en la tabla, se despliega el contenido del registro, sino no hace nada
Consulta Horario ()	Busca los datos de un período de horas que exista en la Base de datos.	Si existe el horario en la tabla, se despliega el contenido del registro, sino no hace nada
Consulta Calendario ()	Busca los datos de un calendario de clases que exista en la Base de datos.	Si existe el calendario en la tabla, se despliega el contenido del registro, sino no hace nada
Consulta Grupo ()	Busca los datos de un grupo que exista en la Base de datos.	Si existe el grupo en la tabla se despliega el contenido del registro, sino no hace nada
Realiza Inscripción ()	Agrega los datos del alumno al concentrado de inscripción.	Si existe el alumno en la tabla agrega el registro en tabla inscripción.

Tabla 7.3.1 Manejo de tareas del Objeto Administrador (continuación)

## MAESTRO

SERVICIO	DESCRIPCIÓN	PSEUDO CÓDIGO
Alta Calificación ()	Agrega las calificaciones al Kardex del alumno.	Si existe el alumno en la tabla kardex, se agrega la calificación, sino no se hace nada.
Cambia Calificación ()	Modifica las calificaciones existentes en el Kardex del alumno.	Si existe el alumno en la tabla kardex, se cambia la calificación, sino no se hace nada.
Consulta Calificación ()	Busca las calificaciones en el Kardex del alumno.	Si existe el alumno en la tabla kardex, se despliegan las calificaciones.
Envía Mensaje ()	Permite remitir mensajes a sus alumnos.	Si existe el grupo al que desea enviar el mensaje, se realiza el envío, sino no.
Recibe Mensaje ()	Permite recibir mensajes de sus alumnos	Si existe el maestro en la tabla y tiene algún mensaje le manda un aviso

Tabla No. 7.3.2 Manejo de Tareas del Objeto Maestro.

## ALUMNO

SERVICIO	DESCRIPCIÓN	PSEUDO CÓDIGO
Consulta Calificación ()	Busca sus calificaciones en su Kardex correspondiente.	Si existe el alumno en la tabla, se despliegan las calificaciones, sino no se hace nada.
Consulta Calendario ()	Busca su correspondiente calendario de clases existente en la Base de datos.	Si existe el calendario en la tabla, se despliega el contenido del registro, sino no se hace nada.
Consulta Kardex ()	Revisa los datos almacenados en su Kardex correspondiente.	Si existe el alumno en la tabla kardex, se despliegan el contenido del registro, sino no se hace nada.
Envía Mensaje ()	Permite remitir mensajes a sus maestros.	Si existe el maestro al que desea enviar el mensaje, lo envía.
Recibe Mensaje ()	Permite recibir mensajes de sus maestros.	Si existe el alumno en la tabla y tiene algún mensaje le manda un aviso

Tabla No 7.3.3 Manejo de Tareas del Objeto Alumno.

## 7.4 Diseño de la Componente de Administración de Datos

El diseño de datos es una de las actividades más importantes del diseño ya que el impacto de la estructura de datos en el diseño del programa y la complejidad procedimental hace que el diseñado de los datos tenga una influencia enorme en la calidad del software.

La principal actividad del diseño de datos es seleccionar representaciones lógicas de objetos de datos (estructuras de datos) identificadas durante la fase de definición y especificación de requerimientos.

Instituto de Ingeniería  
Universidad Veracruzana



El Diseño de datos comprende las siguientes fases:

- ◆ Definición de tablas
- ◆ Modelado entidad relación de la base de datos
- ◆ Diccionario de datos

## 7.5 Definición de tablas.

Nombre de la tabla	Descripción
ADMINISTRADOR	Almacenará todos los valores que corresponden a los atributos del objeto administrador.

Tabla No 7.4.1 Administrador

Nombre de la tabla	Descripción
ALUMNO	Almacenará todos los valores que corresponden a los atributos del objeto alumno.

Tabla No 7.4.2 Alumno

Nombre de la tabla	Descripción
CALENDARIO	Almacenará la relación entre los campos llave de las tablas alumno, periodo y grupo.

Tabla No 7.4.3 Calendario

Nombre de la tabla	Descripción
KARDEX	Almacenará la relación entre los campos llave de las tablas alumno y materia

Tabla No 7.4.4 Kardex

Nombre de la tabla	Descripción
GRUPO	Almacenará todos los valores que corresponden a los atributos del objeto grupo.

Tabla No 7.4.5 Grupo

Nombre de la tabla	Descripción
HORARIO	Almacenará todos los valores que corresponden a los atributos del objeto horario.

Tabla No 7.4.6 Horario

Nombre de la tabla	Descripción
MAESTRO	Almacenará todos los valores que corresponden a los atributos del objeto maestro.

Tabla No 7.4.7 Maestro

Nombre de la tabla	Descripción
MATERIA	Almacenará todos los valores que corresponden a los atributos del objeto materia.

Tabla No 7.4.8 Materia

Nombre de la tabla	Descripción
PERIODO	Almacenará todos los valores que corresponden a los atributos del objeto periodo.

Tabla No 7.4.9 Periodo

Nombre de la tabla	Descripción
ADMINISTRADOR	Almacenará todos los valores que corresponden a los atributos del objeto administrador.

Usuario	Contraseña	Nombre
Campo Llave		

Tabla No 7.4.10 Atributos de la tabla Administrador

Nombre de la tabla	Descripción
ALUMNO	Almacenará todos los valores que corresponden a los atributos del objeto alumno.

Usuario	Clave	Nombre	Dirección	Teléfono	Estado
Campo Llave					

Tabla No 7.4.11 Atributos de la tabla Alumno

Nombre de la tabla	Descripción
CALENDARIO	Almacenará la relación entre los campos llave de las tablas alumno, periodo y grupo.

Horario	Materia	Maestro	Grupo	Periodo
Campo Llave	Campo Llave	Campo Llave	Campo Llave	Campo Llave

Tabla No 7.4.12 Atributos de la tabla Calendario



Nombre de la tabla		Descripción	
KARDEX		Almacenará la relación entre los campos llave de las tablas alumno y materia	

Usuario	Materia	Calificación	Carácter
Campo Llave	Campo Llave		

Tabla No 7.4.13 Atributos de la tabla Kardex

Nombre de la tabla		Descripción	
GRUPO		Almacenará todos los valores que corresponden a los atributos del objeto grupo.	

Número	ID	Descripción	Nivel
Campo Llave			

Tabla No 7.4.14 Atributos de la tabla Grupo

Nombre de la tabla		Descripción	
HORARIO		Almacenará todos los valores que corresponden a los atributos del objeto horario.	

Número	Inicio	Final	Día
Campo Llave			

Tabla No 7.4.15 Atributos de la tabla Horario

Instituto de Ingeniería  
Universidad Veracruzana

Nombre de la tabla	Descripción
INSCRIPCIÓN	Almacenará la relación entre los campos llave de las tablas grupo periodo, alumno..

Usuario	Periodo	Grupo	Fecha
Campo Llave	Campo Llave	Campo Llave	

Tabla No 7.4.16 Atributos de la tabla Incripción

Nombre de la tabla	Descripción
MAESTRO	Almacenará todos los valores que corresponden a los atributos del objeto maestro.

Usuario	Clave	Nombre	Dirección	Teléfono	Mensaje
Campo Llave					

Tabla No 7.4.17 Atributos de la tabla Maestro

Nombre de la tabla	Descripción
MATERIA	Almacenará todos los valores que corresponden a los atributos del objeto materia.

Número	Nombre	ID	Serie
Campo Llave			

Tabla No 7.4.18 Atributos de la tabla Materia

Nombre de la tabla	Descripción
PERIODO	Almacenará todos los valores que corresponden a los atributos del objeto periodo.

Número	Inicio	Final
Campo Llave		

Tabla No 7.4.19 Atributos de la tabla Periodo

## 7.5.1 Atributos de los Objetos de la Base De Datos

En esta etapa del diseño, la metodología de **Coad & Yourdon** propone la descripción de los atributos de cada objeto, así como también se sugiere que se presente el diseño de la base de datos (relacionales) en donde se encuentran los objetos y dentro de la misma metodología, se propone el empleo de las reglas de normalización.

La base de datos se encuentra en un servidor con Windows 2000, creada con el manejador de base de datos Access 2000, este es un manejador de bases de datos relacional a la cual nos conectamos desde IDServer, por medio del ODBC. A continuación se presentan los atributos para el manejo de la información del SICADRC.



## Plantilla de atributos del objeto Alumno

SISTEMA DE CONTROL DE ALUMNOS DEL DRC					
Objeto:		Descripción:			
ALUMNO		Se representa la persona, la cual va a cursar las materias impartidas en el Diplomado en Redes de Computadoras			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Clave	Clave única que sirve para acceder al sistema	Alfa – Numérico	Máximo 10	SI	ivonne8231
Usuario	Cuenta de identificación del alumno	Alfa – Numérico	Máximo 10	SI	Cesar8611
Nombre	Nombre completo del alumno	Alfa – Numérico	Máximo 40	SI	Fernández del Campo Rabatte Leonel
Dirección	Residencia del alumno	Alfa – Numérico	Máximo 30	SI	Gildardo Gómez # 42B Animas
Teléfono	Número Telefónico del alumno	Numérico	Máximo 10	SI	8-13-54-34
Estado	0-En lista  1-Inscrito	Numérico	1	SI	2

Tabla No 7.5.1.1 Plantilla de atributos del objeto Alumno

## Plantilla de atributos del objeto Calendario

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS					
<b>Objeto:</b> CALENDARIO		<b>Descripción:</b> Se representan las actividades establecidas en el plan de estudios.			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Horario	Son las horas en las cuales el alumno realiza alguna actividad. Estas horas son establecidas por el Diplomado.	Alfa Numérico	-----	SI	10:00 AM – 12:00 PM
Materia	Clave de una asignatura impartida en el Diplomado en Redes de Computadoras	Numérico	-----	SI	05
Maestro	Clave de la persona que imparte alguna asignatura en el Diplomado en Redes de Computadoras	Numérico	-----	SI	02
Grupo	Clave del lugar donde se imparten las asignaturas	Numérico	-----	SI	801

Tabla No. 7.5.1.2 Atributos del objeto Calendario

## Plantilla de atributos del objeto kardex

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS						
Objeto:		Descripción:				
KARDEX		Representa el documento que se le asigna a cada alumno, el cual contiene todas sus calificaciones				
Atributo	Descripción	Tipo	Rango	Longitud	Obligatorio	Ejemplo
Usuario	Cuenta que identifica al alumno	Alfa Numérico	—	Máximo 10	SI	Cesar861
Materia	Clave de la asignatura	Numérico	—	—	SI	015
Calificación	Representación numérica de su aprovechamiento	Numérico	0 - 10	—	SI	10
Carácter	Tipo de examen Primera oportunidad	Numérico	0 - 1	—	SI	1

Tabla No. 7.5.1.3 Atributos del objeto Kardex



## Plantilla de atributos del objeto Grupo

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS					
<b>Objeto:</b> GRUPO	<b>Descripción:</b> Se almacena la información referente al conjunto de personas que toman las mismas asignaturas.				
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Número	Campo índice de un grupo	Numérico		SI	011
ID	Campo de identificación única de un grupo	Alfa Numérico	- 3	SI	101 ABC
Descripción	Identificación del turno, puede ser matutino, vespertino o sabatino.	Alfa Numérico	30	SI	Matutino

Tabla No. 7.5.1.4 Atributos del objeto Grupo

Instituto de Ingeniería  
Universidad Veracruzana

## Plantilla de atributos del objeto Horario

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS					
<b>Objeto:</b>		<b>Descripción:</b>			
HORARIO		Se almacena el rango de horas que pueda durar una actividad.			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Número	Campo índice de un horario	Numérico	-----	SI	03
Inicio	Hora de inicio de la actividad	Tiempo	00:00	SI	7:00
Final	Hora en que termina la actividad	Tiempo	00:00	SI	9:00
Día	Día en que se realiza la actividad	Carácter	15	SI	Miércoles

Tabla No. 7.5.1.5 Atributos del objeto Horario

Instituto de Ingeniería  
Universidad Veracruzana

## Plantilla de atributos del objeto Inscripción

<b>SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS</b>					
<b>Objeto:</b>		<b>Descripción:</b>			
INSCRIPCIÓN		Es la que lleva el control del grupo en que el alumno se inscribió.			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Usuario	Cuenta de identificación del usuario	Alfa Numérico	Máximo 10	SI	leone18913
Periodo	Campo índice del periodo correspondiente	Numérico	-----	SI	02
Grupo	Campo índice del grupo	Numérico	-----	SI	01
Fecha	Fecha en la que se realizó la inscripción	Fecha	dd/mm/aa	SI	10/01/02

Tabla No. 7.5.1.6 Atributos del objeto Inscripción

Instituto de Ingeniería  
Universidad Veracruzana



## Plantilla de atributos del objeto Materia

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS					
<b>Objeto:</b> MATERIA		<b>Descripción:</b> Se almacena toda la información referente a cada una de las asignaturas que se imparten en el Diplomado en Redes de Computadoras.			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Número	Campo llave de la asignatura	N Numérico	-----	SI	02 03
Nombre	Nombre completo de la asignatura	Alfa - Numérico	25	SI	Redes I Redes II
Id	Identificador de las materias seriadas	Alfa - Numérico	10	SI	03 03
Serie	Prioridad en las materias seriadas	N Numérico	3		1 2

Tabla No 7.5.1.7 Atributos del objeto Materia

Instituto de Ingeniería  
Universidad Veracruzana

## Plantilla de atributos del objeto Maestro

<b>SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS</b>					
<b>Objeto:</b>		<b>Descripción:</b>			
MAESTRO		Se representa la persona la cual va a impartir las materias en el Diplomado en Redes de Computadoras.			
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Clave	Clave única que sirve para acceder al sistema	Alfa - Numérico	Máximo 10	SI	Ivonne1
Usuario	Cuenta de identificación del maestro	Alfa - Numérico	Máximo 10	SI	Ivonne8321
Nombre	Nombre completo del maestro	Alfa - Numérico	Máximo 40	SI	Rabatte Suárez Ivonne
Dirección	Residencia del maestro	Alfa-Numérico	Máximo 30	SI	Gildardo Gómez # 42B Animas
Teléfono	Número Telefónico del maestro	Numérico	Máximo 10	SI	8-24-09-63
Mensaje	Pizarrón virtual de mensajes para los alumnos	Alfa - Numérico	100	NO	Mañana no hay clases

Tabla No. 7.5.1.8 Atributos del objeto Maestro

Instituto de Ingeniería  
 Universidad Veracruzana

## Plantilla de atributos del objeto Periodo

SISTEMA DE CONTROL DE ALUMNOS DEL DIPLOMADO EN REDES DE COMPUTADORAS					
<b>Objeto:</b>	<b>Descripción:</b>				
PERIODO	Se almacena el tiempo de duración de cada uno de los periodos del Diplomado				
Atributo	Descripción	Tipo	Longitud	Obligatorio	Ejemplo
Número	Campo índice del periodo	N Numérico	-----	SI	015
Inicio	Fecha de inicio del periodo	Fecha	dd/mm/aa	SI	10/01/02
Fin	Fecha de término del periodo	Fecha	dd/mm/aa	SI	10/01/02

Tabla No. 7.5.1.9 Atributos del objeto Periodo

## 7.6 Diagrama Entidad-Relación

El modelo E-R es otro punto importante en esta etapa del diseño. La anotación principal de la modelización de datos es el diagrama Entidad-Relación (E-R). El principal propósito del diagrama E-R es representar los objetos de datos y sus relaciones.



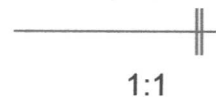
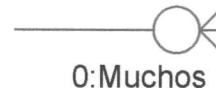
Se presenta el modelo E-R para las tablas identificadas en el sistema de control de alumnos del Diplomado en Redes de Computadoras, el cual se elaboró tomando como base del análisis del sistema, la definición de objetos y atributos así como cada una de las instancias de los mismos. Es importante recalcar que el diseño de las tablas se realizará en el manejador de bases de datos Access 2000.

La notación de diagrama Entidad Relación E-R es relativamente sencilla. Los objetos de datos se representan como rectángulos etiquetados, las relaciones se indican mediante rombos, las conexiones entre los objetos de datos y las relaciones se establecen mediante varias líneas especiales de conexión.

Representación gráfica de la notación:



Conexiones:



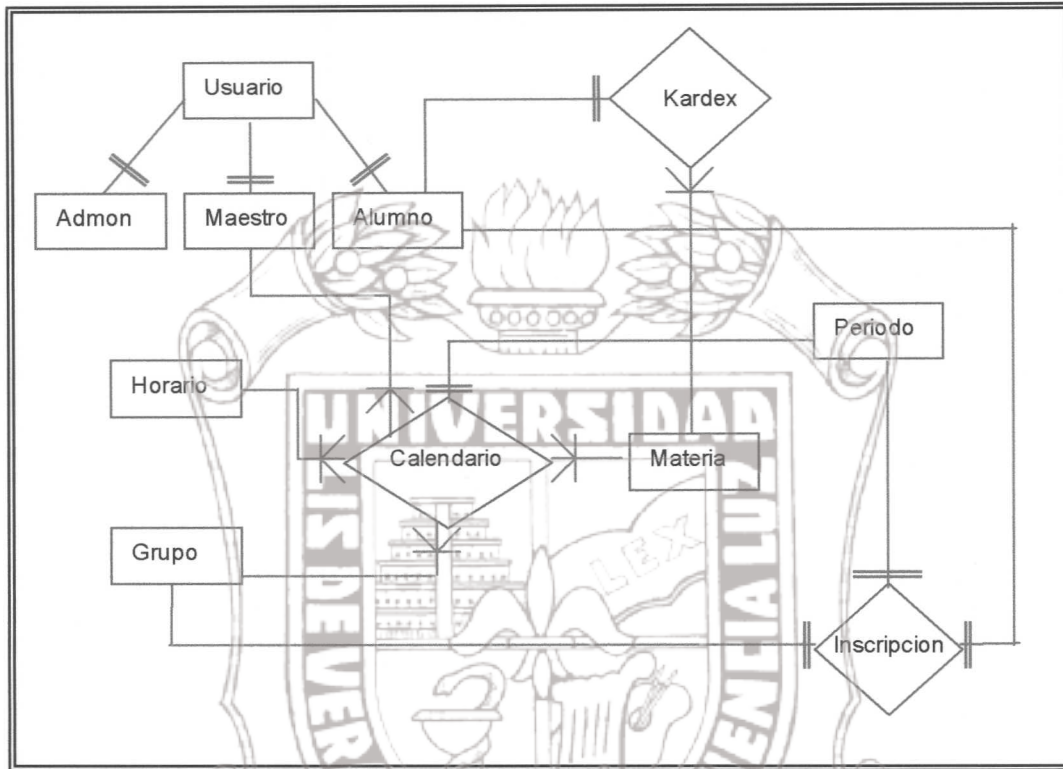


Figura No 7.6.1 Modelo E-R

## 7.7 Diccionario de Datos

Atributo	Descripción	Tipo
Clave	Clave única que sirve para acceder al sistema	Alfa - Numérico
Usuario	Cuenta de identificación del alumno	Alfa - Numérico
Nombre	Nombre completo del alumno	Alfa - Numérico

Tabla No 7.7.1 Diccionario de Datos

Atributo	Descripción	Tipo
Dirección	Residencia del alumno	Alfa - Numérico
Teléfono	Número Telefónico del alumno	Numérico
Estado	0-En lista 1-Inscrito	Numérico
Horario	Son las horas en las cuales el alumno toma alguna actividad. Estas horas son establecidas por el Diplomado.	Numérico
Materia	Clave de una asignatura impartida en el Diplomado en Redes de Computadoras	Numérico
Maestro	Clave de la persona que imparte alguna asignatura en el Diplomado en Redes de Computadoras	Numérico
Grupo	Clave del lugar donde se imparten las asignaturas	Numérico
Usuario	Cuenta que identifica al alumno	Alfa – Numérico
Materia	Clave de la asignatura	Numérico
Calificación	Representación numérica de su aprovechamiento	Numérico
Carácter	Tipo de examen 0 –Primera oportunidad 1 – Segunda oportunidad.	Numérico

Tabla No 7.7.1 Diccionario de Datos (Continuación)

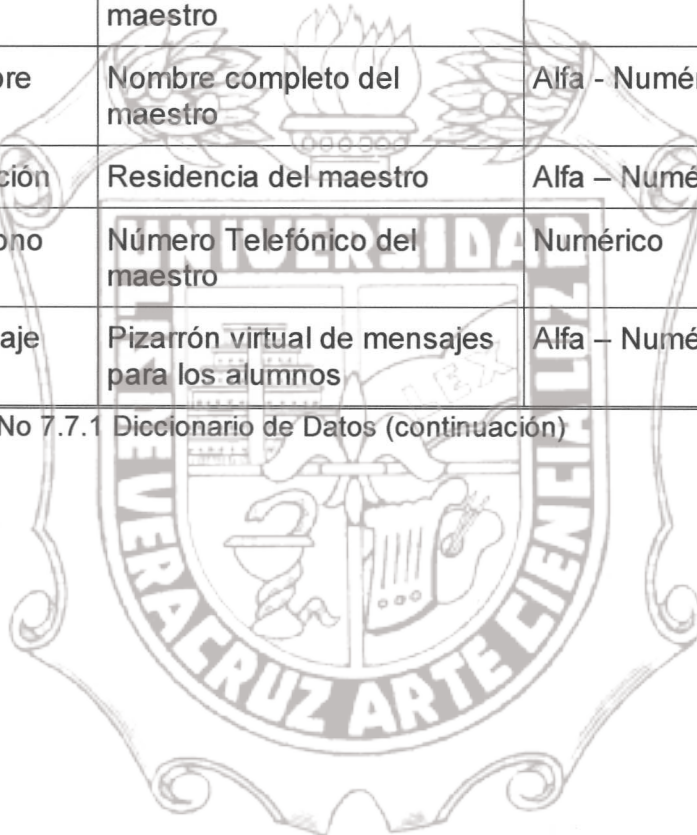


Atributo	Descripción	Tipo
Número	Campo índice de un grupo	Numérico
ID	Campo de identificación única de un grupo	Alfa - Numérico
Descripción	Leyenda de identificación. Puede ser matutino, vespertino o sabatino.	Alfa - Numérico
Número	Campo índice de un horario	Numérico
Inicio	Hora de inicio de la actividad	Tiempo
Final	Hora en que termina la actividad	Tiempo
Día	Día en que se realiza la actividad	Carácter
Usuario	Cuenta de identificación del usuario	Alfa – Numérico
Periodo	Campo índice del periodo correspondiente	Numérico
Grupo	Campo índice del grupo	Numérico
Fecha	Fecha en la que se realizó la inscripción	Fecha
Número	Campo llave de la asignatura	Numérico
Nombre	Nombre completo de la asignatura	Alfa – Numérico
Id	Identificador de las materias seriadas	Alfa – Numérico
Serie	Prioridad en las materias seriadas	Numérico

Tabla No 7.7.1 Diccionario de Datos (continuación)

Atributo	Descripción	Tipo
Clave	Clave única que sirve para acceder al sistema	Alfa - Numérico
Usuario	Cuenta de identificación del maestro	Alfa - Numérico
Nombre	Nombre completo del maestro	Alfa - Numérico
Dirección	Residencia del maestro	Alfa - Numérico
Teléfono	Número Telefónico del maestro	Numérico
Mensaje	Pizarrón virtual de mensajes para los alumnos	Alfa - Numérico

Tabla No 7.7.1 Diccionario de Datos (continuación)



## 8. DESARROLLO

Tomando como base los 35 servicios que llevan a cabo los objetos del SICADRC mostrados en el apartado 6 del análisis, se dio lugar a 45 unidades o bloques de código en lenguaje JAVA, los cuáles se presentan como anexos. JAVA es un lenguaje de programación orientado a objetos que ha sido diseñado para crear ambientes distribuidos y heterogéneos en Internet. La portabilidad de las aplicaciones constituye una de las grandes ventajas de JAVA sobre otros lenguajes de programación, adicionalmente hace posible la creación de aplicaciones basadas en Internet y en WWW; incorpora el manejo de cadenas, gráficos, componentes de interfaces gráficas con el usuario, manejo de excepciones, multihilos, multimedia (audio, imágenes, animación y video), procesamiento de archivos, trabajo con redes cliente / servidor.

El motivo primordial por el cual se seleccionó JAVA como lenguaje de programación para el SICADRC, es porque utiliza el concepto de Applets, los cuales son programas en JAVA que se encuentran en una página web (incrustado en el código HTML) y son ejecutados por un navegador web del lado del cliente, además de que JAVA es software libre, así que no es necesario obtener una licencia para poder utilizarlo y es posible sin costo alguno contar con la versión mas reciente y la documentación del mismo para su manipulación vía Internet.

A pesar de que JAVA es un lenguaje muy completo, nos encontramos con la desventaja de que no tiene un manejador de bases de datos, razón por la cual fue necesario buscar un servidor que nos permitiera utilizar algún tipo de manejador de bases de datos dentro de programas escritos en JAVA. Este servidor fue el IDSServer, el cual permite conectarnos a una base de datos diseñada en Access 2000. La opción de utilizar el IDSServer (que es una aplicación shareware) fué porque no existe ningún servidor de base de datos



gratuito que tenga el acceso a la tecnología JDBC (Java Data Base Conectividad), por lo tanto, se buscó el mas barato y el mas sencillo para nuestro sistema.

Cabe hacer mención que el servidor IDSserver sólo se utilizó para las pruebas del software, pero habiendo cumplido con los requisitos por parte del usuario, otra etapa es la implantación en un servidor de bases de datos propio de la FEI.

Acces 2000 es un gestor de bases de datos relacionales, esto significa que puede crear bases de datos que se puedan definir como el conjunto de todas las herramientas u objetos necesarios para poder almacenar y gestionar información. La elección de adoptar Access 2000 como manejador de la base de datos de SICADRC obedeció a tres razones:

1. Access 2000 es un manejador de bases de datos estándar y de fácil acceso.
2. Facilita enormemente el trabajo en Internet, ya que permite la utilización de las páginas de acceso de datos para que así si varios usuarios quieren utilizar la base de datos y se encuentran físicamente separados, puedan mediante este tipo de páginas gestionar en una única base de datos al mismo tiempo.
3. Es un manejador de base de datos conocido y reconocido mundialmente, además de que cuenta con una interfaz amigable al usuario, ya que permite una fácil operación del mismo.

Cabe señalar, que SICADRC, fue implementado en una PC pero su arquitectura fue diseñada para ejecutarse en una red, todo dependerá de las condiciones y las limitaciones con las que cuente la administración del Diplomado en Redes de Computadoras de Computadoras de la FEI.

## 8.1 Recursos para el desarrollo del proyecto

En las tablas siguientes, se muestra la información referente a cada uno de los recursos que se utilizarán para el desarrollo del proyecto.

La tabla No 8.1.1 corresponde al Recurso Humano, es decir, se muestra el personal que estará involucrado en el desarrollo del proyecto, así como una breve descripción de las funciones que cada uno de ellos llevará a cabo

Puesto	Descripción
Desarrollador	Persona que realizará las funciones de administrador del proyecto, analista, diseñador y programador.
Asesor	Persona que se encargará de supervisar y revisar el trabajo realizado por el desarrollador.

Tabla No 8.1.1 Recursos Humanos

La tabla No 8.1.2 corresponde a los recursos de Hardware que serán útiles a lo largo del desarrollo del sistema.

Equipo	Características
Computadora	Pentium 4 256 MB en memoria RAM 75 GB en Disco Duro Unidad de CD – ROM
Impresora	Modelo COMPAQ IJ600. Inyección de Tinta.

Tabla No 8.1.2 Recursos de Hardware

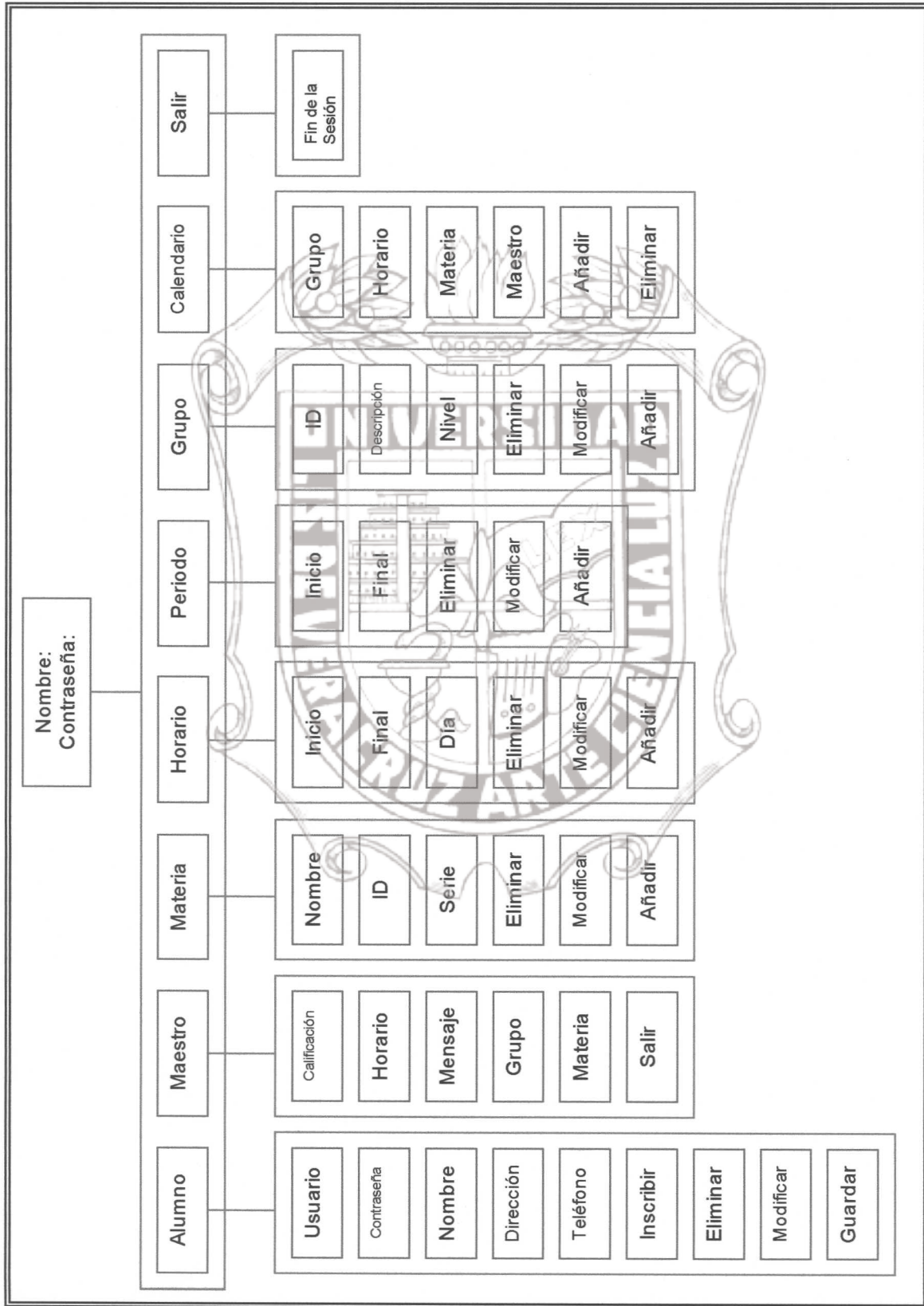
La tabla No 8.1.3 corresponde a los recursos de software involucrados en el desarrollo del proyecto, ya sea como herramienta de desarrollo o como herramienta auxiliar para el desarrollador en sus diferentes actividades.

Nombre	Descripción
Windows 2000	Sistema Operativo Base
Access 2000	Generador de Bases de Datos
IDSServer	Manejador de JDBC – ODBC
JAVA Versión 1.3.1	Lenguaje de programación orientado a objetos.
FreeJava	Editor de texto para escribir programas en lenguaje JAVA
WinZip	Software compresor y descompresor de archivos.

Tabla No 8.1.3 Recursos de Software.

Se especifica que la dirección IP con la que trabaja el sistema varía, dependiendo donde se encuentra funcionando, es decir si la máquina donde se encuentra instalado el sistema tiene como dominio 'www.dominio.com', entonces el sistema se ubicará en esa dirección IP. A su vez en esta tesis se trabaja con la dirección 'localhost', que es la dirección predeterminada para referirse a la computadora en uso.





8.2 Diagrama Jerárquico de Árbol

## 9. IMPLANTACIÓN.

El día 17 de Agosto del 2001 se realizó la prueba de implantación del sistema en la máquina del Jefe del Centro de Cómputo de la FEI; se realizaron pruebas para verificar que el software cumplía con los requisitos esperados por los usuarios finales.

El sistema fue aprobado en todas sus funciones y actividades que desempeña, sin tener ningún problema, error o modificación, teniendo en cuenta la administración del diplomado que deberá pagar la licencia para poder utilizar el servidor IDServer o implantarlo en su propio servidor de base de datos.

## 10. VERIFICACIÓN Y VALIDACIÓN

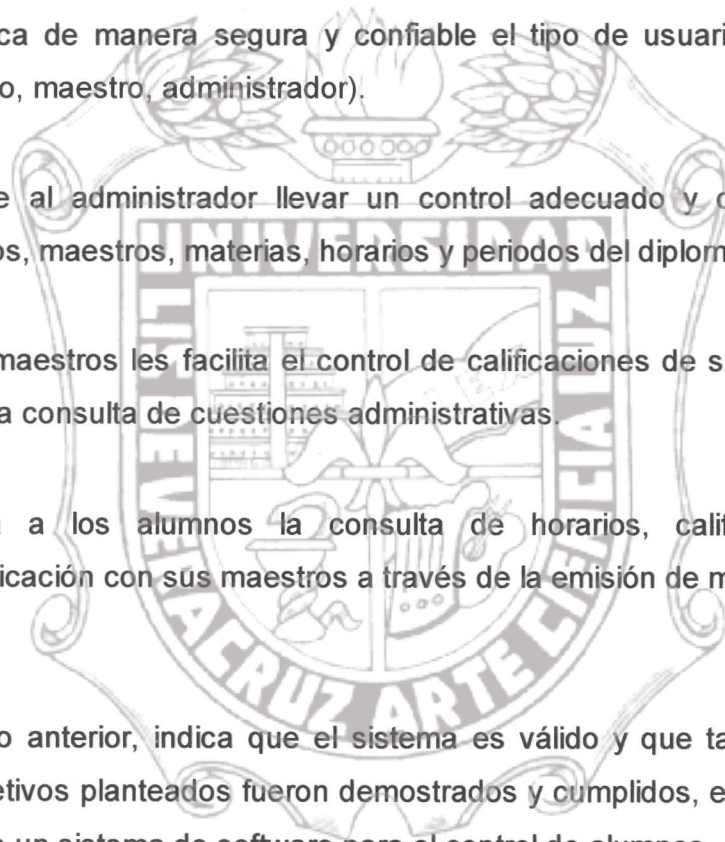
El proceso de verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica y el proceso de validación se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente. La importancia de estos procesos radica en que son sinónimos de calidad del software [DEU79].

Después de finalizar la implementación de SICADRC, pudimos darnos cuenta que la elección del lenguaje de programación y la herramienta para el diseño de la base de datos, se realizó de manera adecuada, ya que nos permitieron llevar a cabo las actividades que se deseaban en SICADRC con éxito, entre las que se encuentran el manejo de agentes, la actualización y consulta de la base de datos vía Internet, etc. Una vez que pusimos en la práctica SICADRC, se pudo constatar que realmente cubría las necesidades y los objetivos para los que fue creado. Algunas de las verificaciones que se realizaron en SICADRC fueron:

El sistema monoagente a través del cual se implementó, realmente actúa como secretario virtual, puesto que ayuda al usuario en su utilización, ya que lo guía en su trabajo, así como también lo mantiene informado de los trámites académicos y administrativos que suceden en el Diplomado en Redes de Computadoras.

- Identifica de manera segura y confiable el tipo de usuario de SICADRC (alumno, maestro, administrador).
- Permite al administrador llevar un control adecuado y oportuno de los alumnos, maestros, materias, horarios y periodos del diplomado.
- A los maestros les facilita el control de calificaciones de sus alumnos, así como la consulta de cuestiones administrativas.
- Facilita a los alumnos la consulta de horarios, calificaciones y la comunicación con sus maestros a través de la emisión de mensajes.

Todo lo anterior, indica que el sistema es válido y que tanto la hipótesis como los objetivos planteados fueron demostrados y cumplidos, en términos de la integración de un sistema de software para el control de alumnos.





## 11. PRUEBAS Y EVALUACIÓN DEL SISTEMA

El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea o imperfecta. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad [PRE97].

La prueba del software es un elemento crítico para la garantía de calidad de software y representa una revisión final de las especificaciones, del diseño y la codificación [DEU79].

La filosofía detrás de las pruebas es la de encontrar errores. Los casos de prueba se diseñan con este propósito en mente. Un caso de prueba es un conjunto de datos que el sistema procesará como una entrada normal. Sin embargo, los datos se crean con la intención expresa de determinar si el sistema los procesará correctamente [KEN96].

### 11.1 Pruebas de Unidad.

Las pruebas de unidad son las que lleva a cabo el desarrollador, antes de la integración de la unidad en un sistema más grande. Se utiliza el análisis y diseño detallado como guía. Una unidad de programa es lo suficientemente pequeña como para ser probada minuciosamente.

Existen cuatro tipos de pruebas que se deben efectuar a una unidad de programa, las cuáles son:

- Pruebas Funcionales
- Pruebas de Desempeño
- Pruebas de Caja blanca
- Pruebas de Caja Negra

Para las pruebas de unidad, de acuerdo a cada subprueba (funcionales, de tensión y caja blanca) se procederá a ejecutar el código con datos de entrada para los cuales se conocen los resultados esperados, además de valores límites (valores mínimos, máximos y valores fuera de los límites funcionales).

### 11.1.1 Pruebas Funcionales

Dentro de este tipo de prueba se ubica la prueba de interfaz, el propósito principal de estas pruebas es verificar la operación de las interfaces entre módulos. Se deben probar tanto las interfaces de control como las de condiciones de límite, se ejecutan los caminos independientemente y finalmente se prueban los caminos del manejo de errores.

Caso de Prueba	Entrada	Respuesta Esperada	Respuesta Obtenida	Observaciones
Consultar datos de alumnos	Clic en la opción Alumno y Tabla de Alumnos	Lista de datos de los alumnos	Lista de datos de los alumnos	Permite visualizar los registros de la tabla alumnos
Alta de un alumno	Clic en la opción alumno y guardar. Datos del alumno	Acepta los datos y los almacena en la tabla alumno	Acepta los datos y los almacena en la tabla alumno	El resultado obtenido es el esperado.
Cambio de la cuenta de usuario de un alumno	Clic en la opción alumno y modificar la nueva cuenta de usuario.	No permita el cambio de cuenta de usuario.	No realiza el cambio, sino que genera otro alumno	El resultado obtenido es el esperado, ya que no puede alterarse su cuenta de usuario.
Baja de un alumno	Clic en la opción alumno y eliminar	Borrar la información de la tabla alumno	Borrar la información de la tabla alumno	El resultado obtenido fue el esperado
Emisión de Mensajes del alumno al maestro.	Clic en la opción mensaje, nombre del maestro y enviar mensaje	Enviar el mensaje al correspondiente maestro.	Al momento de que el maestro acceda al sistema visualiza el mensaje que le fue enviado.	El resultado obtenido fue el esperado.
Consulta de calificaciones	Clic en la opción boleta y tabla Kardex	Desplegar las calificaciones del alumno	Visualizó las calificaciones en cada una de las materias que cursa el alumno	El resultado obtenido es el correcto.

Tabla No 11.1.1 Pruebas Funcionales

Instituto de Ingeniería  
Universidad Veracruzana



## 11.1.2 Pruebas de Desempeño

Estas pruebas se emplean en varias partes de la unidad para determinar el tiempo de ejecución y la eficiencia del programa, así como el tiempo de respuesta. Las pruebas de desempeño son muy productivas en los niveles de subsistema y sistema.

Caso de Prueba	Tiempo de Ejecución	Eficiencia del programa	Observaciones
Claves de usuario	3 segundos	Buena	Se ejecuta al momento de terminar de teclear las claves, y en caso de no existir envía los mensajes correspondientes
Consulta de alumnos	5 segundos	Buena	Considerando que se tiene aproximadamente una tabla de 100 alumnos
Eliminar maestros	2 segundos	Buena	La eliminación se realiza de manera rápida.
Actualizaciones de alumnos	2 segundos	Buena	Las actualizaciones se efectúan de forma rápida y eficiente.
Limpiar pantalla principal	15 segundos	Buena	SICADRC no permite tener varias ventanas abiertas y considerando que cuenta con imágenes y variedad de colores, es aceptable la cantidad de tiempo utilizada.
Alta de un maestro	1 segundo	Buena	Dado que son pocos los datos que se van a almacenar es aceptable el tiempo empleado.

Tabla No 11.1.2 Pruebas de Desempeño.

### 11.1.3 Prueba de Caja Blanca

La prueba de caja blanca es un método que se basa en un minucioso examen de los detalles procedimentales; a través de este tipo de prueba es posible obtener casos de prueba que nos garanticen que se están ejercitando por lo menos una vez todos los caminos independientes, que se ejecuten los bucles en sus valores límites y que se utilicen las estructuras internas de datos para verificar que hayan sido declaradas correctamente [PRE97]. Para esta prueba es necesario llevar a cabo los siguientes procesos:

- ◆ Elegir un servicio al cual se le va aplicar la prueba.
- ◆ Aplicar al servicio elegido un pequeño número de pruebas.

No. de prueba	Servicio	Descripción
1	Consulta Alumno	Verificar que muestre los datos correctos del Alumno.
2	Alta de Maestros	Verificar que se almacenen los datos correctos en la tabla maestro

Tabla No 11.1.3 Pruebas de Caja Blanca.

Para cada una de las pruebas arriba mencionadas se tienen uno o varios casos de prueba que se listan a continuación:

No. Prueba	No. Caso	Resultado Esperado	Resultado Obtenido
1	1	Muestra en la pantalla de consulta del alumno la siguiente información:  Usuario: erandi Clave: erandi Nombre: Rocio Erandi Barrientos Martínez Dirección: Blas Galindo # 6 Teléfono: 8-12-65-43 Grupo: 101 Periodo: 10/01/01 – 16/08/01	El Resultado Esperado
	2	Usuario: vidi Contraseña: vidi Nombre: Isidro Vicencio Delgadillo Dirección: Priv. Francisco Sarabia # 112 Teléfono: 8-15-40-52 Grupo: 101 Periodo: 10/01/01 – 16/08/01	
2	1	Datos de Entrada:  Usuario: azamolah Clave: azamolah Nombre: Abigail Hernández Zamora Dirección: Betancourt # 95 Teléfono: 8-13-54-34	El Resultado Esperado

Tabla No 11.1.3 Pruebas de Caja Blanca (continuación).



## 11.1.4 Prueba de Caja Negra.

Dentro de las pruebas de caja negra, las pruebas de clases tratan de probar las operaciones encapsuladas que contiene cada clase y el comportamiento de cada una de éstas. Para ello, se debe verificar que cada uno de los servicios de los objetos permita las entradas que deben ser las correctas para producir un resultado correcto [PRE97]. La siguiente tabla muestra todos los servicios que se definieron.

No. Prueba	Servicio	Descripción
<b>SERVICIOS DEL OBJETO ADMINISTRADOR</b>		
1	Alta Alumno ( )	Verificar que guarde en la tabla alumno los datos proporcionados.
2	Alta Maestro ( )	Verificar que guarde en la tabla maestro los datos proporcionados.
3	Alta Materia ( )	Verificar que guarde en la tabla materia los datos proporcionados.
4	Alta Horario ( )	Verificar que guarde en la tabla horario los datos proporcionados.
5	Alta Calendario ( )	Verificar que guarde en la tabla calendario los datos proporcionados.
6	Alta Grupo ( )	Verificar que guarde en la tabla grupo los datos proporcionados.
7	Baja Alumno ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla alumno.

Tabla No 11.1.4 Pruebas de Caja Negra

Instituto de Ingeniería  
Universidad Veracruzana

No. Prueba	Servicio	Descripción
<b>SERVICIOS DEL OBJETO ADMINISTRADOR</b>		
8	Baja Maestro ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla maestro.
9	Baja Materia ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla materia.
10	Baja Horario ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla horario.
11	Baja Calendario ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla calendario.
12	Baja Grupo ( )	Verificar que elimine el registro que corresponda a los datos proporcionados de la tabla grupo.
13	Cambia Alumno ( )	Verificar que guarde en la tabla alumno los nuevos datos proporcionados.
14	Cambia Maestro ( )	Verificar que guarde en la tabla maestro los nuevos datos proporcionados.
15	Cambia Materia ( )	Verificar que guarde en la tabla materia los nuevos datos proporcionados.
16	Cambia Horario ( )	Verificar que guarde en la tabla horario los nuevos datos proporcionados.
17	Cambia Calendario ( )	Verificar que guarde en la tabla calendario los nuevos datos proporcionados.
18	Cambia Grupo ( )	Verificar que guarde en la tabla grupo los nuevos datos proporcionados.
19	Consulta Alumno ( )	Verificar que asigne los valores de la instancia alumno, a las variables correspondientes.

Tabla No 11.1.4 Pruebas de Caja Negra (continuación)



No. Prueba	Servicio	Descripción
<b>SERVICIOS DEL OBJETO ADMINISTRADOR</b>		
20	Consulta Maestro ( )	Verificar que asigne los valores de la instancia maestro, a las variables correspondientes.
21	Consulta Materia ( )	Verificar que asigne los valores de la instancia materia, a las variables correspondientes.
22	Consulta Horario ( )	Verificar que asigne los valores de la instancia horario, a las variables correspondientes.
23	Consulta Calendario ( )	Verificar que asigne los valores de la instancia calendario, a las variables correspondientes.
24	Consulta Grupo ( )	Verificar que asigne los valores de la instancia grupo, a las variables correspondientes.
25	Realiza Inscripción ( )	Verifica que se cambie el estado de la instancia alumno y se guarde en la tabla.
<b>SERVICIOS DEL OBJETO MAESTRO</b>		
26	Alta Calificación ( )	Verificar que guarde en la tabla Kardex los datos proporcionados del alumno correspondiente.
27	Cambia Calificación ( )	Verificar que guarde en la tabla Kardex los nuevos datos proporcionados del alumno correspondiente
28	Consulta Calificación ( )	Verificar que asigne los valores de la instancia Kardex, a las variables correspondientes.
29	Consulta Horario ( )	Verifica que asigne los valores de la instancia calendario, a las variables correspondientes.

Tabla No 11.1.4 Pruebas de Caja Negra (continuación)



No. Prueba	Servicio	Descripción
<b>SERVICIOS DEL OBJETO ALUMNO</b>		
30	Consulta Calificación ( )	Verificar que asigne los valores de la instancia Kardex, a las variables correspondientes.
31	Consulta Calendario ( )	Verificar que asigne los valores de la instancia calendario, a las variables correspondientes.

Tabla No 11.1.4 Pruebas de Caja Negra (continuación)

Para algunas de las pruebas arriba mencionadas se tiene un caso de prueba para su verificación, los cuales se enlistan en la siguiente tabla:

No. Prueba	Entrada	Resultado Esperado	Resultado Obtenido
3	Nombre: Linux Id: 3 Serie: 2	Agregar registro en la tabla materia	Se creó el registro
5	Horario: Lunes 7:00 – 9:00 Materia: TCP/IP Maestro: Abigail Hernández Zamora Grupo: 101 Periodo: 10/01/01 – 16/08/01	Agregar el registro en la tabla calendario	Se agregó el registro en la tabla.

Tabla No 11.1.4 Pruebas de Caja Negra (continuación).

 Instituto de Ingeniería  
 Universidad Veracruzana

No. Prueba	Entrada	Resultado Esperado	Resultado Obtenido
29	<p>Si el maestro que esta consultando es Abigail Hernández Zamora, muestra en pantalla:</p> <p>Grupo: 101</p> <p>Horario: Lunes 7:00 – 9:00</p> <p>Materia: TCP/IP</p> <p>Grupo: 102</p> <p>Horario: Jueves 7:00 – 9:00</p> <p>Materia: Windows NT.</p>	<p>Desplegar los datos correspondientes de la tabla calendario</p>	Datos Correctos.
30	<p>Si el alumno que esta consultando es Rocio Erandi Barrientos Martínez, muestra en pantalla:</p> <p>Materia: Linux</p> <p>Calificación: 5</p> <p>Carácter: 1ra. Oportunidad.</p> <p>Materia: Windows NT</p> <p>Calificación: 0</p> <p>Carácter: 1ra. Oportunidad.</p>	<p>Desplegar los datos correspondientes de la tabla calendario</p>	Datos Correctos

Tabla No 11.1.4 Pruebas de Caja Negra (continuación)

Instituto de Ingeniería  
Universidad Veracruzana

## 11.2 Pruebas de Validación.

Después de las pruebas de unidad que generalmente se basan en pruebas referentes a la programación para verificar la lógica de los programas y de los métodos de un sistema, es necesario hacer las pruebas de validación que consisten en que el sistema cumpla con las expectativas del cliente el cual tendrá que usar el sistema para poderlo calificar, estas pruebas se apoyan a través de un cuestionario que llenan los usuarios seleccionados al azar.

Uno de los aspectos más importantes de evaluación es la usabilidad ya que mediante esto se podrá saber que tan fácil y eficiente es nuestro sistema y que satisfacción dejará al usuario en cuanto a su uso.

Los atributos antes mencionados se podrán evaluar con el cuestionario que se le dará a cada uno de los usuarios y para ello se determinará una escala de evaluación con lo que obtendremos resultados para la evaluación del sistema.

Para la evaluación de aceptación de SICADRC se decidió llevar a cabo las pruebas alfa, que consisten en que el usuario pueda manipular el sistema con toda libertad pero en un entorno controlado y el desarrollador esté como un espectador, para ir tomando nota de los detalles que puedan surgir en ese momento y poder corregirlos posteriormente. Los usuarios dieron su opinión en cuanto a la interfaz presentada por el sistema, la cuál fue calificada como buena, ya que es fácil de utilizar y rápida de aprender.



## CUESTIONARIO

Conteste las siguientes preguntas asignando un valor entre 1 y 5 de acuerdo con su criterio.

- 1.- ¿Le agradan los colores que SICADRC presenta? [ ]
- 2.- ¿Tiene experiencia en computación? [ ]
- 3.- ¿Ha manejado sistemas parecidos a SICADRC? [ ]
- 4.- ¿La ayuda que le proporciona el agente es de apoyo suficiente? [ ]
- 5.- ¿Los datos de captura están bien distribuidos? [ ]
- 6.- ¿Los datos de salida están bien distribuidos? [ ]
- 7.- ¿Los mensajes que el agente envía son congruentes con la información que esta manejando? [ ]
- 8.- ¿El agente envía avisos cuando se presentan errores al ingresar algún dato equivocado? [ ]
- 9.- ¿SICADRC muestra indicadores de lo que realiza cada botón? [ ]
- 10.- ¿Es fácil de memorizar y aprender la interfaz? [ ]
- 11.- ¿SICADRC requiere de alguna otra aplicación para su operación? [ ]
- 12.- ¿Las opciones que maneja SICADRC son representativas de lo que el sistema realiza? [ ]
- 13.- ¿Cuándo se realizan cambios al ejecutar alguna función puede deshacerla? [ ]
- 14.- ¿Cómo considera el tiempo de respuesta en cada una de las operaciones de SICADRC? [ ]
- 15.- ¿Considera apropiadas las funciones que lleva a cabo el agente? [ ]

Escala de calificaciones para el cuestionario:

Concepto	Valor
Excelente	5
Muy bueno	4
Bueno	3
Regular	2
Malo	1

Tabla No 11.2.1 Escala de calificaciones

A continuación se muestra una tabla de datos con los niveles de los usuarios y su calificación con respecto al cuestionario.

No. Usuario	Nombre del Usuario	Tipo de Usuario	Puesto	Puntos
1	Genaro Rebolledo Méndez	Avanzado	Técnico Académico	53
2	Alfonso Marín Lozano	Avanzado	Catedrático de la Facultad de Administración de la UV.	62
3	Adolfo Ortiz Sedano	Avanzado	Jefe del Centro de Cómputo de la FEI	64
4	Ivonne Fernández del Campo Rabatte	Intermedio	Prestadora de Servicio Social	42
5	Javier González Saíz	Intermedio	Prestador de Servicio Social	47

Tabla No 11.2.2 Niveles de usuario

A continuación se describe las categorías de usuarios que se observan en el lugar donde se implantará SICADRC:

Nombre de la Categoría	Descripción
Novato	Persona que ha tenido poco o ningún contacto con un equipo de cómputo y que necesita asesoría para iniciar a trabajar con SICADRC.
Intermedio	Persona que ha tenido mas contacto con un equipo de cómputo y que no tiene mayores problemas para trabajar con SICADRC.
Avanzado	Persona que esta perfectamente relacionada con el funcionamiento y operación de un equipo de cómputo, es totalmente independiente y no requiere asesoría alguna para operar SICADRC.

Tabla No 11.2.3 Categoría de usuarios.

Para la elaboración de la siguiente tabla, se tomaron en cuenta las calificaciones proporcionadas por los usuarios a los cuáles se les aplicó, para poder evaluar el nivel de aprendizaje de los usuarios en el manejo y uso de SICADRC.



Pregunta	No. de Usuario				
	1	2	3	4	5
1.- ¿Le agradan los colores que SICADRC presenta?	4	5	5	4	5
2.- ¿Tiene experiencia en computación?	5	5	5	3	3
3.- ¿Ha manejado sistemas parecidos a SICADRC?	1	5	5	1	1
4.- ¿La ayuda que le proporciona el agente es de apoyo suficiente?	4	4	5	3	4
5.- ¿Los datos de captura están bien distribuidos?	4	4	5	3	4
6.- ¿Los datos de salida están bien distribuidos?	3	4	4	3	4
7.- ¿Los mensajes que el agente envía son congruentes con la información que maneja?	4	5	4	3	3
8.- ¿El agente envía avisos cuando se presentan errores al ingresar algún dato equivocado?	4	3	3	2	3
9.- ¿SICADRC muestra indicadores de lo que realiza cada botón?	3	4	5	2	3
10.- ¿Es fácil de memorizar y aprender la interfaz?	4	5	4	3	3
11.- ¿SICADRC requiere de alguna otra aplicación para su operación?	4	4	4	2	2
12.- ¿Las opciones que maneja SICADRC son representativas de lo que el sistema realiza?	3	5	5	3	2
13.- ¿Cuando se realizan cambios al ejecutar alguna función puede deshacerla?	2	2	5	2	2
14.- ¿Como considera el tiempo de respuesta en cada una de las operaciones de SICADRC?	4	4	4	4	4
15.- ¿Considera apropiadas las funciones que lleva a cabo el agente?	4	3	3	4	4
<b>Total</b>	<b>53</b>	<b>62</b>	<b>66</b>	<b>42</b>	<b>47</b>

Tabla No 11.2.4 Resultados del cuestionario.

Enseguida se muestra la escala de calificaciones y rango de valores de aceptación de SICADRC de acuerdo a los casos de prueba aplicados.

Descripción	Puntos	Rango
Excelente	5	75-65
Muy Bueno	4	64-55
Bueno	3	54-45
Regular	2	44-35
Malo	1	Menor a 34

Tabla No 11.2.5 Valores de Aceptación.

Los datos recopilados para cada uno de los usuarios, de acuerdo con los casos de prueba que se le aplicaron, nos ayudaron a determinar los niveles de los usuarios, además de que sirve como referencia para saber que tanto SICADRC es útil y eficiente, y a la vez cuál es la experiencia con las computadoras, para el caso de SICADRC, fue un software bastante aceptable.

### 11.3 PRUEBAS DEL SISTEMA

Las pruebas del Sistema son importantes ya que el software desarrollado es incorporado a otros elementos, en si es un elemento de un sistema mayor basado en computadora.

Se ejecuto de manera profunda cada modulo del SICADRC para verificar el comportamiento que tiene al interactuar con el Hardware y ver como reacciona ante situaciones imprevistas.

Para esto se realizaron tres tipos de prueba:

- **SEGURIDAD**
- **RECUPERACIÓN**
- **RENDIMIENTO**

## **SEGURIDAD**

SICADRC se encuentra instalado en un servidor y la base de datos que utiliza está creada en Acces 2000, que es una base de datos relacional, bajo la plataforma Windows 2000.

Dicho sistema cuenta con una opción de acceso y un modulo de usuarios, los cuales trabajan de tal manera que si alguna persona desea acceder al sistema, está debe haber sido dada de alta como usuario por el administrador, de otra forma no podrá trabajar con el sistema ya que este le pedirá el Password o Clave de Acceso.

## **Casos de Prueba**

C1:{ Usuario con acceso al sistema }

C2:{Usuario desconocido, sin acceso al sistema }

C3:{Usuario tratando de acceder a la Base de Datos por programas }



CASO DE PRUEBA	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	OBSERVACIONES
C1	Clave del sistema correcta (Administrador)	Acceso al sistema con opciones habilitadas de acuerdo al tipo de usuario	Correcto	
C2	Claves del sistema incorrectas	Acceso denegado	Correcto	
C3	Conectándose al servidor y a la Base de Datos	Acceso denegado	Correcto	El servidor y la Base de Datos cuentan con Password diferentes para cada usuario, y el único que las conoce es el Administrador

Tabla No 11.3.1 Pruebas de Seguridad

## RECUPERACIÓN

En este tipo de prueba se estudian los tipos de fallas producidas por causas ajenas al sistema, cabe hacer mención que SICADRC no cuenta con la opción de auto-recuperación de datos en caso de fallas no producidas por el propio sistema, ya sean ocasionadas intencionalmente o imprevistas.

### Casos de pruebas

C1: {Interrupción de energía eléctrica}

C2: {Apagado de Computadora ejecutando consultas}

C3: {Interrupción de un proceso (Altas, consultas , etc.) Intencionalmente}

CASO DE PRUEBA	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	OBSERVACIONES
C1	Interrupción de energía eléctrica	Registros en buen estado en la Base de Datos y sistema en situación normal	Base de Datos y sistema en buen estado	
C2	Desconectar la computadora	Base de Datos estable sin daños	Base de Datos y sistema en buen estado	
C3	Reinicio de la computadora	Acceso denegado	Base de Datos y sistema en buen estado	

Tabla No 11.3.2 Pruebas de Recuperación

Como el sistema no cuenta con auto-recuperación para cualquier tipo de falla, las Bases de Datos son revisadas directamente por el administrador para ver si sufrieron algún daño o si están marcadas como sospechosas, esto se hace directamente desde Acces 2000.

- Tiempo máximo en revisar la Base de Datos de 10 a 20 minutos
- En caso de pérdidas de datos, instalar de nuevo
- Verificar con el usuario la última operación que estaba realizando

## RENDIMIENTO.

La prueba de rendimiento viene a corroborar el buen funcionamiento del SICADRC, ya que se basó en el funcionamiento del software en tiempo de ejecución, dentro del contexto de un sistema integrado, ya que anteriormente se aseguró el rendimiento de los módulos individuales, mas sin embargo, hasta que no estén completamente integrados todos los módulos de SICADRC no se puede asegurar el buen rendimiento del mismo.

Los casos de prueba se realizaron en todos los módulos, principalmente en los módulos de Administrador, Maestro y Alumno.

## Casos de Prueba

C1: {consulta de calendario en el módulo administrador}

C2: {consulta de calificaciones en el módulo alumno}

C3: {consulta de horario en el módulo maestro}



CASO DE PRUEBA	ENTRADA	RESULTADO ESPERADO	RESULTADO OBTENIDO	OBSERVACIONES
C1	Consulta del calendario de clases en el módulo del administrador	Rendimiento Optimo	Rendimiento Optimo	Las consultas se hicieron de forma secuencial, ya que no se pueden realizar dos consultas al mismo tiempo.
C2	Consulta de calificaciones en el módulo de alumno	Rendimiento Optimo	Rendimiento Optimo	Las consultas se hicieron de forma secuencial, ya que no se pueden realizar dos consultas al mismo tiempo.
C3	Consulta del horario de clases en el módulo del maestro.	Rendimiento Optimo	Rendimiento Optimo	Las consultas se hicieron de forma secuencial, ya que no se pueden realizar dos consultas al mismo tiempo.

Tabla No 11.3.3 Pruebas de Rendimiento

Instituto de Ingeniería  
Universidad Veracruzana

## 12. MANTENIMIENTO.

El software indudablemente sufrirá cambios después de ser entregado al cliente, porque el mantenimiento del software cuenta con mas esfuerzo que cualquier actividad de ingeniería de software. La facilidad de mantenimiento es la facilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos. No hay forma de medir directamente la facilidad de mantenimiento; por consiguiente, se deben utilizar medidas indirectas. Una simple métrica orientada al tiempo es el tiempo medio de cambio, es decir, el tiempo que se tarda en analizar la petición de cambio, en diseñar una modificación adecuada, en implementar el cambio, en probarlo y en distribuir el cambio a todos los usuarios. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo.

Cuando se construye software de computadora, los cambios son inevitables y los cambios aumentan el grado de confusión entre los ingenieros del software que están trabajando en el proyecto. La confusión surge cuando no se han analizado los cambios antes de realizarlos, no se han registrado antes de implementarlos, no se han comunicado a las personas que necesitan saberlo o no se han controlado de manera que mejoren la calidad y reduzcan los errores.

Uno de los objetivos principales de la ingeniería del software es mejorar la facilidad con la que se pueden implantar los cambios y reducir la cantidad de esfuerzo necesario para implementarlos. [PRE97]

## 13. CONCLUSIONES

El Sistema de Control de Alumnos para el Diplomado en Redes de Computadoras de Computadoras (SICADRC) fue desarrollado con la finalidad de proporcionar al Diplomado en Redes de Computadoras que se imparte en la Facultad de Estadística e Informática un instrumento que le permitiera hacer frente a las actividades administrativas y de control de los estudiantes.

Durante el desarrollo de SICADRC, se aplicaron todas las etapas del modelo de ciclo de vida clásico y la metodología orientada a objetos. La aplicación de un modelo y una metodología siempre es muy útil ya que apearse disciplinadamente a un esquema de trabajo da certeza a las actividades que se realizan y, encausadas correctamente, nos permiten alcanzar el desarrollo de un software, si bien no perfecto, si con una disminución considerable en el número de errores, que pueden ser controlados y corregidos con mayor facilidad, porque ciertamente, el producto final del desarrollo de software no es un producto perfecto sino perfectible.

Lo primordial en el desarrollo de SICADRC fue el uso de un sistema monoagente, el cuál es una práctica que involucra conocimientos de distintas áreas de la informática como son: manejo y consulta de bases de datos, programación e ingeniería de software y representación del conocimiento. Dada la gran cantidad de ramas de la informática que se usan, fue importante analizar y diseñar este sistema.

La hipótesis planteada fue demostrada al observar las características que presenta el método actual de control escolar en el Diplomado en Redes de Computadoras de Computadoras y SICADRC. Dicha observación arrojó los siguientes resultados:



Método actual de control escolar	SICADRC
Requiere de 5 a 6 personas para llevar el control administrativo del diplomado.	Requiere de una persona para el control administrativo del diplomado
No tiene un control para el acceso a la información.	El acceso a la información se controla a través del login y password de los usuarios.
Tiempo de atención (basado en la experiencia 2 horas).	Tiempo de atención (planeado 15 minutos).
La disponibilidad de la información depende del personal del diplomado.	La información está disponible en el momento que se requiera.
La interacción maestro – alumnos esta determinada por la disponibilidad del maestro.	Facilita la interacción maestro – alumno.
La organización de los horarios y la asignación de los grupos en cada nuevo periodo es tardada.	Agiliza la elaboración de horarios y la asignación de grupos al inicio de cada periodo.
En ocasiones es posible que se extravié la información de periodos anteriores.	La información de todos los periodos es segura, ya que se encuentra almacenada en una base de datos.

En base a lo anterior, podemos establecer que SICADRC, efectivamente es de más fácil acceso que el método que se usa actualmente, además de que presenta mayores ventajas dentro del control escolar del Diplomado en Redes de Computadoras de Computadoras.

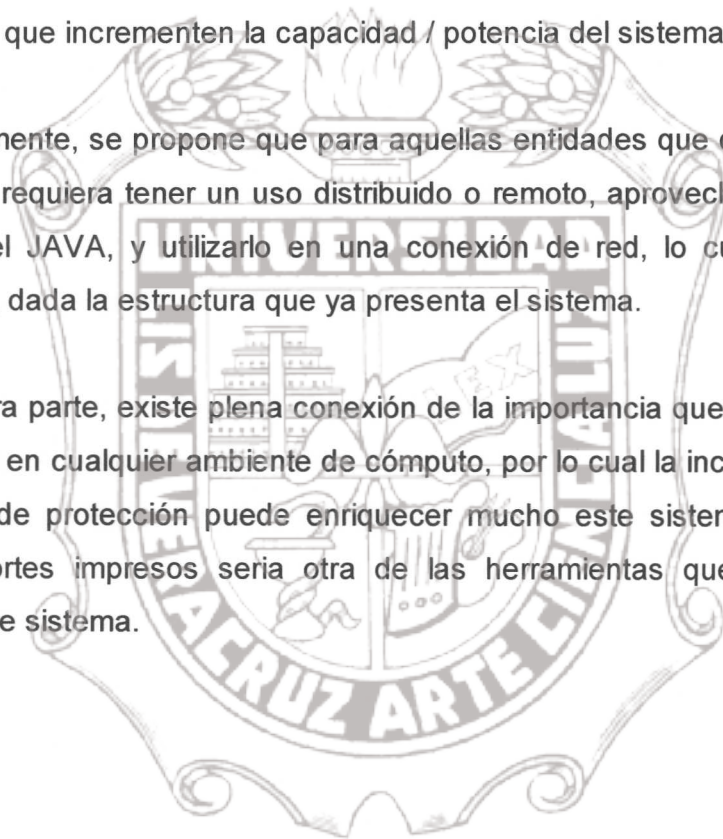
Instituto de Ingeniería  
Universidad Veracruzana

## 14. RECOMENDACIONES Y SUGERENCIAS

Las limitaciones que hasta el momento presenta, pueden ser corregidas a futuro, perfeccionando los modelos existentes e incluyendo nuevas opciones y herramientas que incrementen la capacidad / potencia del sistema.

Inicialmente, se propone que para aquellas entidades que deseen utilizar el SICADRC, y requiera tener un uso distribuido o remoto, aprovechar la plataforma que ofrece el JAVA, y utilizarlo en una conexión de red, lo cual es en forma transparente, dada la estructura que ya presenta el sistema.

Por otra parte, existe plena conexión de la importancia que tiene el aspecto de seguridad en cualquier ambiente de cómputo, por lo cual la inclusión de alguna herramienta de protección puede enriquecer mucho este sistema, así como el agregar reportes impresos sería otra de las herramientas que se le podrían agregar a este sistema.



## BIBLIOGRAFÍA

- [BOO96] Booch, Grady. Análisis y diseño Orientado a Objetos con Aplicaciones. Copublicación de Addison-Wesley Iberoamericana, S.A. y Ediciones Díaz de Santos. Segunda Edición 1996.
- [COAD91] Coad, P. and Edward Yourdon, Object Oriented Analysis. Prentice – Hall, Second Edition 1991.
- [DEU79] Deutsch, M., Verification and validation, in software Engineering. Editorial Prentice Hall. 1979
- [DGB00] Dirección General de Banner 2000 de la Universidad Veracruzana.
- [GER99] Germand Weiss (Edited by ) Multiagent Systems A. Modern Approach to Distributed Artificial Intelligence The MIT Press Cambridge, Massachusetts London, England ISBN 0-262-23203-0 1998.
- [JAI92] Jacobson, I., Christerson, M., Jonsson, P. & Overgaard, G. Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, 1992, ISBN: 0-201-54435-0.
- [JEN98] Jennings Nicolás, Sykara Katia and Wooldridge Michael. A Roadmap of Agent Research and Development Autonomous Agents and Multi-Agents Systems, 1, 7-38, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands, 1998.
- [KEN96] Kendall. Análisis y Diseño de Sistemas. Editorial Prentice-Hall, 1996



[KOR95] Korth, F. Henry. Silberschatz, Abraham. Fundamentos de bases de datos. Segunda Edición. Editorial Mc. Graw Hill. Mexico 1995.

[LEM00] Lemaitre Christian, Sistemas Multiagentes, Newsletter, Año 9, Vol. 33 y 34, Otoño-Invierno 2000.

[PRE97] Pressman, Roger. Ingeniería del Software. Editorial McGraw Hill, 1997

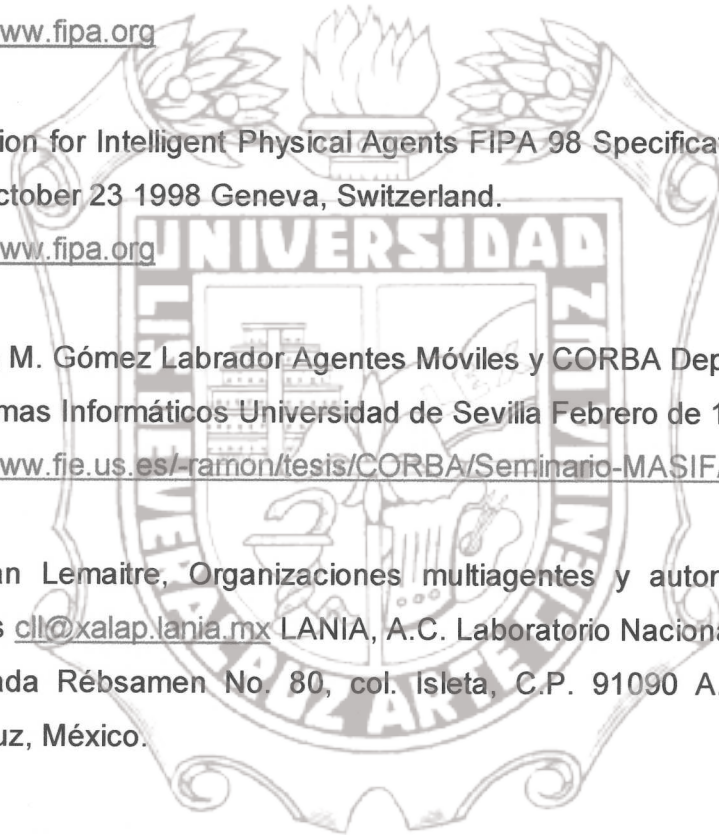
[RUE96] Rueda Rodríguez Sandra y Otros, Agentes y Sistemas multiagente. Integración de Conceptos Básicos Grupo de Investigación HIDRA Departamento de Ingeniería de Sistemas y Computación Universidad de los Andes. 1996

[SEN92] Senn A. James. Análisis y diseño de sistemas de Información. Segunda Edición. Editorial McGraw Hill. 1992.

## Referencias de Internet

- Daniel Le Berre, Universidad de Newscatle, Australia. Facultad de Economía y Comercio 25 de enero del 2001.  
<http://café.newcastle.edu.au/daniel/>
- Terri Nabeth. Información Tecnológica de Agentes Tecnológicos. El centro para aprendizaje avanzado en tecnologías.  
[http://www.insead.fr\(CALT/encyclopedia/computerSciences/agents](http://www.insead.fr(CALT/encyclopedia/computerSciences/agents)
- Tim Finnin y Yannis Labrou. UMBC Red Agentes. <http://agents.umbc.edu/>
- José M Vidal. Sistemas Multiagentes. 26 de febrero de 2001.  
<http://www.multiagent.com/news/index.html>

- Canal Europeo por excelencia para computación basado en Agentes. Agentes de Ligas. <http://www.agentlink.org/happenings/News.html>
- Foundation for Intelligent Physical Agents FIPA 97 Specifications <http://www.fipa.org>
- Foundation for Intelligent Physical Agents FIPA 98 Specifications Publication date October 23 1998 Geneva, Switzerland. <http://www.fipa.org>
- Ramón M. Gómez Labrador Agentes Móviles y CORBA Depto. De Lenguaje y Sistemas Informáticos Universidad de Sevilla Febrero de 1999. <http://www.fie.us.es/~ramon/tesis/CORBA/Seminario-MASIF/>
- Christian Lemaitre, Organizaciones multiagentes y automatizaciones de oficinas [cll@xalap.lania.mx](mailto:cll@xalap.lania.mx) LANIA, A.C. Laboratorio Nacional de Informática Avanzada Rébsamen No. 80, col. Isleta, C.P. 91090 A.P. 696 Xalapa, Veracruz, México.

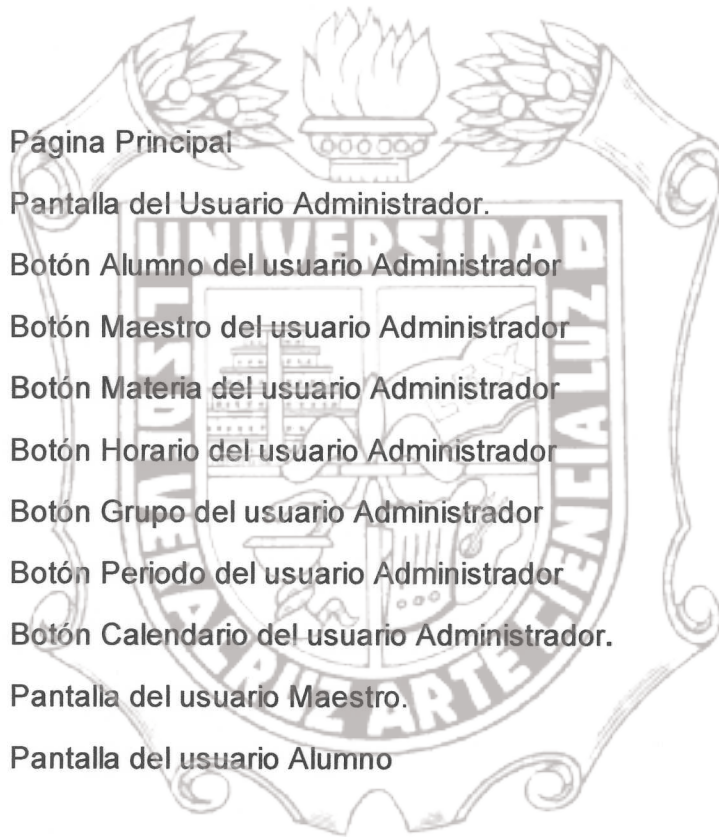






## INDICE DE FIGURAS DE ANEXOS

Número		Pág.
B.1	Página Principal	107
B.2	Pantalla del Usuario Administrador.	108
B.3	Botón Alumno del usuario Administrador	109
B.4	Botón Maestro del usuario Administrador	110
B.5	Botón Materia del usuario Administrador	111
B.6	Botón Horario del usuario Administrador	112
B.7	Botón Grupo del usuario Administrador	113
B.8	Botón Periodo del usuario Administrador	113
B.9	Botón Calendario del usuario Administrador.	114
B.10	Pantalla del usuario Maestro.	115
B.11	Pantalla del usuario Alumno	117



## A. Instrucciones de Instalación

El sistema SICADRC Versión 1.0 requiere para su instalación, un equipo de cómputo con las siguientes características:

- ◆ Plataforma Windows.
- ◆ Sistema operativo Windows 98 o superior.
- ◆ Procesador Pentium II o superior.
- ◆ Capacidad mínima libre en disco duro de 500 MB.
- ◆ Memoria RAM de 32 MB mínimo.
- ◆ Monitor SVGA en una resolución mínima de 800 x 600
- ◆ Navegador Explorer versión 5 en adelante.
- ◆ Navegador Netscape versión 4 en adelante.

Para poder ejecutar el programa SICADRC, se deben llevar a cabo los siguientes pasos:

Se introduce el CD de instalación y automáticamente aparecera el programa de instalación, el cual mostrara las opciones del tipo de instalación que se requiera.

Las opciones de instalación son:

- Instalación para Windows 95,98,Me.
- Instalación para Windows NT y 2000
- Instalación IDSserver
- Ayuda

Cuando se esta ejecutando la instalación, se muestran los siguientes pasos:

- Si muestra  se realizó correctamente la instalación
- Si muestra  se realizó incorrectamente la instalación

- Si al momento de introducir el CD no aparece el programa de instalación, se tendrá que ejecutar manualmente, dando clic en el archivo SICADRC que se encuentra dentro del CD.

### **Como empezar el llenado de información de SICADRC.**

1. Se empieza dando de alta un período desde su inicio y su final
2. Después se dan de alta los maestros que estarán dando curso en ese período, así como las materias, horarios, grupos y los alumnos solo se guardan en lista, en espera de su inscripción.
3. Ya que se dio de alta lo anterior se comienza a armar el calendario de clases.
4. Por último se inscriben los alumnos en el grupo que se les asigne.

### **B. Agentes de Interfaz.**

Estos agentes ponen énfasis en su autonomía y aprendizaje para realizar sus tareas. El caso más claro de este tipo de agentes es el de un asistente personal que colabora con su usuario en el mismo ambiente de trabajo. Los agentes de interfaz asisten y dan soporte al usuario para aprender el uso de una aplicación.

Los agentes de interfaz aprenden para mejorar su ayuda al usuario en cuatro formas:

1. Al observar e imitar al usuario.
2. Al recibir retroalimentación del usuario.
3. Al recibir instrucciones explícitas del usuario.
4. Al pedir consejo a otros agentes.

La colaboración con otros agentes (si es que existen) se limita a preguntar consejos y no a conseguir compromisos.



## C. Manual del Usuario

Para poder acceder a SICADRC Versión 1.0, debes seguir los pasos de instalación anteriormente mencionados. Una vez que se acceso a SICADRC aparecerá una pantalla (ver figura # B.1) de inicio del sistema, en la cual deberá introducir su cuenta de usuario y contraseña (si es que el coordinador le ha proporcionado alguna con anterioridad).

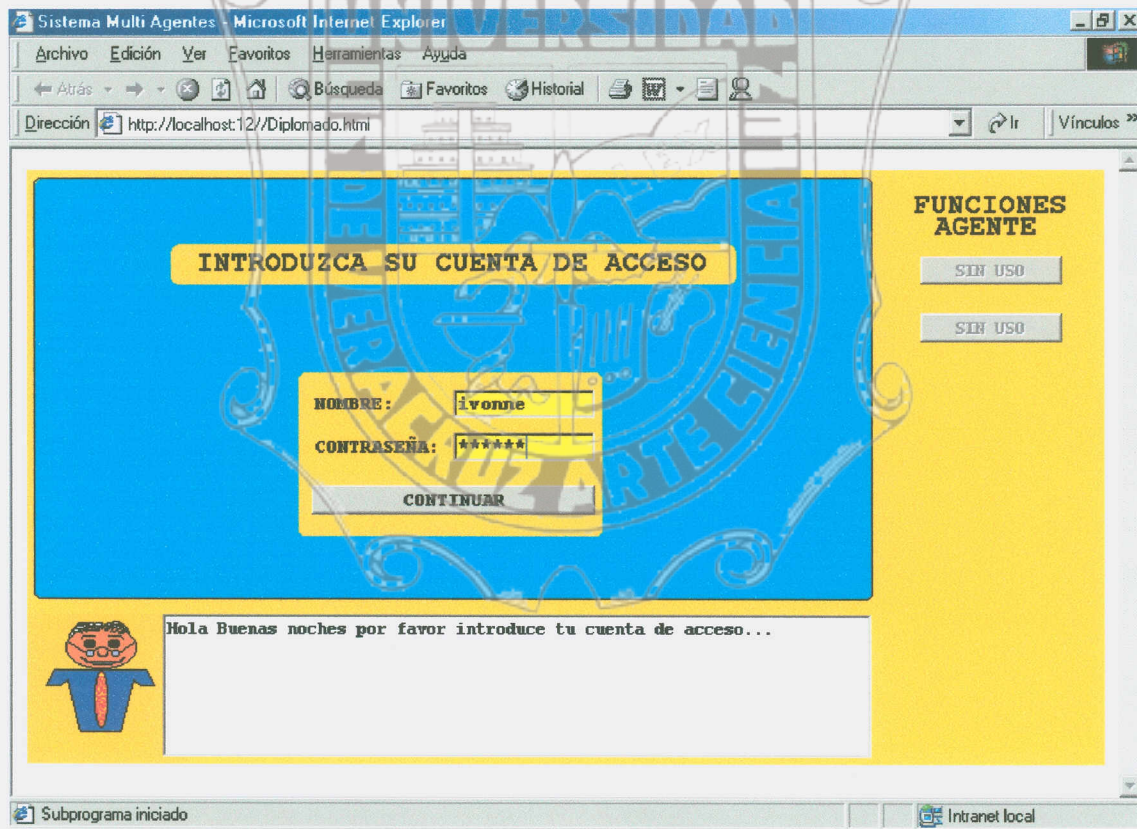


Figura No B.1 Pantalla Principal.

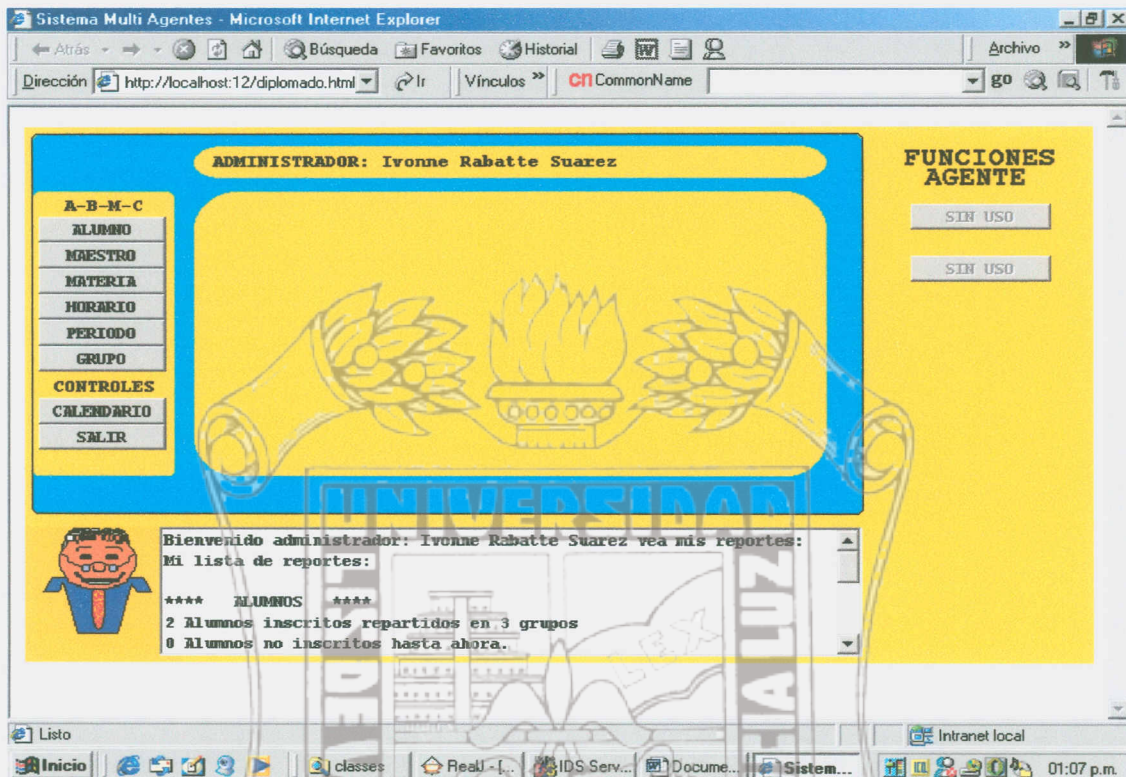


Figura No B.2 Pantalla del Usuario Administrador.

- ♦ **Alumno:** Este botón permitirá al administrador visualizar la lista de alumnos que hasta el momento se encuentran en el diplomado, así como también permitirá al momento de seleccionar alguno de los alumnos en lista, borrarlos al presionar el botón de eliminar o en su defecto cambiar el contenido de algunos de sus datos, para ello debe dar un clic en el botón de modificar, realizar los cambios y por último presionar el botón de guardar. También cuenta con la opción de dar de alta un nuevo alumno, para ello debe únicamente llenar la información que se le pide en la pantalla (ver figura # B.3) y al finalizar deberá presionar el botón de guardar.



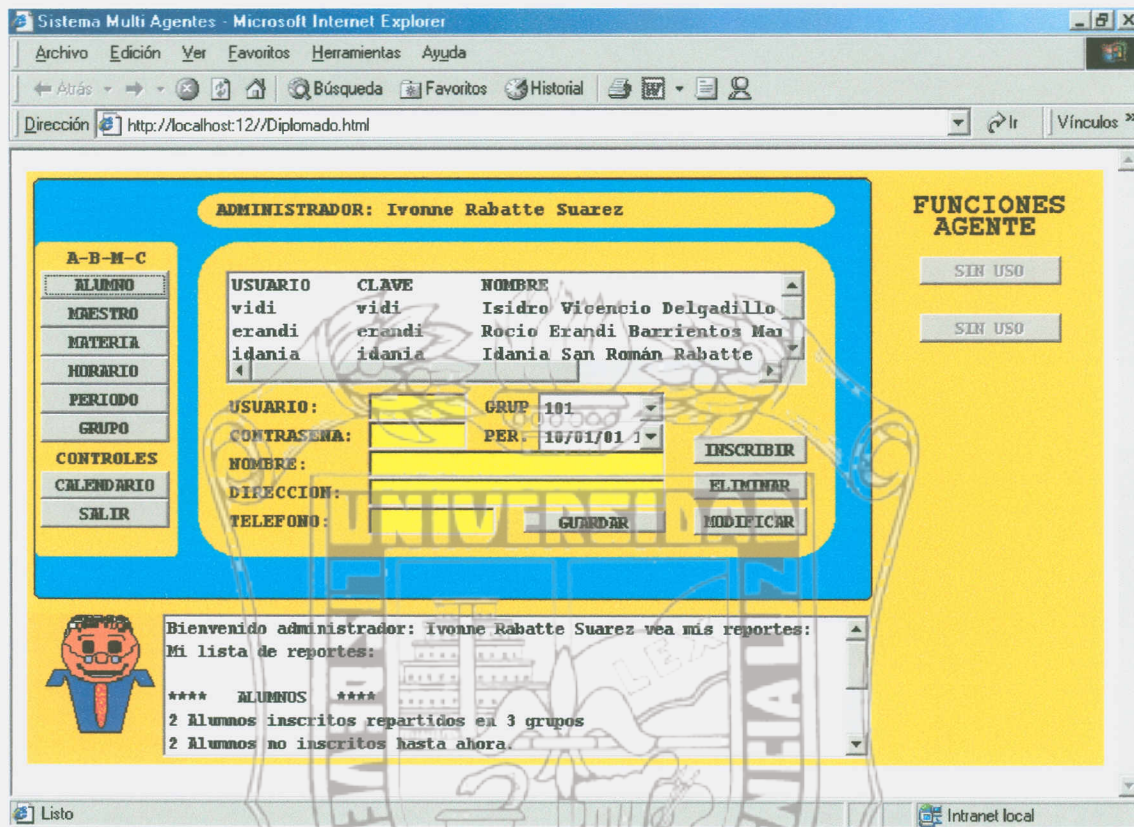


Figura No B.3 Botón Alumno del usuario Administrador

- ♦ **Maestro:** Al igual que el botón de alumno, permite al administrador dar de alta, modificar y eliminar los maestros que actualmente imparten alguna asignatura en el Diplomado en Redes de Computadoras. Para borrar de la lista algún maestro, deberá seleccionarlo de la lista y presionar el botón de eliminar. Para modificar el contenido del registro de algún maestro, deberá seleccionarlo, dar un clic en el botón de modificar, realizar los cambios y por último presionar el botón de guardar y para agregar un nuevo maestro únicamente debe llenar la información que se le pide en la pantalla (ver figura # B.4) y al finalizar presionar el botón de guardar.



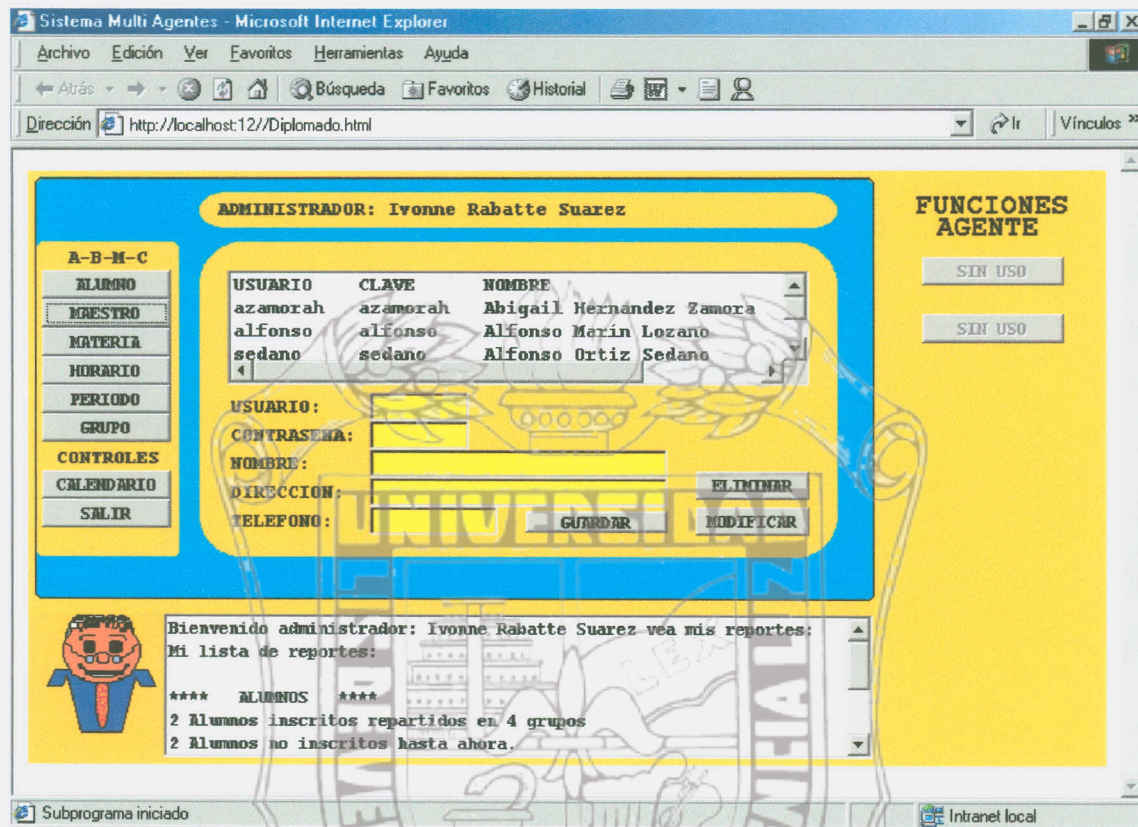


Figura No B.4 Botón Maestro del usuario Administrador

- ◆ **Materia:** Permite dar de alta, eliminar o modificar alguna de las asignaturas que se imparten en el diplomado, para ello, únicamente deberá en el caso de querer modificar ó eliminar, seleccionar la asignatura de la lista que se muestra en la pantalla (ver figura # B.5) y presionar el correspondiente botón de la actividad a realizar (ya sea eliminar o modificar). Para agregar una asignatura solamente deberá llenar la información de dicha materia (ver figura # B.5) y hacer un clic en el botón de guardar y automáticamente aparecerá en la lista la nueva asignatura.

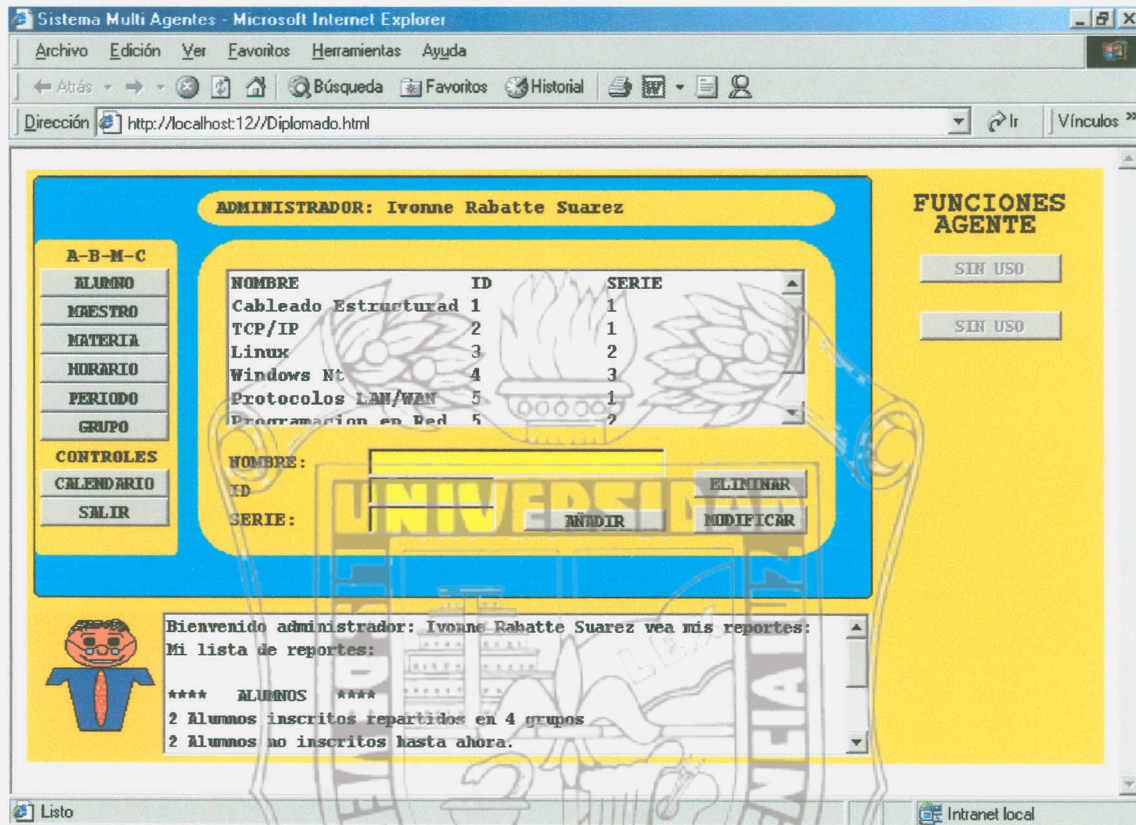


Figura No B.5 Botón Materia del usuario Administrador

- ◆ **Horario:** Le permite al administrador dar de alta el día con su correspondiente rango de horas en el cuál se impartirá alguna asignatura en el diplomado, para ello se debe incluir el rango de horas (por ejemplo 7:00 – 9:00) y seleccionar el día, que por default el sistema los incluye (ver figura # B.6). También podrá llevar a cabo la eliminación o cambio de alguno de los horarios que ya estén incluidos en la lista.
- ◆ **Grupo:** Este botón permite dar de alta, eliminar y modificar los grupos con el correspondiente turno que tendrán lugar en el actual periodo dentro del diplomado. Ver figura # B.7



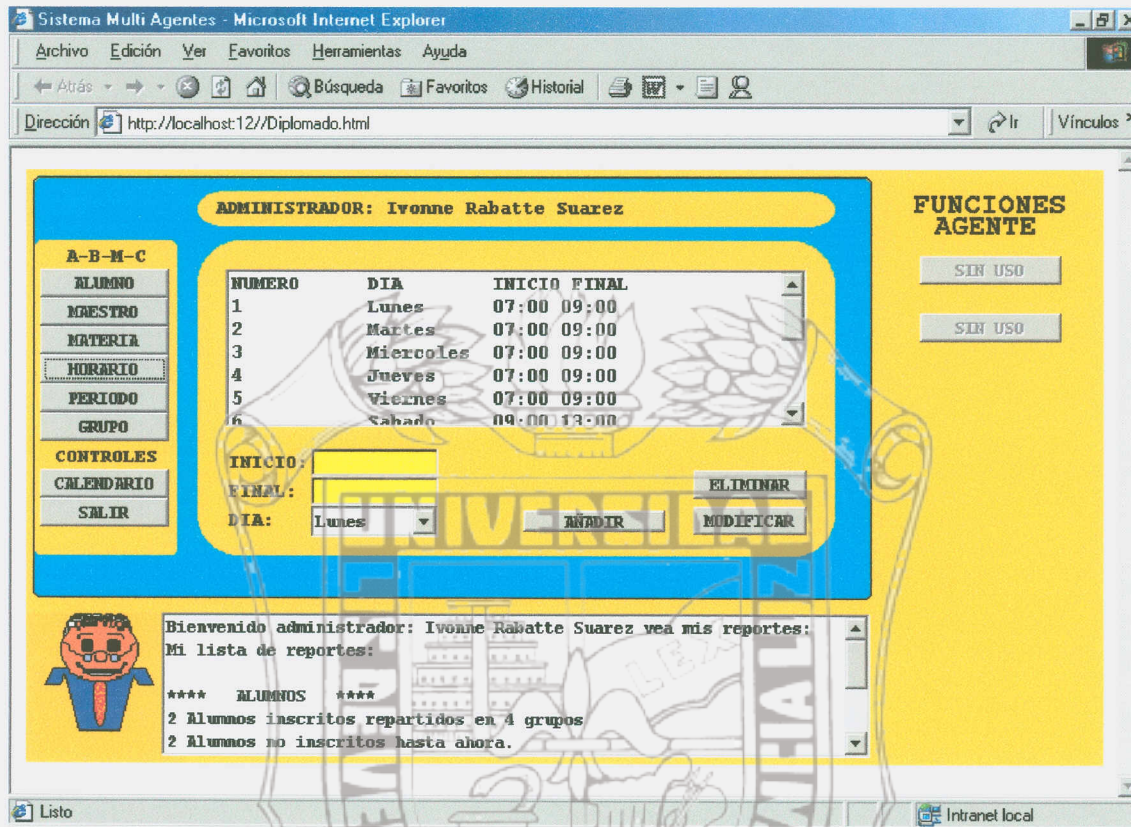


Figura No B.6 Botón Horario del usuario Administrador

- ◆ **Periodo:** En este botón podrá agregar cada uno de los periodos en los que se llevarán a cabo el diplomado, para ello únicamente deberá introducir la fecha de inicio y término de cada uno de los periodos. Ver figura # B.8
- ◆ **Calendario:** Este botón proporciona al administrador la facilidad de armar su horario de clases para el período actual, es decir, permite agregar la hora, la materia, el maestro y el grupo correspondiente, para ello deberá elegir de entre los elementos mostrados en la pantalla (Ver figura # B.9) las horas, los maestros, las materias y los grupos correspondientes para que el administrador pueda armar su calendario de clases.



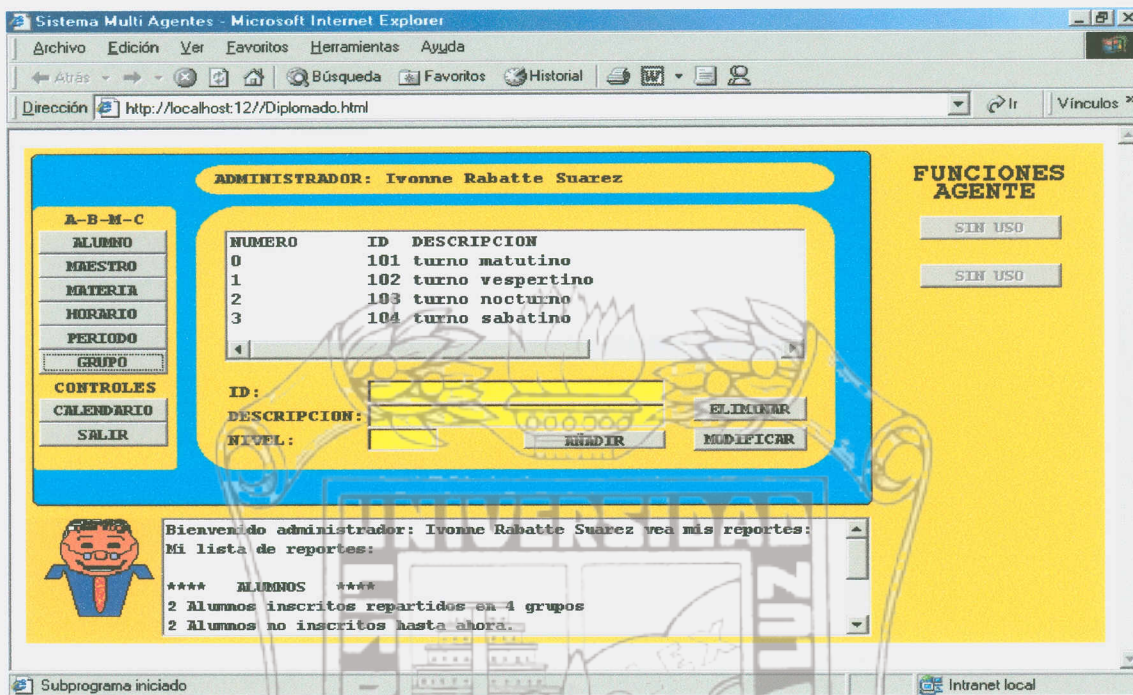


Figura No B.7 Botón Grupo del usuario Administrador

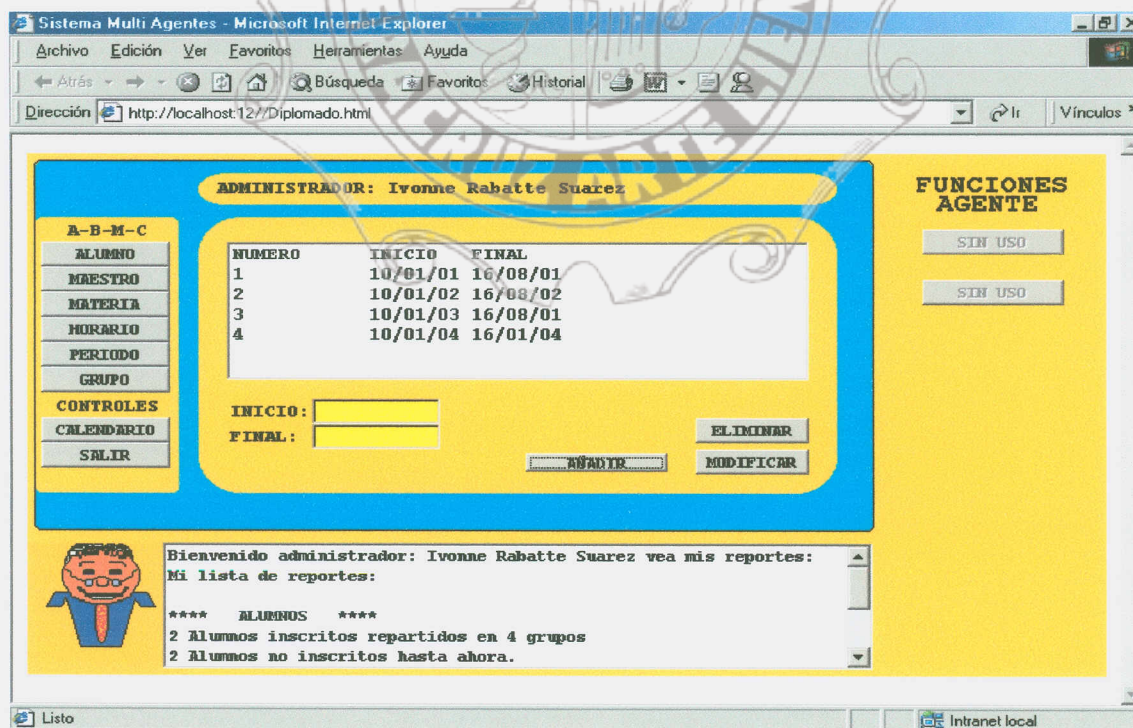


Figura No B.8 Botón Periodo del usuario Administrador

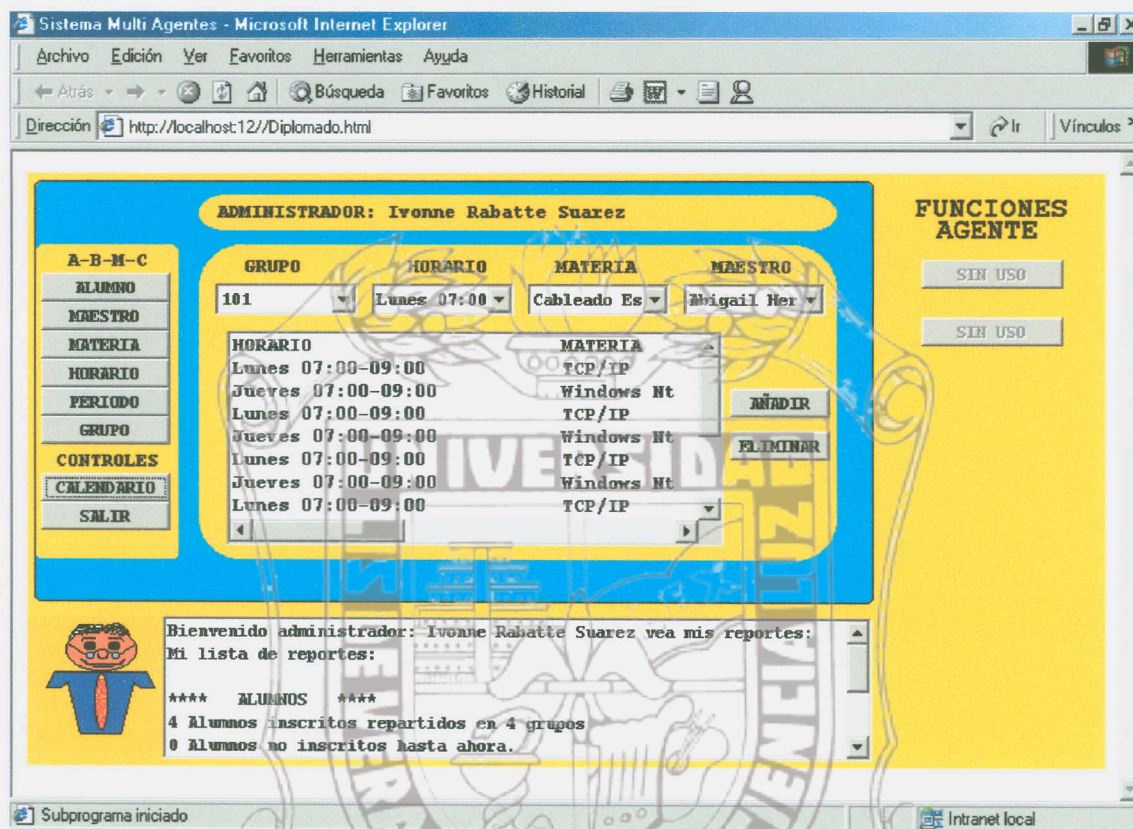


Figura No B.9 Botón Calendario del usuario Administrador.

En el caso de ser un usuario del tipo Maestro, SICADRC, te mostrará una pantalla (ver figura # B.10) que incluye una serie de botones que te permitirán llevar a cabo las actividades siguientes:

- ♦ **Alta de Calificación:** El maestro agregará al Kardex de cada uno de sus alumnos sus correspondientes calificaciones, para ello necesita presionar el botón de calificaciones que aparece en la pantalla, y seleccionar el grupo, la materia y el alumno correspondiente para agregar su calificación y el tipo de examen para guardar dicha información.



- ◆ **Cambio de Calificación:** Debe seleccionar de la pantalla del botón de calificación, el grupo, la materia y el alumno de la lista y modificar su calificación.
- ◆ **Enviar mensajes:** Esta actividad, le permitirá al maestro estar en comunicación con sus alumnos, ya que permite enviar mensajes a cada uno de sus alumnos, para ello tendrá únicamente que escribir el mensaje en la pantalla, elegir el grupo y presionar el botón de enviar.
- ◆ Cuando un maestro inicie su sesión y quiera saber si algún alumno le mandó algún mensaje, el agente le indicará y mostrará el contenido de dicho mensaje asimismo como el alumno que se lo envió.

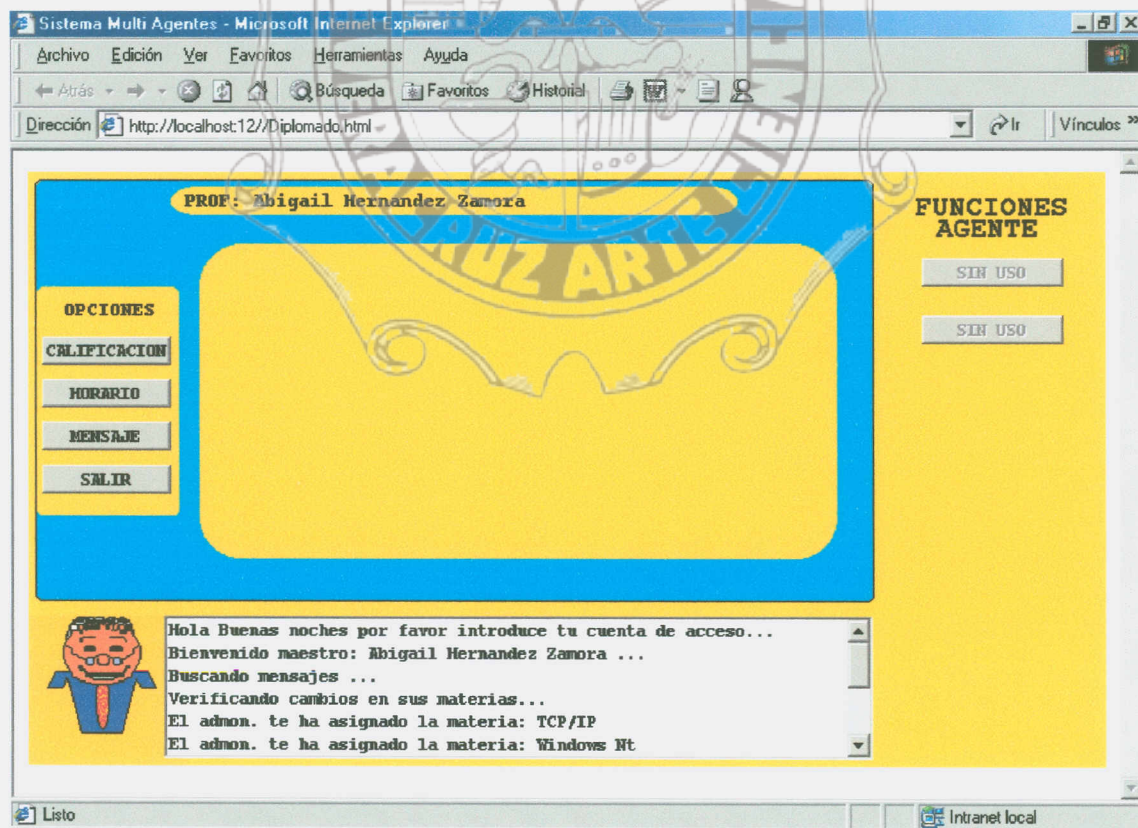
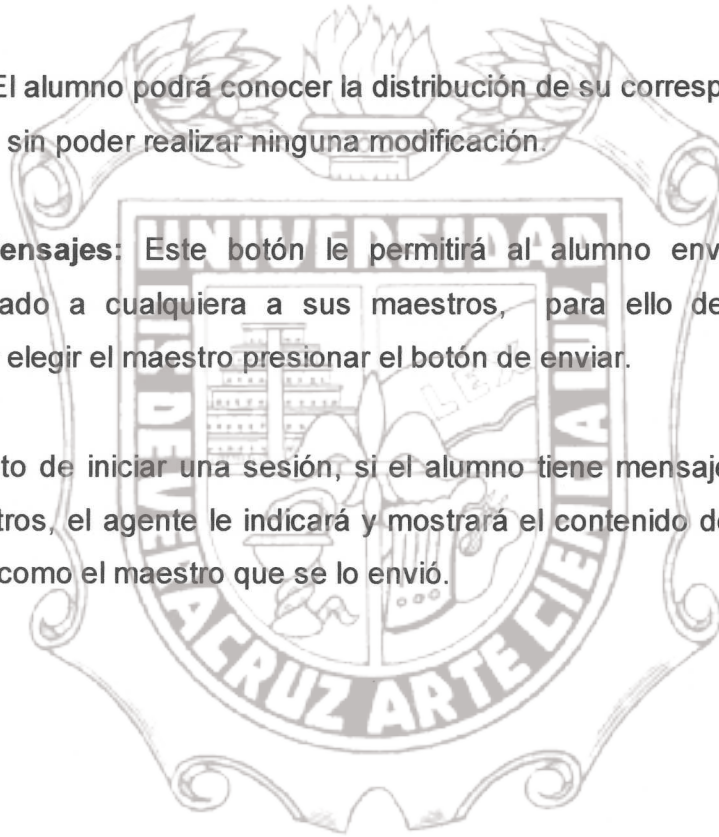


Figura No B.10 Pantalla del usuario Maestro.



En el caso de que el usuario sea un Alumno aparecerá una tercera pantalla (ver figura # B.11) la cual cuenta con las opciones siguientes:

- ◆ **Calificación:** Esta opción le permite al alumno visualizar las calificaciones de todas las asignaturas que hasta el momento ha cursado dentro del diplomado.
- ◆ **Horario:** El alumno podrá conocer la distribución de su correspondiente horario de clases, sin poder realizar ninguna modificación.
- ◆ **Enviar Mensajes:** Este botón le permitirá al alumno enviar un mensaje personalizado a cualquiera de sus maestros, para ello deberá escribir el mensaje y elegir el maestro presionar el botón de enviar.
- ◆ Al momento de iniciar una sesión, si el alumno tiene mensajes de alguno de sus maestros, el agente le indicará y mostrará el contenido de dicho mensaje asimismo como el maestro que se lo envió.



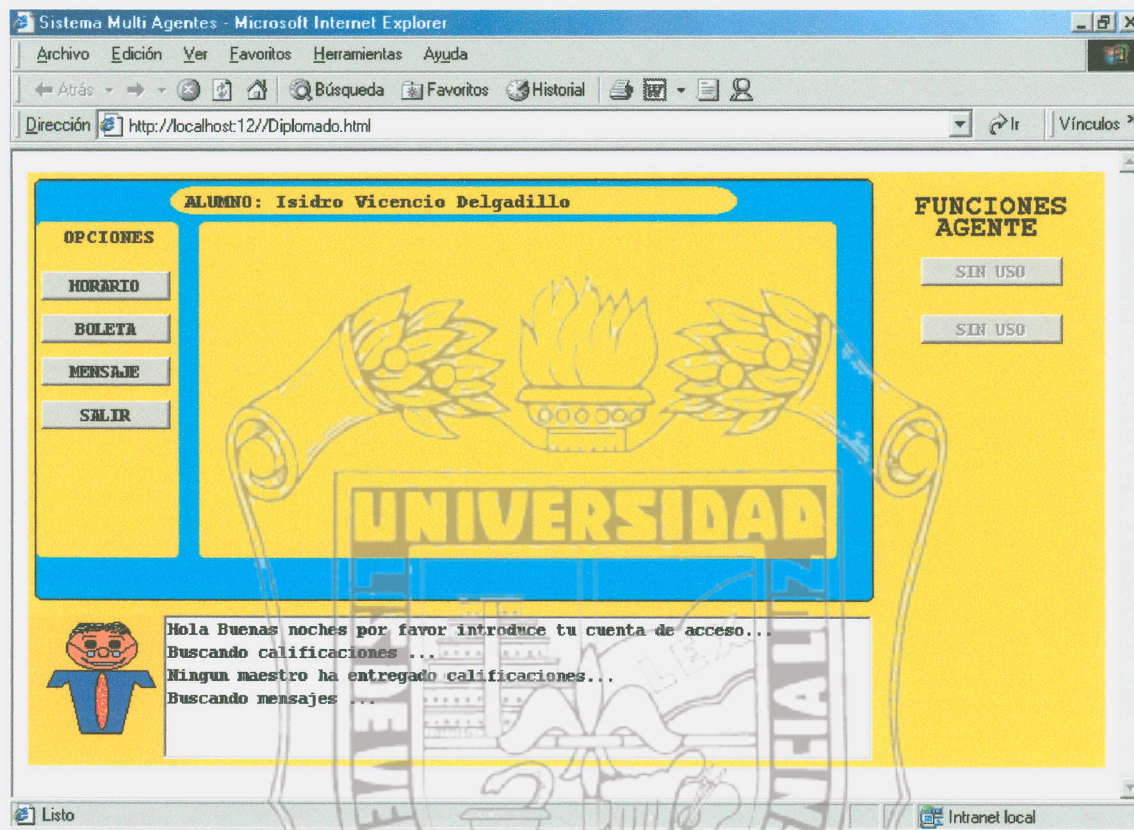


Figura No B.10 Pantalla del usuario Alumnos.

- **Respaldo y restauración de información:**

El respaldo y restauración se realizará manualmente, tomando en cuenta en la decisión del super usuario y de cuanto tiempo lo va a realizar. La base de datos como se ha explicado anteriormente se encuentra diseñada en Access 2000, simplemente para realizar un respaldo de la base de datos se abre el archivo y se guarda con otro nombre para poderlo desligar de la base de datos actual y así poder tener un respaldo exacto de la información en otro archivo.

## D. Código

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java2.sql.*;
import java.net.*;
import java.io.*;
import java.util.Calendar;

class Conexion {
    IDSDriver        driver;
    Connection       con;
    Statement        condicion;
    ResultSet        resultado;
    ResultSetMetaData metadatos;
    String nombre;

    public Conexion() {
        driver = new IDSDriver();
        try {
            con = driver.connect("jdbc:ids://localhost:12/conn?dsn=Diplomado", null);
            condicion = con.createStatement();
        } catch (SQLException e) {
        }
    }

    public int reconoceCuenta(String usuario, String contraseña) {
        int cuenta = 0;
        try {
            resultado = condicion.executeQuery("SELECT nombre FROM alumno WHERE
            usuario='"+usuario+"' and clave='"+contraseña+'");
            if (resultado.next()) {
                cuenta = 1;
                nombre = resultado.getString(1);
            } else {
                resultado = condicion.executeQuery("SELECT nombre FROM maestro
                WHERE usuario='"+usuario+"' and clave='"+contraseña+'");
                if (resultado.next()) {
                    cuenta = 2;
                    nombre = resultado.getString(1);
                } else {
                    resultado = condicion.executeQuery("SELECT nombre FROM
                    administrador WHERE usuario='"+usuario+"' and
                    contraseña='"+contraseña+'");
                    if (resultado.next()) {
                        cuenta = 3;
                        nombre = resultado.getString(1);
                    }
                }
            }
        } catch (Throwable e) {
        }
        return cuenta;
    }

    public String nombreUsuario() {

```





```

        return nombre;
    }

    public void lecturaLista(String cons, List lista) {
        int i, j;
        String res;
        try {
            resultado = condicion.executeQuery(cons);
            metadatos = resultado.getMetaData();
        } catch (Throwable e) {}
        try {
            res = "";
            for (i=0; i < metadatos.getColumnCount(); i++) {
                res = res + (metadatos.getColumnName(i+1)).toUpperCase();
                for (j=(metadatos.getColumnName(i+1)).length(); j <
                    metadatos.getColumnDisplaySize(i+1); j++)
                    res = res + ' ';
            }
            res = res + '\n';
            lista.removeAll();
            lista.add(res);
            while (resultado.next()) {
                res = "";
                for (i=0; i < metadatos.getColumnCount(); i++) {
                    res = res + resultado.getString(i+1);
                    for (j=(resultado.getString(i+1)).length(); j <
                        metadatos.getColumnDisplaySize(i+1); j++)
                        res = res + ' ';
                }
                lista.add(res);
            }
        } catch (Throwable e) {}
    }

    public void lecturaOpcion(String cons, Choice opcion) {
        int i;
        String res;
        try {
            resultado = condicion.executeQuery(cons);
            metadatos = resultado.getMetaData();
        } catch (Throwable e) {}
        try {
            opcion.removeAll();
            while (resultado.next()) {
                res = "";
                for (i=0; i < metadatos.getColumnCount(); i++)
                    res = res + resultado.getString(i+1) + ' ';

                opcion.addItem(res);
            }
        } catch (Throwable e) {}
    }

    protected String regresaDia(String num) {
        Integer Número = new Integer(num);
        switch (Número.intValue()) {
            case 0:
                return "Lunes";
            case 1:

```



```

        return "Martes";
    case 2:
        return "Miercoles";
    case 3:
        return "Jueves";
    case 4:
        return "Viernes";
    case 5:
        return "Sabado";
    case 6:
        return "Domingo";
    }
    return "NULO";
}

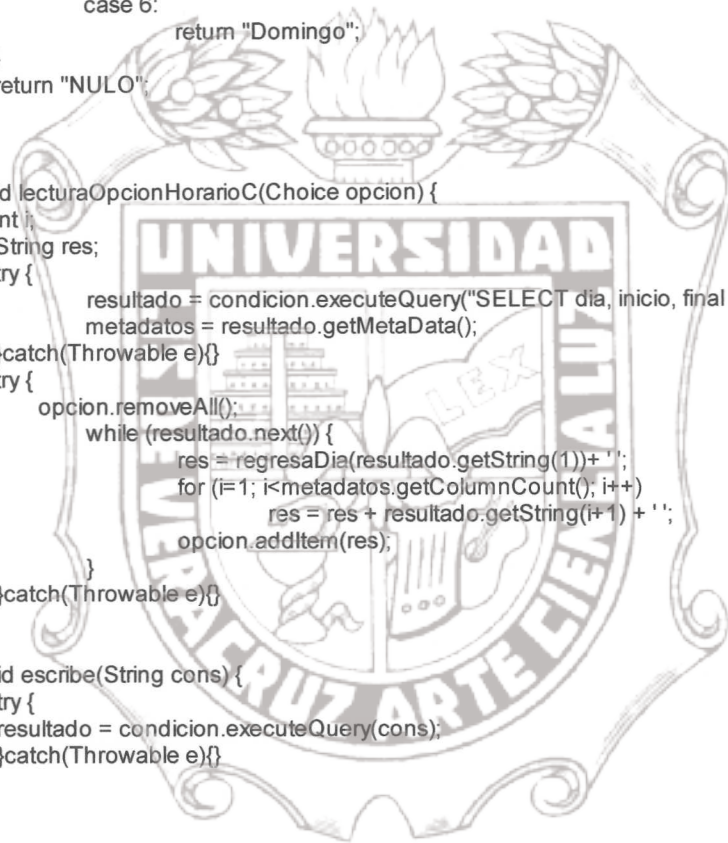
public void lecturaOpcionHorarioC(Choice opcion) {
    int i;
    String res;
    try {
        resultado = condicion.executeQuery("SELECT dia, inicio, final FROM horario");
        metadatos = resultado.getMetaData();
    } catch (Throwable e) {}
    try {
        opcion.removeAll();
        while (resultado.next()) {
            res = regresaDia(resultado.getString(1)) + " ";
            for (i=1; i<metadatos.getColumnCount(); i++)
                res = res + resultado.getString(i+1) + " ";
            opcion.addItem(res);
        }
    } catch (Throwable e) {}
}

public void escribe(String cons) {
    try {
        resultado = condicion.executeQuery(cons);
    } catch (Throwable e) {}
}

public int regresaFilas(String tabla) {
    Integer filas = new Integer(0);
    int si = 0;
    try {
        resultado = condicion.executeQuery("SELECT Número FROM "+tabla);
        while (resultado.next()){
            //recorre la BD
            si++;
        }
        if (si != 0)
            filas = new Integer(resultado.getString(1));
    } catch (Throwable e) {}
    return filas.intValue();
}

public int regresaFilasN(String consulta) {
    Integer filas = new Integer(0);
    int si = 0;
    try {

```



Instituto de Ingeniería  
Universidad Veracruzana

```

        resultado = condicion.executeQuery(consulta);
        while (resultado.next()){
            //recorre la BD
            si++;
        }
    }catch(Throwable e){
    }
    return si;
}

public String regresaUsuario(String tabla) {
    String res = "";
    try {
        resultado = condicion.executeQuery("SELECT usuario FROM maestro WHERE
        nombre = '"+tabla+"'");
        if (resultado.next())
            res = resultado.getString(1);
    }catch(Throwable e){
    }
    return res;
}

public boolean existe(String cons) {
    boolean exis = false;
    try {
        resultado = condicion.executeQuery(cons);
        if (resultado.next())
            exis = true;
    }catch(Throwable e){
    }
    return exis;
}

public String[] lecturaSimple(String cons) {
    int i,j;
    String[] res = new String[10000];
    try {
        resultado = condicion.executeQuery(cons);
        metadatos = resultado.getMetaData();
        j = 0;
        while (resultado.next()) {
            for (i=0; i < metadatos.getColumnCount(); i++) {
                res[j] = resultado.getString(i+1);
                j++;
            }
        }
    }catch(Throwable e){
    }
    return res;
}

public int regresaIndice(String cons) {
    Integer Número = new Integer(0);
    try {
        resultado = condicion.executeQuery(cons);
        if (resultado.next())
            Número = new Integer(resultado.getString(1));
    }catch(Throwable e){
    }
    return Número.intValue();
}

public String regresaPeriodo() {
    String res = "";

```



```

int i;
try {
    resultado = condicion.executeQuery("SELECT final FROM periodo");
    metadatos = resultado.getMetaData();
    while (resultado.next()) {
        for (i=0; i < metadatos.getColumnCount(); i++) {
            res = res + resultado.getString(i+1);
        }
    }
} catch (Throwable e) {}
return res;
}
}
class Utilerias {
public Label Etiqueta(String et, Font f, Color col, int x, int y, int ancho, int alto) {
    Label l = new Label();
    l.setBackground(col);
    l.setFont(f);
    l.setText(et);
    l.setLocation(x, y);
    l.setSize(ancho, alto);
    return l;
}

public TextField CampoTexto(Font f, int x, int y, int ancho, int alto) {
    TextField t = new TextField();
    t.setFont(f);
    t.setLocation(x, y);
    t.setSize(ancho, alto);
    return t;
}

public Button Boton(String et, Font f, int x, int y, int ancho, int alto) {
    Button b = new Button();
    b.setFont(f);
    b.setLabel(et);
    b.setLocation(x, y);
    b.setSize(ancho, alto);
    return b;
}

public List Lista(Font f, int x, int y, int ancho, int alto) {
    List l = new List();
    l.setFont(f);
    l.setLocation(x, y);
    l.setSize(ancho, alto);
    return l;
}

public Choice Opcion(Font f, int x, int y, int ancho, int alto) {
    Choice c = new Choice();
    c.setFont(f);
    c.setLocation(x, y);
    c.setSize(ancho, alto);
    return c;
}

public String regresaPrimero(String cadena, int max_tam) {
    int i = max_tam;
    while (cadena.charAt(i) == ' ')
        i--;
}

```

```

        return cadena.substring(0, i+1);
    }
}

class enChat extends Thread {
    Socket cliente;
    List conectados;
    List platica;
    InputStream entrada;
    OutputStream salida;
    String cadena;
    String usuario;
    byte[] datos = new byte[400];
    int enviados = 0;
    public void datosSocket(List l1, List l2, Socket s) {
        cliente = s;
        conectados = l1;
        platica = l2;
        try {
            entrada = s.getInputStream();
            salida = s.getOutputStream();
        } catch (IOException e) {
            platica.add("Error de I/O: "+e.getMessage());
        }
    }
    public void run() {
        while (true) {
            try {
                entrada.read(datos);
            } catch (IOException e) {}
            cadena = new String(datos);
            if (enviados == 0) {
                usuario = cadena;
                conectados.add(usuario);
                platica.add("El alumno: "+usuario+" acaba de entrar...");
                envia("ADMOR.");
            } else {
                platica.add(usuario+" "+cadena);
                enviados++;
            }
        }
    }

    public String retornaUsuario() {
        return usuario;
    }

    public void envia(String cadena) {
        try {
            salida.write(this.cadena.getBytes());
        } catch (IOException e) {}
        platica.add(usuario+" "+this.cadena);
    }
}

class Servidor implements Runnable {
    List conectados;
    List platica;
    Thread hilo = new Thread(this);
    String usuario;
    String c_usuario;
}

```



```

InputStream entrada;
OutputStream salida;
ServerSocket servidor;
Socket cliente;
enChat lectura = new enChat();
byte[] datos = new byte[400];
int clientes = 0;
int bandera = 0;
int enviados = 0;
int i;
Servidor(List l1, List l2, String nombre) {
    conectados = l1;
    platica = l2;
    usuario = nombre;
    try {
        try {
            cliente = new Socket("otra", 9767);
        } catch (IOException e) { bandera = 1; }

        if (bandera == 1) {
            servidor = new ServerSocket(9767);
            platica.add("**** SERVIDOR DE CHAT V.1.0 ****");
        } else {
            platica.add("BIENVENIDOS AL CHAT DEL DIPLOMADO DE REDES");
        }
    } catch (IOException e) {
        platica.add("Error en conexion:" + e.getMessage());
    }
    catch (SecurityException e) {
        platica.add("Error en permisos:" + e.getMessage());
    }
    conectados.add(nombre);
    hilo.start();
}

public void run() {
    //si es el servidor ...
    if (bandera == 1) {
        while(true) {
            try {
                platica.add("Antes de entrar a accept()");
                lectura.datosSocket(conectados, platica, servidor.accept());
                platica.add("Despues de accept()");
            } catch (IOException e) {
                platica.add("Error en tuberias de I/O");
            } catch (SecurityException e) {
                platica.add("Error en permisos:" + e.getMessage());
            }

            platica.add("Se ha conectado un nuevo usuario");
            lectura.start();
            clientes++;
        }
    }
    } else {
        try {
            entrada = cliente.getInputStream();
            salida = cliente.getOutputStream();
            salida.write(usuario.getBytes());
        } catch (IOException e) {}
        while(true) {
            try {
                entrada.read(datos);
            }
        }
    }
}

```



```

    }catch(IOException e){
    if (enviados == 0) {
        c_usuario = new String(datos);
        conectados.add(c_usuario);
    }
    else
        platica.add(c_usuario+" "+ new String(datos));
    enviados++;
    }
}

public void envia(String cadena) {
    if (bandera == 1)
        lectura.envia(cadena);
    else {
        try {
            salida.write(cadena.getBytes());
        }catch(IOException e){
            platica.add(usuario+" "+cadena);
        }
    }
}

class CuentaAlumno implements ActionListener {
    Font f = new Font("Courier", Font.BOLD, 12);
    Utilerias util = new Utilerias();
    Servidor chat;
    Button btHorario = util.Boton("HORARIO", f, 10, 70, 90, 20);
    Button btHistorial = util.Boton("HISTORIAL", f, 10, 100, 90, 20);
    Button btMensaje = util.Boton("MENSAJE", f, 10, 130, 90, 20);
    Button btBoleta = util.Boton("BOLETA", f, 10, 160, 90, 20);
    Button btChat = util.Boton("PLATICA", f, 10, 190, 90, 20);
    Button btSalir = util.Boton("SALIR", f, 10, 220, 90, 20);
    Button btEnviar = util.Boton("ENVIAR", f, 465, 235, 90, 20);
    List listaAlumnos = util.Lista(new Font("Courier", Font.PLAIN, 12), 145, 50, 60, 170);
    List listaPlatica = util.Lista(new Font("Courier", Font.PLAIN, 14), 215, 50, 340, 170);
    List listaHorario = util.Lista(new Font("Courier", Font.PLAIN, 12), 135, 50, 420, 200);
    TextField tfEnvio = util.CampoTexto(f, 145, 235, 300, 20);
    CuentaAlumno(Container c, String nombre) {
        c.add(btHorario);
        c.add(btHistorial);
        c.add(btMensaje);
        c.add(btBoleta);
        c.add(btChat);
        c.add(btSalir);
        // modulo de chat...
        c.add(listaAlumnos);
        c.add(listaPlatica);
        c.add(tfEnvio);
        c.add(btEnviar);
        //modulo de horario...
        c.add(listaHorario);
        btHorario.addActionListener(this);
        btHistorial.addActionListener(this);
        btMensaje.addActionListener(this);
        btBoleta.addActionListener(this);
        btChat.addActionListener(this);
        btSalir.addActionListener(this);
        btEnviar.addActionListener(this);
        chat = new Servidor(listaAlumnos, listaPlatica, nombre);
        escondeObjetos();
    }
}

```

```

}
private void escondeObjetos() {
    listaAlumnos.setVisible(false);
    listaPlatica.setVisible(false);
    listaHorario.setVisible(false);
    tfEnvio.setVisible(false);
    btEnviar.setVisible(false);
}

public void activaModuloHorario() {
    escondeObjetos();
    listaHorario.setVisible(true);
    listaHorario.removeAll();
    listaHorario.add(" MATERIAS | LUN MAR MIE JUE VIE SAB DOM |
    MAESTRO");
}

public void activaModuloChat() {
    escondeObjetos();
    listaAlumnos.setVisible(true);
    listaPlatica.setVisible(true);
    tfEnvio.setVisible(true);
    btEnviar.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btHorario) {
        escondeObjetos();
        activaModuloHorario();
    }
    if (e.getSource() == btChat) {
        escondeObjetos();
        activaModuloChat();
    }
    if (e.getSource() == btEnviar) {
        chat.envia(tfEnvio.getText());
        tfEnvio.setText("");
    }
}
}

class CuentaMaestro implements ActionListener, ItemListener {
    Font f = new Font("Courier", Font.BOLD, 12);
    Utilerias util = new Utilerias();
    String nombre;
    String[] buffer = new String[10000];
    String[] materias = new String[100];
    Conexion con = new Conexion();
    //modulo principal de opciones...
    Button btCalifica = util.Boton("CALIFICACION", f, 10, 115, 90, 20);
    Button btHorario = util.Boton("HORARIO", f, 10, 145, 90, 20);
    Button btMensaje = util.Boton("MENSAJE", f, 10, 175, 90, 20);
    Button btSalir = util.Boton("SALIR", f, 10, 205, 90, 20);
    Button btCalif = util.Boton("CALIFICAR", f, 140, 220, 100, 20);
    //modulo de evaluacion de los alumnos...
    Label etGrupo = util.Etiqueta("GRUPO:", f, Color.orange, 140, 60, 50, 20);
    Choice chGrupo = util.Opcion(f, 140, 80, 100, 15);
    Label etMateria = util.Etiqueta("MATERIA:", f, Color.orange, 140, 110, 60, 20);
    Choice chMateria = util.Opcion(f, 140, 130, 100, 15);
    Label etCalif = util.Etiqueta("CALIF:", f, Color.orange, 140, 190, 40, 20);
}

```



```

TextField tfCalif = util.CampoTexto(f, 185, 190, 55, 20);
List listaAlumno = util.Lista(f, 260, 60, 290, 200);
List platica;
//modulo de horarios del maestro...
List listaHorario = util.Lista(f, 150, 60, 400, 200);
//modulo de mensajes del maestro...
TextArea ta = new TextArea();
Button btEnviar = util.Boton("ENVIAR", f, 230, 240, 90, 20);
Button btBorrar = util.Boton("BORRAR", f, 380, 240, 90, 20);
CuentaMaestro(Container c, Conexion co, List l) {
    ta.setLocation(150, 60);
    ta.setSize(400, 170);
    c.add(btCalifica);
    c.add(btHorario);
    c.add(btMensaje);
    c.add(btSalir);
    c.add(etGrupo);
    c.add(chGrupo);
    c.add(etMateria);
    c.add(chMateria);
    c.add(etCalif);
    c.add(tfCalif);
    c.add(btCalif);
    c.add(listaAlumno);
    c.add(listaHorario);
    c.add(ta);
    c.add(btEnviar);
    c.add(btBorrar);
    nombre = con.regresaUsuario(co.nombreUsuario());
    con.lecturaOpcion("SELECT grupo.id FROM calendario, grupo "+
        "WHERE calendario.maestro = '"+nombre+"' and "+
        "grupo.Número = calendario.grupo ""GROUP BY grupo.id", chGrupo);
    con.lecturaOpcion("SELECT materia.nombre FROM calendario, materia "+
        "WHERE calendario.maestro = '"+nombre+"' and "+
        "calendario.grupo = '"+chGrupo.getSelectedIndex()+" and materia.Número =
        calendario.materia ""GROUP BY materia.nombre", chMateria);
    buffer = con.lecturaSimple("SELECT Número FROM materia WHERE nombre =
    '"+chMateria.getSelectedItem()+"");
    con.lecturaLista("SELECT alumno, calificacion, caracter FROM cardex "+
        "WHERE periodo = '"+(new Integer(con.regresaFilas("periodo"))).toString()+
        " and materia = '"+buffer[0]+" ""GROUP BY alumno, calificacion, caracter", listaAlumno);
    buffer = con.lecturaSimple("SELECT mensaje FROM maestro WHERE usuario =
    '"+nombre+"'");
    ta.setText(buffer[0]);
    btCalifica.addActionListener(this);
    chGrupo.addItemListener(this);
    chMateria.addItemListener(this);
    btCalif.addActionListener(this);
    btHorario.addActionListener(this);
    btMensaje.addActionListener(this);
    btEnviar.addActionListener(this);
    btBorrar.addActionListener(this);
    platica = l;

    escondeObjetos();
}

protected void escondeObjetos() {
    chGrupo.setVisible(false);
    chMateria.setVisible(false);
}
    
```





```

        btCalif.setVisible(false);
        listaAlumno.setVisible(false);
        etGrupo.setVisible(false);
        etMateria.setVisible(false);
        tfCalif.setVisible(false);
        etCalif.setVisible(false);
        listaHorario.setVisible(false);
        ta.setVisible(false);
        btEnviar.setVisible(false);
        btBorrar.setVisible(false);
    }

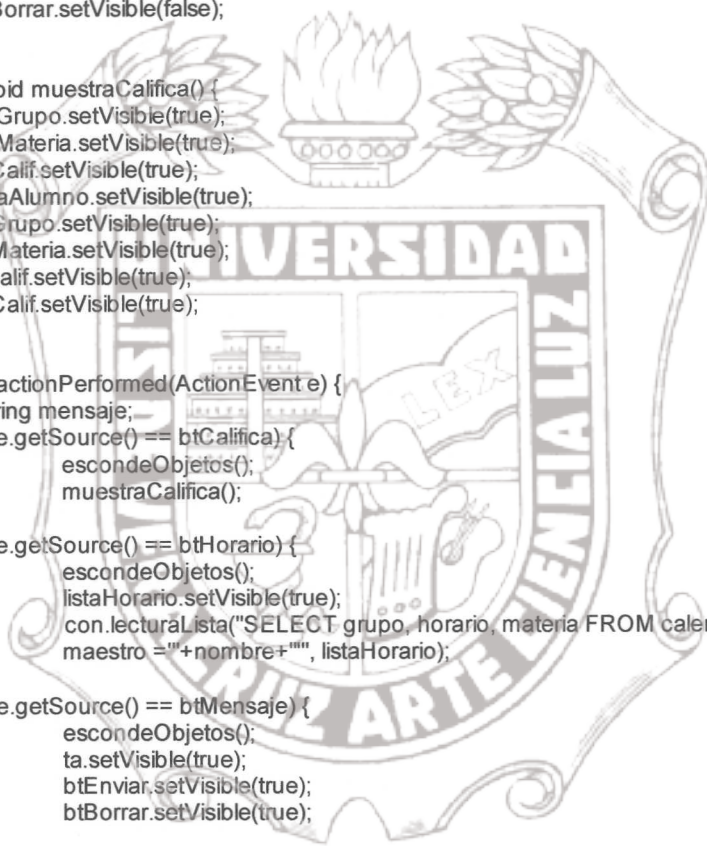
    protected void muestraCalifica() {
        chGrupo.setVisible(true);
        chMateria.setVisible(true);
        btCalif.setVisible(true);
        listaAlumno.setVisible(true);
        etGrupo.setVisible(true);
        etMateria.setVisible(true);
        tfCalif.setVisible(true);
        etCalif.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        String mensaje;
        if (e.getSource() == btCalifica) {
            escondeObjetos();
            muestraCalifica();
        }
        if (e.getSource() == btHorario) {
            escondeObjetos();
            listaHorario.setVisible(true);
            con.lecturaLista("SELECT grupo, horario, materia FROM calendario WHERE
            maestro = '"+nombre+"' ", listaHorario);
        }
        if (e.getSource() == btMensaje) {
            escondeObjetos();
            ta.setVisible(true);
            btEnviar.setVisible(true);
            btBorrar.setVisible(true);
        }

        if (e.getSource() == btEnviar) {
            mensaje = ta.getText();
            con.escribe("UPDATE maestro SET mensaje = '"+mensaje+"' "+
            "WHERE usuario = '"+nombre+"'");
        }

        if (e.getSource() == btBorrar) {
            ta.setText("");
        }
        if (e.getSource() == btCalif) {
            if (tfCalif.getText() == "") {
                tfCalif.setText("0");
            }
        }
        buffer = con.lecturaSimple("SELECT Número FROM materia WHERE nombre =
        '"+chMateria.getSelectedItem()+"");
        platica.add(util.regresaPrimero(listaAlumno.getSelectedItem(), 6)+" "+buffer[0]+"
        "+tfCalif.getText());
        con.escribe("UPDATE cardex SET calificacion = "+tfCalif.getText()+

```



Instituto de Ingeniería  
Universidad Veracruzana

```

    " WHERE alumno = "+util.regresaPrimero(listaAlumno.getSelectedItem(), 6)+" and materia
    = "+buffer[0]);
    con.lecturaLista("SELECT alumno, calificacion FROM cardex "+
    "WHERE periodo = "+(new Integer(con.regresaFilas("periodo"))).toString()+
    " GROUP BY alumno, calificacion", listaAlumno);
    }
}

public void itemStateChanged(ItemEvent e) {
    if (e.getSource() == chGrupo) {
        con.lecturaOpcion("SELECT materia.nombre FROM calendario, materia "+
        "WHERE calendario.maestro = "+nombre+" and " "calendario.grupo =
        "+chGrupo.getSelectedIndex()+" and materia.Número = calendario.materia "+
        "GROUP BY materia.nombre", chMateria);
        con.lecturaLista("SELECT alumno, calificacion FROM cardex "+
        "WHERE periodo = "+(new Integer(con.regresaFilas("periodo"))).toString()+
        " GROUP BY alumno, calificacion", listaAlumno);
    }
    if (e.getSource() == chMateria) {
    }
}

}

class CuentaAdministrador implements ActionListener, ItemListener {
    Font f0= new Font("Courier", Font.PLAIN, 10);
    Font f = new Font("Courier", Font.BOLD, 12);
    Font f1= new Font("Courier", Font.BOLD, 14);
    Utilerias util = new Utilerias();
    Conexion con = new Conexion();
    List textoAgente;
    int total = 0;
    int ind = 0;
    String[] buffer= new String[10000];
    String indice = "";
    String Número = new String();
    //menu del administrador
    Button btAlumno = util.Boton("ALUMNO", f, 10, 70, 90, 20);
    Button btMaestro = util.Boton("MAESTRO", f, 10, 90, 90, 20);
    Button btMateria = util.Boton("MATERIA", f, 10, 110, 90, 20);
    Button btHorario = util.Boton("HORARIO", f, 10, 130, 90, 20);
    Button btPeriodo = util.Boton("PERIODO", f, 10, 150, 90, 20);
    Button btGrupo = util.Boton("GRUPO", f, 10, 170, 90, 20);
    Button btCalendario = util.Boton("CALENDARIO", f, 10, 210, 90, 20);
    Button btSalir = util.Boton("SALIR", f, 10, 230, 90, 20);
    //componentes de alumnos
    List listaAlumno = util.Lista(f1, 140, 70, 410, 80);
    Label etUsuarioA = util.Etiqueta("USUARIO:", f1, Color.orange, 140, 160, 100, 10);
    Label etContrasenaA = util.Etiqueta("CONTRASEÑA:", f1, Color.orange, 140, 180, 100, 10);
    Label etNombreA = util.Etiqueta("NOMBRE:", f1, Color.orange, 140, 200, 100, 10);
    Label etDireccionA = util.Etiqueta("DIRECCIÓN:", f1, Color.orange, 140, 220, 100, 10);
    Label etTelefonoA = util.Etiqueta("TELÉFONO:", f1, Color.orange, 140, 240, 100, 10);
    Label etGrupoA = util.Etiqueta("GRUPO:", f1, Color.orange, 320, 160, 35, 10);
    Label etPeriodoA = util.Etiqueta("PER.:", f1, Color.orange, 320, 180, 35, 10);
    TextField tfUsuarioA = util.CampoTexto(f, 240, 155, 70, 20);
    TextField tfContrasenaA= util.CampoTexto(f, 240, 175, 70, 20);
    TextField tfNombreA = util.CampoTexto(f, 240, 195, 210, 20);
    TextField tfDireccionA = util.CampoTexto(f, 240, 215, 210, 20);
    TextField tfTelefonoA = util.CampoTexto(f, 240, 235, 90, 20);
    Button btGuardarA = util.Boton("GUARDAR", f, 350, 238, 100, 15);
    Button btInscribirA = util.Boton("INSCRIBIR", f, 470, 185, 80, 20);
}

```

Instituto de Ingeniería  
 Universidad Veracruzana



```

Button btEliminarA      = util.Boton("ELIMINAR", f, 470, 210, 80, 20);
Button btModificarA    = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
Choice chGrupoA        = util.Opcion(f, 360, 155, 90, 20);
Choice chPeriodoA      = util.Opcion(f, 360, 175, 90, 20);
//componentes de maestros
List listaMaestro      = util.Lista(f1, 140, 70, 410, 80);
Label etUsuarioM      = util.Etiqueta("USUARIO:", f1, Color.orange, 140, 160, 100, 10);
Label etContrasenaM   = util.Etiqueta("CONTRASEÑA:", f1, Color.orange, 140, 180, 100, 10);
Label etNombreM       = util.Etiqueta("NOMBRE:", f1, Color.orange, 140, 200, 100, 10);
Label etDireccionM    = util.Etiqueta("DIRECCIÓN:", f1, Color.orange, 140, 220, 100, 10);
Label etTelefonoM     = util.Etiqueta("TELÉFONO:", f1, Color.orange, 140, 240, 100, 10);
TextField tfUsuarioM  = util.CampoTexto(f, 240, 155, 70, 20);
TextField tfContrasenaM = util.CampoTexto(f, 240, 175, 70, 20);
TextField tfNombreM   = util.CampoTexto(f, 240, 195, 210, 20);
TextField tfDireccionM = util.CampoTexto(f, 240, 215, 210, 20);
TextField tfTelefonoM = util.CampoTexto(f, 240, 235, 90, 20);
Button btGuardarM     = util.Boton("GUARDAR", f, 350, 238, 100, 15);
Button btEliminarM    = util.Boton("ELIMINAR", f, 470, 210, 80, 20);
Button btModificarM   = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
//componentes de materias
List listaMateria     = util.Lista(f1, 140, 70, 410, 110);
Label etNombreT       = util.Etiqueta("NOMBRE:", f1, Color.orange, 140, 200, 100, 10);
Label etIDT           = util.Etiqueta("ID", f1, Color.orange, 140, 220, 50, 10);
Label etSerieT        = util.Etiqueta("SERIE:", f1, Color.orange, 140, 240, 50, 10);
TextField tfNombreT   = util.CampoTexto(f, 240, 195, 210, 20);
TextField tfIDT       = util.CampoTexto(f, 240, 215, 90, 20);
TextField tfSerieT    = util.CampoTexto(f, 240, 235, 90, 20);
Button btGuardarT     = util.Boton("AÑADIR", f, 350, 238, 100, 15);
Button btEliminarT    = util.Boton("ELIMINAR", f, 470, 210, 80, 20);
Button btModificarT   = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
//componentes de horarios
List listaHorario     = util.Lista(f1, 140, 70, 410, 110);
Label etInicioH       = util.Etiqueta("INICIO:", f1, Color.orange, 140, 200, 60, 10);
Label etFinalH        = util.Etiqueta("FINAL:", f1, Color.orange, 140, 220, 60, 10);
Label etDiaH          = util.Etiqueta("DIA:", f1, Color.orange, 140, 240, 60, 10);
TextField tfInicioH   = util.CampoTexto(f, 200, 195, 90, 20);
TextField tfFinalH    = util.CampoTexto(f, 200, 215, 90, 20);
Choice chDiaH         = util.Opcion(f, 200, 235, 90, 20);
Button btGuardarH     = util.Boton("AÑADIR", f, 350, 238, 100, 15);
Button btEliminarH    = util.Boton("ELIMINAR", f, 470, 210, 80, 20);
Button btModificarH   = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
//componentes de periodos
List listaPeriodo     = util.Lista(f1, 140, 70, 410, 110);
Label etInicioP       = util.Etiqueta("INICIO:", f1, Color.orange, 140, 200, 60, 10);
Label etFinalP        = util.Etiqueta("FINAL:", f1, Color.orange, 140, 220, 60, 10);
TextField tfInicioP   = util.CampoTexto(f, 200, 195, 90, 20);
TextField tfFinalP    = util.CampoTexto(f, 200, 215, 90, 20);
Button btGuardarP     = util.Boton("AÑADIR", f, 350, 238, 100, 15);
Button btEliminarP    = util.Boton("ELIMINAR", f, 470, 210, 80, 20);
Button btModificarP   = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
//componentes de grupos
List listaGrupo       = util.Lista(f1, 140, 70, 410, 110);
Label etIDG           = util.Etiqueta("ID:", f1, Color.orange, 140, 200, 100, 10);
Label etDescripcionG = util.Etiqueta("DESCRIPCIÓN:", f1, Color.orange, 140, 220, 100, 10);
Label etNivelG        = util.Etiqueta("NIVEL:", f1, Color.orange, 140, 240, 50, 10);
TextField tfIDG       = util.CampoTexto(f, 240, 195, 210, 20);
TextField tfDescripcionG = util.CampoTexto(f, 240, 215, 210, 20);
TextField tfNivelG    = util.CampoTexto(f, 240, 235, 50, 20);
Button btGuardarG     = util.Boton("AÑADIR", f, 350, 238, 100, 15);
Button btEliminarG    = util.Boton("ELIMINAR", f, 470, 210, 80, 20);

```



```

Button btModificarG      = util.Boton("MODIFICAR", f, 470, 235, 80, 20);
//componentes de calendarios
List listaCalendario     = util.Lista(f1, 140, 110, 350, 150);
Label etGrupoC          = util.Etiqueta("GRUPO", f1, Color.orange, 150, 60, 60, 10);
Label etHorarioC        = util.Etiqueta("HORARIO", f1, Color.orange, 265, 60, 60, 10);
Label etMateriaC        = util.Etiqueta("MATERIA", f1, Color.orange, 370, 60, 60, 10);
Label etMaestroC        = util.Etiqueta("MAESTRO", f1, Color.orange, 480, 60, 60, 10);
Choice chGrupoC          = util.Opcion(f, 132, 77, 100, 20);
Choice chHorarioC        = util.Opcion(f, 242, 77, 100, 20);
Choice chMateriaC        = util.Opcion(f, 352, 77, 100, 20);
Choice chMaestroC        = util.Opcion(f, 462, 77, 100, 20);
Button btGuardarC        = util.Boton("AÑADIR", f, 495, 150, 70, 20);
Button btEliminarC       = util.Boton("ELIMINAR", f, 495, 180, 70, 20);
CuentaAdministrador(Container c, List l) {
    textoAgente = l;
    textoAgente.add("Mi lista de reportes.");
    textoAgente.add("");
    textoAgente.add("**** ALUMNOS ****");
    textoAgente.add((new Integer(con.regresaFilasN("SELECT * FROM alumno WHERE estado =
1"))).toString()+
" Alumnos inscritos repartidos en "+(new Integer(con.regresaFilasN("SELECT * FROM
grupo"))).toString()+" grupos");
    textoAgente.add((new Integer(con.regresaFilasN("SELECT * FROM alumno WHERE estado =
0"))).toString()+
" Alumnos no inscritos hasta ahora.");
    textoAgente.add((new Integer(con.regresaFilasN("SELECT * FROM alumno"))).toString() " Alumnos
totales registrados.");
    textoAgente.add("");
    textoAgente.add("**** MAESTROS ****");
    textoAgente.add((new Integer(con.regresaFilasN("SELECT maestro FROM calendario GROUP BY
maestro"))).toString() +" Maestros impartiendo clases.");
    textoAgente.add((new Integer(con.regresaFilasN("SELECT * FROM maestro"))).toString() " Maestros
totales registrados.");
    con.lecturaOpcion("SELECT id FROM grupo", chGrupoA);
    con.lecturaOpcion("SELECT inicio, final FROM periodo", chPeriodoA);
    con.lecturaOpcion("SELECT id FROM grupo", chGrupoA);
    chDiaH.addItem("Lunes");
    chDiaH.addItem("Martes");
    chDiaH.addItem("Miercoles");
    chDiaH.addItem("Jueves");
    chDiaH.addItem("Viernes");
    chDiaH.addItem("Sabado");
    chDiaH.addItem("Domingo");
    con.lecturaLista("SELECT usuario, clave, nombre, estado FROM alumno", listaAlumno);
    con.lecturaLista("SELECT usuario, clave, nombre FROM maestro", listaMaestro);
    con.lecturaLista("SELECT nombre, id, serie FROM materia", listaMateria);
    con.lecturaLista("SELECT Número, día, inicio, final FROM horario", listaHorario);
    con.lecturaLista("SELECT Número, inicio, final FROM periodo", listaPeriodo);
    con.lecturaLista("SELECT Número, id, descripcion, nivel FROM grupo", listaGrupo);
    //añadimos al contenedor el menu
    c.add(btAlumno);
    c.add(btMaestro);
    c.add(btMateria);
    c.add(btHorario);
    c.add(btPeriodo);
    c.add(btGrupo);
    c.add(btCalendario);
    c.add(btSalir);
    //añadimos al contenedor los componentes de alumnos
    c.add(listaAlumno);

```

```
c.add(etUsuarioA);
c.add(etContrasenaA);
c.add(etNombreA);
c.add(etDireccionA);
c.add(etTelefonoA);
c.add(etGrupoA);
c.add(etPeriodoA);
c.add(tfUsuarioA);
c.add(tfContrasenaA);
c.add(tfNombreA);
c.add(tfDireccionA);
c.add(tfTelefonoA);
c.add(btGuardarA);
c.add(chGrupoA);
c.add(chPeriodoA);
c.add(btInscribirA);
c.add(btEliminarA);
c.add(btModificarA);
//añadimos al contenedor los componentes de maestros
c.add(listaMaestro);
c.add(etUsuarioM);
c.add(etContrasenaM);
c.add(etNombreM);
c.add(etDireccionM);
c.add(etTelefonoM);
c.add(tfUsuarioM);
c.add(tfContrasenaM);
c.add(tfNombreM);
c.add(tfDireccionM);
c.add(tfTelefonoM);
c.add(btGuardarM);
c.add(btEliminarM);
c.add(btModificarM);
//añadimos al contenedor los componentes de materias
c.add(listaMateria);
c.add(etNombreT);
c.add(etIDT);
c.add(etSerieT);
c.add(tfNombreT);
c.add(tfIDT);
c.add(tfSerieT);
c.add(btGuardarT);
c.add(btEliminarT);
c.add(btModificarT);
//añadimos al contenedor los componentes de horarios
c.add(listaHorario);
c.add(etInicioH);
c.add(etFinalH);
c.add(etDiaH);
c.add(tfInicioH);
c.add(tfFinalH);
c.add(chDiaH);
c.add(btGuardarH);
c.add(btEliminarH);
c.add(btModificarH);
//añadimos al contenedor los componentes de periodos
c.add(listaPeriodo);
c.add(etInicioP);
c.add(etFinalP);
c.add(tfInicioP);
```



```

c.add(tfFinalP);
c.add(btGuardarP);
c.add(btEliminarP);
c.add(btModificarP);
//añadimos al contenedor los componentes de grupos
c.add(listaGrupo);
c.add(etIDG);
c.add(etDescripcionG);
c.add(etNivelG);
c.add(tfIDG);
c.add(tfDescripcionG);
c.add(tfNivelG);
c.add(btGuardarG);
c.add(btEliminarG);
c.add(btModificarG);
//añadimos al contenedor los componentes de calendarios
c.add(listaCalendario);
c.add(etGrupoC);
c.add(etHorarioC);
c.add(etMaestroC);
c.add(etMateriaC);
c.add(chGrupoC);
c.add(chMateriaC);
c.add(chMaestroC);
c.add(chHorarioC);
c.add(btGuardarC);
c.add(btEliminarC);
escondeComponentes();
btAlumno.addActionListener(this);
btMaestro.addActionListener(this);
btMateria.addActionListener(this);
btHorario.addActionListener(this);
btPeriodo.addActionListener(this);
btGrupo.addActionListener(this);
btCalendario.addActionListener(this);
btSalir.addActionListener(this);
btGuardarA.addActionListener(this);
btEliminarA.addActionListener(this);
btModificarA.addActionListener(this);
btInscribirA.addActionListener(this);
btGuardarM.addActionListener(this);
btEliminarM.addActionListener(this);
btModificarM.addActionListener(this);
btGuardarT.addActionListener(this);
btEliminarT.addActionListener(this);
btModificarT.addActionListener(this);
btGuardarH.addActionListener(this);
btEliminarH.addActionListener(this);
btModificarH.addActionListener(this);
btGuardarP.addActionListener(this);
btEliminarP.addActionListener(this);
btModificarP.addActionListener(this);
btGuardarG.addActionListener(this);
btEliminarG.addActionListener(this);
btModificarG.addActionListener(this);
btGuardarC.addActionListener(this);
btEliminarC.addActionListener(this);

```

}

```
private void muestraOpcionCalendario() {
```



```

con.lecturaOpcion("SELECT id FROM grupo", chGrupoC);
con.lecturaOpcion("SELECT nombre FROM materia", chMateriaC);
con.lecturaOpcionHorarioC(chHorarioC);
con.lecturaOpcion("SELECT nombre FROM maestro", chMaestroC);
con.lecturaLista("SELECT * FROM calendario", listaCalendario);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == btAlumno) {
        escondeComponentes();
        muestraAlumno();
    }
    if (e.getSource() == btMaestro) {
        escondeComponentes();
        muestraMaestro();
    }
    if (e.getSource() == btMateria) {
        escondeComponentes();
        muestraMateria();
    }
    if (e.getSource() == btHorario) {
        escondeComponentes();
        muestraHorario();
    }
    if (e.getSource() == btPeriodo) {
        escondeComponentes();
        muestraPeriodo();
    }
    if (e.getSource() == btGrupo) {
        escondeComponentes();
        muestraGrupo();
    }
    if (e.getSource() == btCalendario) {
        escondeComponentes();
        muestraOpcionCalendario();
        muestraCalendario();
    }
    //eventos del modulo de alumnos.
    if (e.getSource() == btGuardarA) {
        if (con.existe("SELECT * FROM alumno WHERE
            usuario="+tfUsuarioA.getText()+""))
            con.escribe("UPDATE alumno SET usuario="+tfUsuarioA.getText()+"",
                clave="+tfContrasenaA.getText()+"", nombre="+
                tfNombreA.getText()+"", direccion="+tfDireccionA.getText()+"",
                telefono="+tfTelefonoA.getText()+" WHERE
                usuario="+tfUsuarioA.getText()+"");
        else
            con.escribe("INSERT INTO alumno VALUES
                (""+tfUsuarioA.getText()+"", ""+tfContrasenaA.getText()+"", ""+
                tfNombreA.getText()+"", ""+tfDireccionA.getText()+"", ""+tfTelefonoA.getTExt()+"", =)");
        con.lecturaLista("SELECT usuario, clave, nombre, estado FROM alumno", listaAlumno);
        tfUsuarioA.setText("");
        tfContrasenaA.setText("");
        tfNombreA.setText("");
        tfDireccionA.setText("");
        tfTelefonoA.setText("");
    }
    if (e.getSource() == btEliminarA) {

```



Instituto de Ingeniería  
Universidad Veracruzana

```

        con.escribe("DELETE FROM alumno WHERE
usuario="" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"");
        con.lecturaLista("SELECT usuario, clave, nombre, estado FROM alumno",
listaAlumno);
    }
    if (e.getSource() == btModificarA) {
        buffer = con.lecturaSimple("SELECT * FROM alumno WHERE
usuario="" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"");
        tfUsuarioA.setText(buffer[0]);
        tfContrasenaA.setText(buffer[1]);
        tfNombreA.setText(buffer[2]);
        tfDireccionA.setText(buffer[3]);
        tfTelefonoA.setText(buffer[4]);
    }
    if (e.getSource() == btInscribirA) {
        con.escribe("INSERT INTO inscripcion VALUES
(" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"", "+
(new Integer(con.regresaFilas("periodo"))).toString()+", "+
(new Integer(chGrupoA.getSelectedIndex()).toString()+", "+
(new java.util.Date().toString()+"");
        con.escribe("UPDATE alumno SET estado = 1 WHERE usuario =
" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"");
        buffer = con.lecturaSimple("SELECT nombre FROM alumno WHERE usuario =
" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"");
        textoAgente.add("El alumno "+ buffer[0]+ " se inscribio en el grupo
" +chGrupoA.getSelectedItem());
        total = con.regresaFilasN("SELECT materia FROM calendario WHERE grupo =
" +chGrupoA.getSelectedIndex());
        buffer = con.lecturaSimple("SELECT materia FROM calendario WHERE grupo =
" +chGrupoA.getSelectedIndex());
        for (ind=0; ind< total; ind++) {
            con.escribe("INSERT INTO cardex VALUES
(" +util.regresaPrimero(listaAlumno.getSelectedItem(), 9)+"", "+
buffer[ind]+"", 0, 0,"+(new
Integer(con.regresaFilas("periodo"))).toString()+"");
        }
        con.lecturaLista("SELECT usuario, clave, nombre, estado FROM alumno",
listaAlumno);
    }
    //eventos del modulo de maestros.
    if (e.getSource() == btGuardarM) {
        if (con.existe("SELECT * FROM maestro WHERE usuario="" +tfUsuarioM.getText()
+ ""))
            con.escribe("UPDATE maestro SET usuario="" +tfUsuarioM.getText()+",
clave="" +tfContrasenaM.getText()+", nombre="" +
tfNombreM.getText()+", direccion="" +tfDireccionM.getText()+",
telefono="" +tfTelefonoM.getText()+ " WHERE
usuario="" +tfUsuarioM.getText()+ """);
        else
            con.escribe("INSERT INTO maestro VALUES (" +tfUs
uarioM.getText()+", "+tfContrasenaM.getText()+", "+
tfNombreM.getText()+", "+tfDireccionM.getText()+", "+tfTelefonoM gText()
",
'SIN MENSAJE');
        con.lecturaLista("SELECT usuario, clave, nombre FROM maestro", listaMaestro);
        tfUsuarioM.setText("");
        tfContrasenaM.setText("");
        tfNombreM.setText("");
        tfDireccionM.setText("");
        tfTelefonoM.setText("");
    }
    if (e.getSource() == btEliminarM) {

```

```

con.escribe("DELETE FROM maestro WHERE
usuario="" + util.regresaPrimero(listaMaestro.getSelectedItem(), 9) + """);
con.lecturaLista("SELECT usuario, clave, nombre FROM maestro", l
istaMaestro);
}
if (e.getSource() == btModificarM) {
buffer = con.lecturaSimple("SELECT * FROM maestro WHERE
usuario="" + util.regresaPrimero(listaMaestro.getSelectedItem(), 9) + """);
tfUsuarioM.setText(buffer[0]);
tfContraseñaM.setText(buffer[1]);
tfNombreM.setText(buffer[2]);
tfDireccionM.setText(buffer[3]);
tfTelefonoM.setText(buffer[4]);
}
//eventos del modulo de materias.
if (e.getSource() == btGuardarT) {
if (con.existe("SELECT * FROM materia WHERE nombre="" + indice + """))
con.escribe ("UPDATE materia SET nombre="" + tfNombreT.getText() + """,
id="" + tfIDT.getText() + """,
serie="" + tfSerieT.getText() + "" WHERE nombre="" + indice + """);
else
con.escribe("INSERT INTO materia VALUES (" + (new
Integer(con.regresaFilas("materia")+1)).toString() + "",
"" + tfNombreT.getText() + "", "" + tfIDT.getText() + "", "" +
tfSerieT.getText() + "");");
con.lecturaLista("SELECT nombre, id, serie FROM materia", listaMateria);
indice = "";
tfNombreT.setText("");
tfIDT.setText("");
tfSerieT.setText("");
}
if (e.getSource() == btEliminarT) {
con.escribe("DELETE FROM materia WHERE
nombre="" + util.regresaPrimero(listaMateria.getSelectedItem(), 19) + """);
con.lecturaLista("SELECT nombre, id, serie FROM materia", listaMateria);
}
if (e.getSource() == btModificarT) {
buffer = con.lecturaSimple("SELECT nombre, id, serie FROM materia WHERE
nombre="" + util.regresaPrimero(listaMateria.getSelectedItem(), 19) + """);
indice = buffer[0];
tfNombreT.setText(buffer[0]);
tfIDT.setText(buffer[1]);
tfSerieT.setText(buffer[2]);
}
//eventos del modulo de horarios.
if (e.getSource() == btGuardarH) {
if (con.existe("SELECT * FROM horario WHERE Número="" + buffer[0])
con.escribe ("UPDATE horario SET inicio="" + tfInicioH.getText() + """,
final="" + tfFinalH.getText() + "" + """,
dia="" + chDiaH.getSelectedIndex() + "" WHERE Número="" + buffer[0]);
else
con.escribe("INSERT INTO horario VALUES (" + (new
Integer(con.regresaFilas("horario")+1)).toString() + "",
"" + tfInicioH.getText() + "", "" +
tfFinalH.getText() + "", "" + chDiaH.getSelectedIndex() + "");");
con.lecturaLista("SELECT Número, dia, inicio, final FROM horario", listaHorario);
buffer[0] = "";
tfInicioH.setText("");
tfFinalH.setText("");
chDiaH.select(0);
}
if (e.getSource() == btEliminarH) {

```



```

con.escribe("DELETE FROM horario WHERE
Número="+util.regresaPrimero(listaHorario.getSelectedItem(), 3));
con.lecturaLista("SELECT Número, dia, inicio, final FROM horario", listaHorario);
}
if (e.getSource() == btModificarH) {
buffer = con.lecturaSimple("SELECT Número, dia, inicio, final FROM horario
WHERE Número="+util.regresaPrimero(listaHorario.getSelectedItem(), 2));

tfInicioH.setText(buffer[2]);
tfFinalH.setText(buffer[3]);
chDiaH.select(buffer[1]);
}
//eventos del modulo de periodos.
if (e.getSource() == btGuardarP) {
if (con.existe("SELECT * FROM periodo WHERE Número="+buffer[0]))
con.escribe ("UPDATE periodo SET inicio="+tfInicioP.getText()+"",
final="+tfFinalP.getText()+"",
" WHERE Número="+buffer[0]);
else
con.escribe("INSERT INTO periodo VALUES ("+
con.regresaFilas("periodo")+",""+tfInicioP.getText()+"", ""+
tfFinalP.getText()+"");
con.lecturaLista("SELECT Número, inicio, final FROM periodo", listaPeriodo);
buffer[0] = "";
tfInicioP.setText("");
tfFinalP.setText("");
}
if (e.getSource() == btEliminarP) {
con.escribe("DELETE FROM periodo WHERE
Número="+util.regresaPrimero(listaPeriodo.getSelectedItem(), 2));
con.lecturaLista("SELECT Número, inicio, final FROM periodo", listaPeriodo);
}
if (e.getSource() == btModificarP) {
buffer = con.lecturaSimple("SELECT Número, inicio, final FROM periodo WHERE
Número="+util.regresaPrimero(listaPeriodo.getSelectedItem(), 2));
tfInicioP.setText(buffer[1]);
tfFinalP.setText(buffer[2]);
}
//eventos del modulo de grupos.
if (e.getSource() == btGuardarG) {
if (con.existe("SELECT * FROM grupo WHERE Número="+buffer[0]))
con.escribe ("UPDATE grupo SET id="+tfIDG.getText()+"",
descripcion="+tfDescripcionG.getText()+"",
" nivel="+tfNivelG.getText()+" WHERE Número="+buffer[0]);
else
con.escribe("INSERT INTO grupo VALUES ("+
con.regresaFilas("grupo")+",""+tfIDG.getText()+"", ""+
tfDescripcionG.getText()+"", ""+tfNivelG.getText()+"");
con.lecturaLista("SELECT Número, id, descripcion, nivel FROM grupo", listaGrupo);
buffer[0] = "";
tfIDG.setText("");
tfDescripcionG.setText("");
tfNivelG.setText("");
}
if (e.getSource() == btEliminarG) {
con.escribe("DELETE FROM grupo WHERE
Número="+util.regresaPrimero(listaGrupo.getSelectedItem(), 2));
con.lecturaLista("SELECT Número, id, descripcion, nivel FROM grupo", listaGrupo);
}
if (e.getSource() == btModificarG) {

```

```

        buffer = con.lecturaSimple("SELECT Número, id, descripcion, nivel FROM grupo
        WHERE Número="+util.regresaPrimero(listaGrupo.getSelectedItem(), 2));
        tfIDG.setText(buffer[1]);
        tfDescripcionG.setText(buffer[2]);
        tfNivelG.setText(buffer[3]);
    }
    //eventos del modulo de calendarios
    if (e.getSource() == btGuardarC) {
        con.escribe("INSERT INTO calendario VALUES ("+(new
        Integer(chHorarioC.getSelectedIndex()).toString()+", "
        +(new Integer(chMateriaC.getSelectedIndex()).toString()+", "
        +con.regresaUsuario(chMaestroC.getSelectedItem())+", "
        +(new Integer(chGrupoC.getSelectedIndex()).toString()+", "
        +(new Integer(con.regresaFilas("periodo"))).toString()+")");
        con.lecturaLista("SELECT * FROM calendario", listaCalendario);
        buffer[0] = "";
    }
    if (e.getSource() == btEliminarC) {
        con.escribe("DELETE FROM calendario WHERE periodo="+
        (new Integer(con.regresaFilas("periodo"))).toString());
        con.lecturaLista("SELECT * FROM calendario", listaCalendario);
    }
}

private void muestraCalendario() {
    listaCalendario.setVisible(true);
    etGrupoC.setVisible(true);
    etHorarioC.setVisible(true);
    etMaestroC.setVisible(true);
    etMateriaC.setVisible(true);
    chGrupoC.setVisible(true);
    chHorarioC.setVisible(true);
    chMaestroC.setVisible(true);
    chMateriaC.setVisible(true);
    btEliminarC.setVisible(true);
    btGuardarC.setVisible(true);
}

private void muestraGrupo() {
    listaGrupo.setVisible(true);
    etIDG.setVisible(true);
    etDescripcionG.setVisible(true);
    etNivelG.setVisible(true);
    tfIDG.setVisible(true);
    tfDescripcionG.setVisible(true);
    tfNivelG.setVisible(true);
    btEliminarG.setVisible(true);
    btModificarG.setVisible(true);
    btGuardarG.setVisible(true);
}

private void muestraPeriodo() {
    listaPeriodo.setVisible(true);
    etInicioP.setVisible(true);
    etFinalP.setVisible(true);
    tfInicioP.setVisible(true);
    tfFinalP.setVisible(true);
    btEliminarP.setVisible(true);
    btModificarP.setVisible(true);
}

```

```
        btGuardarP.setVisible(true);
    }

    private void muestraHorario() {
        listaHorario.setVisible(true);
        etInicioH.setVisible(true);
        etFinalH.setVisible(true);
        etDiaH.setVisible(true);
        tfInicioH.setVisible(true);
        tfFinalH.setVisible(true);
        chDiaH.setVisible(true);
        btEliminarH.setVisible(true);
        btModificarH.setVisible(true);
        btGuardarH.setVisible(true);
    }

    private void muestraMateria() {
        listaMateria.setVisible(true);
        etNombreT.setVisible(true);
        etDT.setVisible(true);
        etSerieT.setVisible(true);
        tfNombreT.setVisible(true);
        tfDT.setVisible(true);
        tfSerieT.setVisible(true);
        btEliminarT.setVisible(true);
        btModificarT.setVisible(true);
        btGuardarT.setVisible(true);
    }

    private void muestraMaestro() {
        listaMaestro.setVisible(true);
        etUsuarioM.setVisible(true);
        etContraseñaM.setVisible(true);
        etNombreM.setVisible(true);
        etDireccionM.setVisible(true);
        etTelefonoM.setVisible(true);
        tfUsuarioM.setVisible(true);
        tfContraseñaM.setVisible(true);
        tfNombreM.setVisible(true);
        tfDireccionM.setVisible(true);
        tfTelefonoM.setVisible(true);
        btEliminarM.setVisible(true);
        btModificarM.setVisible(true);
        btGuardarM.setVisible(true);
    }

    private void muestraAlumno() {
        listaAlumno.setVisible(true);
        etUsuarioA.setVisible(true);
        etContraseñaA.setVisible(true);
        etNombreA.setVisible(true);
        etDireccionA.setVisible(true);
        etTelefonoA.setVisible(true);
        etGrupoA.setVisible(true);
        etPeriodoA.setVisible(true);
        tfUsuarioA.setVisible(true);
        tfContraseñaA.setVisible(true);
        tfNombreA.setVisible(true);
        tfDireccionA.setVisible(true);
        tfTelefonoA.setVisible(true);
    }
}
```





```
chGrupoA.setVisible(true);
chPeriodoA.setVisible(true);
btInscribirA.setVisible(true);
btEliminarA.setVisible(true);
btModificarA.setVisible(true);
btGuardarA.setVisible(true);
}

private void escondeComponentes() {
    listaAlumno.setVisible(false);
    etUsuarioA.setVisible(false);
    etContraseñaA.setVisible(false);
    etNombreA.setVisible(false);
    etDireccionA.setVisible(false);
    etTelefonoA.setVisible(false);
    etGrupoA.setVisible(false);
    etPeriodoA.setVisible(false);
    tfUsuarioA.setVisible(false);
    tfContraseñaA.setVisible(false);
    tfNombreA.setVisible(false);
    tfDireccionA.setVisible(false);
    tfTelefonoA.setVisible(false);
    chGrupoA.setVisible(false);
    chPeriodoA.setVisible(false);
    btInscribirA.setVisible(false);
    btEliminarA.setVisible(false);
    btModificarA.setVisible(false);
    btGuardarA.setVisible(false);
    listaMaestro.setVisible(false);
    etUsuarioM.setVisible(false);
    etContraseñaM.setVisible(false);
    etNombreM.setVisible(false);
    etDireccionM.setVisible(false);
    etTelefonoM.setVisible(false);
    tfUsuarioM.setVisible(false);
    tfContraseñaM.setVisible(false);
    tfNombreM.setVisible(false);
    tfDireccionM.setVisible(false);
    tfTelefonoM.setVisible(false);
    btEliminarM.setVisible(false);
    btModificarM.setVisible(false);
    btGuardarM.setVisible(false);
    listaMateria.setVisible(false);
    etNombreT.setVisible(false);
    etIDT.setVisible(false);
    etSerieT.setVisible(false);
    tfNombreT.setVisible(false);
    tfIDT.setVisible(false);
    tfSerieT.setVisible(false);
    btEliminarT.setVisible(false);
    btModificarT.setVisible(false);
    btGuardarT.setVisible(false);
    listaHorario.setVisible(false);
    etInicioH.setVisible(false);
    etFinalH.setVisible(false);
    etDiaH.setVisible(false);
    tfInicioH.setVisible(false);
    tfFinalH.setVisible(false);
    chDiaH.setVisible(false);
    btEliminarH.setVisible(false);
}
```



Instituto de Ingeniería  
Universidad Veracruzana

```

        btModificarH.setVisible(false);
        btGuardarH.setVisible(false);
        listaPeriodo.setVisible(false);
        etInicioP.setVisible(false);
        etFinalP.setVisible(false);
        tfInicioP.setVisible(false);
        tfFinalP.setVisible(false);
        btEliminarP.setVisible(false);
        btModificarP.setVisible(false);
        btGuardarP.setVisible(false);
        listaGrupo.setVisible(false);
        etIDG.setVisible(false);
        etDescripcionG.setVisible(false);
        etNivelG.setVisible(false);
        tfIDG.setVisible(false);
        tfDescripcionG.setVisible(false);
        tfNivelG.setVisible(false);
        btEliminarG.setVisible(false);
        btModificarG.setVisible(false);
        btGuardarG.setVisible(false);
        listaCalendario.setVisible(false);
        etGrupoC.setVisible(false);
        etHorarioC.setVisible(false);
        etMaestroC.setVisible(false);
        etMateriaC.setVisible(false);
        chGrupoC.setVisible(false);
        chHorarioC.setVisible(false);
        chMaestroC.setVisible(false);
        chMateriaC.setVisible(false);
        btEliminarC.setVisible(false);
        btGuardarC.setVisible(false);
    }

class Seguridad {
    Font f = new Font("Courier", Font.BOLD, 14);
    Utilerias util = new Utilerias();
    Label etNombre = util.Etiqueta("NOMBRE:", f, Color.orange, 200, 153, 60, 20);
    TextField tfNombre = util.CampoTexto(f, 300, 153, 100, 20);
    Label etContrasena = util.Etiqueta("CONTRASEÑA:", f, Color.orange, 200, 183, 100, 20);
    TextField tfContrasena = util.CampoTexto(f, 300, 183, 100, 20);
    Button btContinuar = util.Boton("CONTINUAR", f, 200, 220, 200, 20);
    Seguridad(Container c) {
        tfContrasena.setEchoChar('*');
        c.add(etNombre);
        c.add(tfNombre);
        c.add(etContrasena);
        c.add(tfContrasena);
        c.add(btContinuar);
    }

    public void anadeAccion(ActionListener al) {
        btContinuar.addActionListener(al);
    }

    public void esconde() {
        etNombre.setVisible(false);
        tfNombre.setVisible(false);
        etContrasena.setVisible(false);
        tfContrasena.setVisible(false);
        btContinuar.setVisible(false);
    }
}

```



Instituto de Ingeniería  
Universidad Veracruzana

```

    }
}

public class diplomado extends Applet implements ActionListener, Runnable{
    Seguridad Cuenta;
    Conexion con = new Conexion();
    CuentaAlumno alumno;
    CuentaMaestro maestro;
    CuentaAdministrador administrador;
    List l = new List();
    Thread agentehilo = new Thread(this);
    int pintaSig = 0;
    int pintaUp = 0;
    int errores = 0;
    int ConsultaPeriodo = 0;
    int TipoDeAgente = 0;
    Image anim[] = new Image[10];
    Utilerias util = new Utilerias();
    Button bt1 = util.Boton("SIN USO", new Font("Courier", Font.BOLD, 12), 630, 60, 100, 20);
    Button bt2 = util.Boton("SIN USO", new Font("Courier", Font.BOLD, 12), 630, 100, 100, 20);
    Button bt3 = util.Boton("SIN USO", new Font("Courier", Font.BOLD, 12), 630, 140, 100, 20);
    Button bt4 = util.Boton("SIN USO", new Font("Courier", Font.BOLD, 12), 630, 180, 100, 20);

    public void init() {
        setLayout(null);
        setBackground(Color.orange);
        Cuenta = new Seguridad(this);
        Cuenta.anadeAccion(this);
        anim[0] = getImage(getCodeBase(), "agente.gif");
        anim[1] = getImage(getCodeBase(), "AgentePregunta.gif");
        anim[2] = getImage(getCodeBase(), "AgenteAni01.gif");
        anim[3] = getImage(getCodeBase(), "AgenteAni02.gif");
        anim[4] = getImage(getCodeBase(), "AgenteMolesto01.gif");
        anim[5] = getImage(getCodeBase(), "AgenteMolesto02.gif");
        anim[6] = getImage(getCodeBase(), "AgenteConsultando.gif");
        l.setSize(500, 100);
        l.setLocation(95, 310);
        l.setFont(new Font("Courier", Font.BOLD, 12));
        bt1.setEnabled(false);
        bt2.setEnabled(false);
        bt3.setEnabled(false);
        bt4.setEnabled(false);
        add(l);
        add(bt1);
        add(bt2);
        add(bt3);
        add(bt4);
        l.add("Hola "+SaludaAgente()+" por favor introduce tu cuenta de acceso...");
        bt1.addActionListener(this);
        bt2.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == Cuenta.btContinuar) {
            pintaSig = con.reconoceCuenta(Cuenta.tfNombre.getText(),
            Cuenta.tfContrasena.getText());
            if (pintaSig != 0) {
                Cuenta.esconde();
                switch(pintaSig) {
                    case 1:

```



```

        alumno = new CuentaAlumno(this, Cuenta.tfNombre.getText());
        TipoDeAgente = 1;
        break;

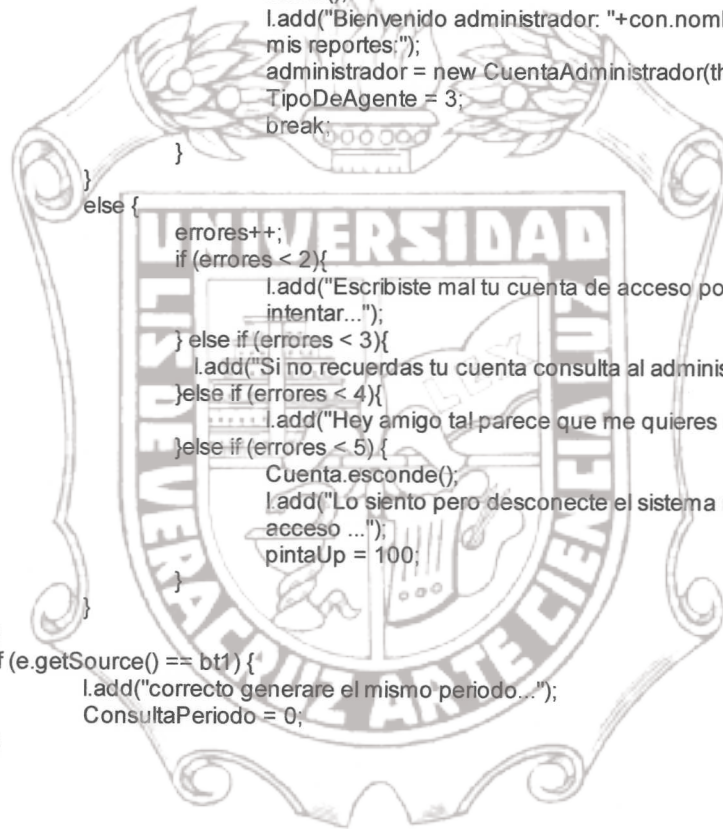
    case 2:
        l.add("Bienvenido maestro: "+con.nombreUsuario()+" ...");
        maestro= new CuentaMaestro(this, con, l);
        TipoDeAgente = 2;
        break;

    case 3:
        l.clear();
        l.add("Bienvenido administrador: "+con.nombreUsuario()+" vea
        mis reportes:");
        administrador = new CuentaAdministrador(this, l);
        TipoDeAgente = 3;
        break;
    }
}
else {
    errores++;
    if (errores < 2){
        l.add("Escribiste mal tu cuenta de acceso por favor vuelve a
        intentar...");
    } else if (errores < 3){
        l.add("Si no recuerdas tu cuenta consulta al administrador...");
    } else if (errores < 4){
        l.add("Hey amigo tal parece que me quieres sabotear...");
    } else if (errores < 5){
        Cuenta.esconde();
        l.add("Lo siento pero desconecte el sistema no podras tener
        acceso ...");
        pintaUp = 100;
    }
}
}
if (e.getSource() == bt1) {
    l.add("correcto generare el mismo periodo...");
    ConsultaPeriodo = 0;
}
}

public void run() {
    while (true) {
        repaint();
        pintaUp++;
        switch(TipoDeAgente) {
            case 3:
                ConsultaPeriodo++;
                break;
        }
        try {Thread.sleep(1000);}
        catch(InterruptedException e) {}
    }
}

public void start() {
    if (agentehilo != null)
        agentehilo.start();
}
}

```



Instituto de Ingeniería  
Universidad Veracruzana

```

public void stop() {
    pintaSig -= 10;
}

public void update(Graphics g) {
// animaciones del agente..
    switch(pintaUp) {
        //estado pasivo...
        case 2:
            g.drawImage(anim[0], 0, 310, this);
            break;
        case 11:
            g.drawImage(anim[2], 0, 310, this);
            break;
        case 13:
            g.drawImage(anim[3], 0, 310, this);
            break;
        case 14:
            g.drawImage(anim[2], 0, 310, this);
            break;
        case 15:
            g.drawImage(anim[3], 0, 310, this);
            break;
        case 17:
            g.drawImage(anim[0], 0, 310, this);
            pintaUp = 0;
            break;
        //estado molesto
        case 102:
            g.drawImage(anim[4], 0, 310, this);
            break;
        case 105:
            g.drawImage(anim[5], 0, 310, this);
            pintaUp = 100;
            break;
    }
    switch(TipoDeAgente) {
        case 3:
            //actividad de consultar el periodo
            switch(ConsultaPeriodo) {
                case 5:
                    ConsultaPeriodo = 0;
                    g.drawImage(anim[6], 0, 310, this);
                    AgenteConsultaPeriodo();
                    g.drawImage(anim[0], 0, 310, this);
                    break;
                //estado de espera...
                case 6:
                    ConsultaPeriodo = 5;
                    break;
            }
            break;
    }

    if (pintaSig == 2) {
        g.setColor(Color.black);
        g.drawRoundRect(5, 5, 590, 294, 10, 10);
        g.setFont(new Font("Courier", Font.BOLD, 14));
        g.setColor(Color.cyan);
    }
}

```



Instituto de Ingeniería  
 Universidad Veracruzana

```

        g.fillRoundRect(6, 6, 589, 293, 10, 10);
        g.setColor(Color.orange);
        g.fillRoundRect(6, 80, 100, 160, 10, 10);
        g.fillRoundRect(120, 50, 450, 220, 60, 60);
        g.fillRoundRect(100, 10, 400, 20, 40, 40);
        g.setColor(Color.black);
        g.drawString("PROF: "+con.nombreUsuario(),110, 24);
        g.drawString("OPCIONES", 25, 100);
        pintaSig += 10;
    }
    if (pintaSig == 3) {
        g.setColor(Color.black);
        g.drawRoundRect(5, 5, 590, 294, 10, 10);
        g.setFont(new Font("Courier", Font.BOLD, 14));
        g.setColor(Color.cyan);
        g.fillRoundRect(6, 6, 589, 293, 10, 10);
        g.setColor(Color.orange);
        g.fillRoundRect(6, 50, 100, 220, 10, 10);
        g.fillRoundRect(120, 50, 450, 220, 60, 60);
        g.fillRoundRect(120, 15, 450, 25, 40, 40);
        g.setColor(Color.black);
        g.drawString("ADMINISTRADOR: "+con.nombreUsuario(), 133, 31);
        g.drawString("A-B-M-C", 28, 65);
        g.drawString("CONTROLES", 20, 205);
        pintaSig += 10;
    }
}

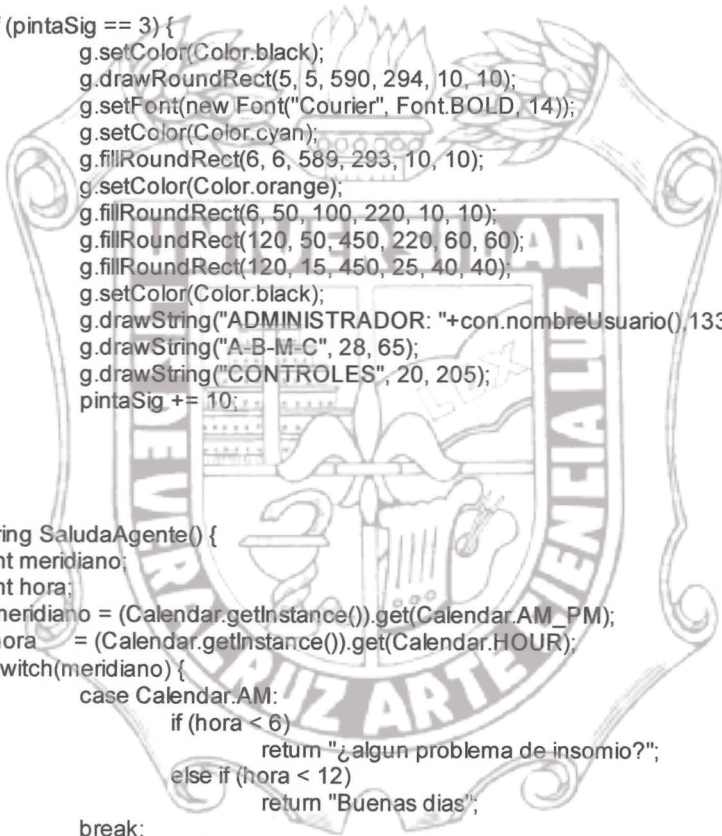
private String SaludaAgente() {
    int meridiano;
    int hora;
    meridiano = (Calendar.getInstance()).get(Calendar.AM_PM);
    hora = (Calendar.getInstance()).get(Calendar.HOUR);
    switch(meridiano) {
        case Calendar.AM:
            if (hora < 6)
                return "¿algun problema de insomnio?";
            else if (hora < 12)
                return "Buenas dias";

            break;
        case Calendar.PM:
            if (hora < 7)
                return "Buenas tardes";
            else if (hora < 12)
                return "Buenas noches";

            break;
    }
    return "amigo";
}

public void paint(Graphics g) {
    switch(pintaSig) {
        case 0:
            g.setColor(Color.black);
            g.drawRoundRect(5, 5, 590, 294, 10, 10);
            g.setFont(new Font("Courier", Font.BOLD, 20));
            g.setColor(Color.cyan);
            g.fillRoundRect(6, 6, 589, 293, 10, 10);
            g.setColor(Color.orange);

```





```

g.fillRoundRect(101, 51, 399, 29, 15, 15);
g.setColor(Color.black);
g.drawString("INTRODUZCA SU CUENTA DE ACCESO", 119,
70);
g.drawString("FUNCIONES", 625, 30);
g.drawString("AGENTE", 640, 45);
g.setColor(Color.orange);
g.fillRoundRect(191, 141, 214, 114, 10, 10);

break;

case 1:
g.setColor(Color.black);
g.drawRoundRect(5, 5, 590, 294, 10, 10);
g.setFont(new Font("Courier", Font.BOLD, 14));

g.setColor(Color.cyan);
g.fillRoundRect(6, 6, 589, 293, 10, 10);
g.setColor(Color.orange);
g.fillRoundRect(6, 35, 100, 235, 10, 10);
g.fillRoundRect(100, 10, 400, 20, 40, 40);
g.fillRoundRect(120, 35, 450, 235, 10, 10);
g.setColor(Color.black);
g.drawString("OPCIONES", 25, 50);

break;
}
}

protected void AgenteConsultaPeriodo(){
int diaA, mesA, anioA;
int diaP, mesP, anioP;
String cad, cad2;
anioA = (Calendar.getInstance()).get(Calendar.YEAR);
mesA = (Calendar.getInstance()).get(Calendar.MONTH);
diaA = (Calendar.getInstance()).get(Calendar.DAY_OF_MONTH);
cad = con.regresaPeriodo();
cad2 = "";
cad2 = cad2 + cad.charAt(0);
cad2 = cad2 + cad.charAt(1);
diaP = (new Integer(cad2)).intValue();
cad2 = "";
cad2 = cad2 + cad.charAt(3);
cad2 = cad2 + cad.charAt(4);
mesP = (new Integer(cad2)).intValue();
cad2 = "";
cad2 = cad2 + cad.charAt(6);
cad2 = cad2 + cad.charAt(7);
anioP = (new Integer(cad2)).intValue();
if ( anioA <= anioP){
    AgenteGeneraReglaPeriodo();
} else {
    if ( mesA <= mesP) {
        if ( diaA > diaP) {
            AgenteGeneraReglaPeriodo();
        }
    }
    else
        AgenteGeneraReglaPeriodo();
}
}

protected void AgenteGeneraReglaPeriodo(){

```



Instituto de Ingeniería  
Universidad Veracruzana

```
        bt1.setEnabled(true);  
        bt2.setEnabled(true);  
        bt1.setLabel("SI");  
        bt2.setLabel("NO");  
        l.add("El periodo de clases ha terminado...");  
        l.add("¿ Genero el mismo periodo con su mismo plan de estudios?");  
        ConsultaPeriodo = 5;  
    }
```

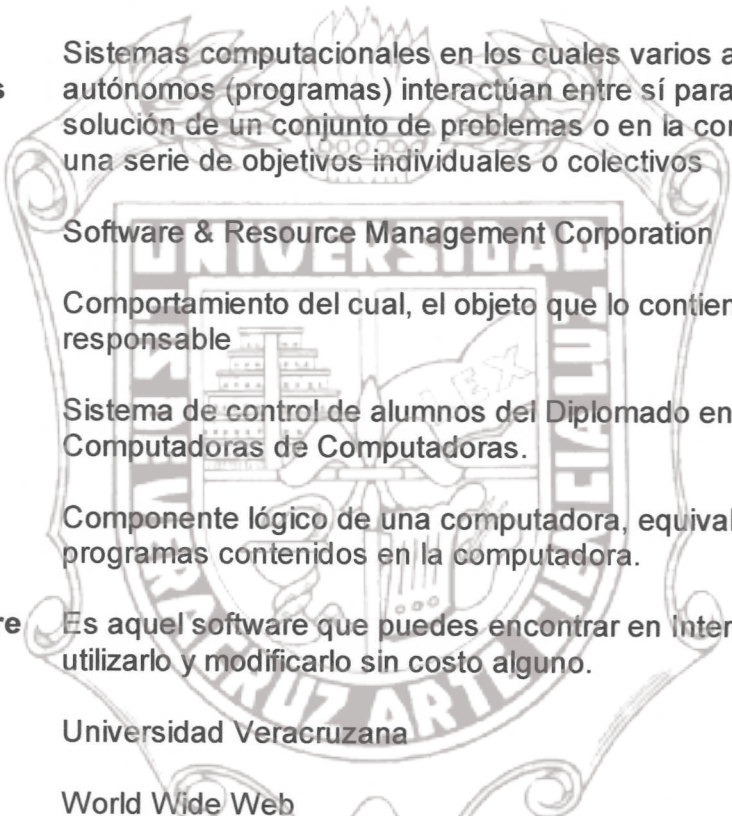


## GLOSARIO

<b>Abstracciones</b>	Denota las características esenciales de un objeto lo cual lo distingue de todas las otras clases de objetos
<b>Access 2000</b>	Software generador de Bases de datos
<b>Agentes</b>	Sistema computacional situado en algún ambiente, capaz de actuar autónomamente y dotado de flexibilidad para alcanzar sus objetivos
<b>Applets</b>	Aplicaciones escritas en lenguaje JAVA, accesibles en un servidor Internet.
<b>API</b>	Programa de Aplicación e Interfase es una lista de rutinas, protocolos y herramientas para desarrollar aplicaciones de software. Entre mejor sea el API será mas fácil desarrollar un programa.
<b>Atributos</b>	Es aquel que denota una parte de un objeto y se utiliza para expresar una característica de una clase.
<b>Banner 2000</b>	Sistema Integral de Información Universitaria.
<b>Base de datos</b>	Conjunto de datos relacionados entre sí.
<b>Campo Llave.</b>	Es el conjunto mínimo de atributos que identifican fielmente a la relación.
<b>Clase</b>	Grupo de objetos con características y comportamientos idénticos o similares
<b>Compilador</b>	Programa que traduce cadenas de un lenguaje a otro distinto, el proceso consiste en traducir todas las cadenas y después lo ejecuta.
<b>DRC</b>	Diplomado en Redes de Computadoras
<b>FEI</b>	Facultad de Estadística e Informática



<b>Free Java</b>	Editor y compilador de programas escritos en lenguaje JAVA.
<b>Hardware</b>	Componente físico de una computadora, son cada una de las partes materiales de una computadora, como monitor, CPU, teclado, etc.
<b>Herencia</b>	Mecanismo que nos permite expresar similitud entre clases
<b>HTML</b>	Lenguaje que sirve para escribir páginas WEB. Hypertext Markup Language; Lenguaje de Marcación de Hipertexto.
<b>IDSServer</b>	Es un servidor de acceso a base de datos desde una intranet o internet que ofrece dos soluciones de bases de datos para aplicaciones web IDS JDBC Driver y IDS HTML extensiones.
<b>Ingeniería de Software</b>	Técnicas para el desarrollo de software de calidad.
<b>Inteligencia Artificial</b>	El arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia.
<b>Internet</b>	Es una red específica mundial que se usa ampliamente para conectar universidades, oficinas de gobierno, compañías y finalmente individuos.
<b>Metodología</b>	Es un conjunto de técnicas y herramientas para resolver un problema.
<b>Modelo E-R</b>	Modelo Entidad Relación
<b>Objeto</b>	Instancia que se identifica a través de un nombre único que lo diferencia de los demás, contiene datos o atributos que toman valores en determinados momentos.
<b>Portabilidad</b>	Es el esfuerzo necesario para transferir el programa de u entorno de sistemas de hardware y / o software a otro.
<b>Protocolos</b>	Descripción formal de formatos de mensajes y las reglas que tienen que seguir dos computadoras para intercambiar dichos mensajes.
<b>Prototipo</b>	Conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida.
<b>RAM</b>	Random Access Memory; Memoria de Acceso Aleatorio



<b>Programación Concurrente</b>	Es cuando dos tareas quieren ocupar el mismo recurso al mismo tiempo, entonces el procesador y el control de tareas deben resolver este recurriendo a ella.
<b>Programación Orientada a Objetos</b>	se refiere al estilo de programación que se basa en los conceptos y encapsulamiento
<b>Sistemas Multiagentes</b>	Sistemas computacionales en los cuales varios agentes semi-autónomos (programas) interactúan entre sí para colaborar en la solución de un conjunto de problemas o en la consecución de una serie de objetivos individuales o colectivos
<b>SCT</b>	Software & Resource Management Corporation
<b>Servicios</b>	Comportamiento del cual, el objeto que lo contiene, se hace responsable
<b>SICADRC</b>	Sistema de control de alumnos del Diplomado en Redes de Computadoras de Computadoras.
<b>Software</b>	Componente lógico de una computadora, equivale a los programas contenidos en la computadora.
<b>Software libre</b>	Es aquel software que puedes encontrar en Internet y puedes utilizarlo y modificarlo sin costo alguno.
<b>UV</b>	Universidad Veracruzana
<b>WWW</b>	World Wide Web